

UNIVERSIDADE FEDERAL DO PARANÁ

MARIANE GAVIOLI BERGAMINI FERNANDES

OTIMIZAÇÃO MULTI OBJETIVO UTILIZANDO METAHEURÍSTICAS PARA
MODELOS EM ENGENHARIA

CURITIBA

2024

MARIANE GAVIOLI BERGAMINI FERNANDES

OTIMIZAÇÃO MULTIOBJETIVO UTILIZANDO METAHEURÍSTICAS PARA
MODELOS EM ENGENHARIA

Tese apresentada ao curso de Pós-Graduação em Engenharia Elétrica, Áreas de Concentração Sistemas Eletrônicos, Departamento de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutora em Engenharia Elétrica.

Orientador: Prof. Dr. Gideon Villar Leandro
Coorientador: Prof. Dr. Eduardo Parente Ribeiro

CURITIBA

2024

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Fernandes, Mariane Gavioli Bergamini

Otimização multiobjetivo utilizando metaheurísticas para modelos em engenharia / Mariane Gavioli Bergamini Fernandes. – Curitiba, 2024.

1 recurso on-line : PDF.

Tese (Doutorado) - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica.

Orientador: Gideon Villar Leandro

Coorientador: Eduardo Parente Ribeiro

1. Estimativa de parâmetros. 2. Otimização matemática. 3. Sistemas dinâmicos. 4. Modelos em engenharia. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Engenharia Elétrica. III. Leandro, Gideon Villar. IV. Ribeiro, Eduardo Parente. V. Título.

Bibliotecário: Elias Barbosa da Silva CRB-9/1894



MINISTÉRIO DA EDUCAÇÃO
SETOR DE TECNOLOGIA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA
ELÉTRICA - 40001016043P4

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **MARIANE GAVIOLI BERGAMINI FERNANDES** intitulada: **OTIMIZAÇÃO MULTIOBJETIVO UTILIZANDO METAHEURÍSTICAS PARA MODELOS EM ENGENHARIA**, que após terem inquirido a aluna e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutora está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 10 de Maio de 2024.

Assinatura Eletrônica
06/06/2024 10:55:29.0
GIDEON VILLAR LEANDRO
Presidente da Banca Examinadora

Assinatura Eletrônica
06/06/2024 10:59:46.0
RICARDO SCHUMACHER
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
06/06/2024 18:07:37.0
GILBERTO REYNOSO MEZA
Avaliador Externo (PONTIFÍCIA UNIVERSIDADE CATÓLICA DO
PARANÁ)

Assinatura Eletrônica
07/06/2024 16:24:39.0
ROBERTO ZANETTI FREIRE
Avaliador Externo (PONTIFÍCA UNIVERSIDADE CATÓLICA DO
PARANÁ)

Av. Cel. Francisco H. dos Santos, 210, Bairro Jardim das Américas, Bloco PK/PL - DELT, Setor Tecnologia, Campus Centro Politécnico - Curitiba - Paraná - Brasil
CEP 81530-000 - Tel: (41) 3361-3622 - E-mail: ppgee@ufpr.br

Documento assinado eletronicamente de acordo com o disposto na legislação federal Decreto 8539 de 08 de outubro de 2015.
Gerado e autenticado pelo SIGA-UFPR, com a seguinte identificação única: 370872

**Para autenticar este documento/assinatura, acesse <https://siga.ufpr.br/siga/visitante/autenticacaoassinaturas.jsp>
e insira o código 370872**

Para Edith

AGRADECIMENTOS

Primeiro e mais importante, agradeço a Deus por me guiar e me manter firme na fé em toda a minha trajetória para que pudesse conquistar meus sonhos.

Aos meus pais, Maura e Edval, por abdicarem de seus sonhos e manter os meus sonhos. Esta conquista não é só minha, mas sim, nossa.

Ao Felipe, pelo amor, paciência, incentivo para alcançar meus sonhos e por embarcar junto nesta jornada comigo.

A minha irmã Nayara e ao meu sobrinho Eduardo, por se manterem presentes a todo o momento me dando forças.

A minha madrinha Leni, pelos inúmeros conselhos e palavras de amor que me manteve focada na minha trajetória.

A Adriane, Vanderlei e Ana Fernanda pelo incentivo, orações e carinho.

A minha amiga de longa data Ana Rúbia pelas boas conversas e palavras de afetos.

Aos meus orientadores, Eduardo, Gideon (principalmente) e Gustavo, por apostarem em mim, pelos ensinamentos diários, paciência e toda contribuição neste trabalho.

Sem a contribuição de vocês este trabalho não teria dado certo. Vocês são exemplos que levarei por toda a minha trajetória.

Aos demais professores e funcionários da Universidade Federal do Paraná que contribuíram diretamente e indiretamente em minha jornada.

Aos amigos do laboratório LaSICO, Carla, Renato, Ádamo, Wagner e Cristiano pela convivência, aprendizados e boas risadas. E, aos demais colegas.

E, ao apoio financeiro de suma importância da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

“Confia ao Senhor as tuas obras, e
teus pensamentos serão estabelecidos.”

Provérbios, 16:3

RESUMO

Na área da engenharia, ao se trabalhar com um problema de otimização, pode haver necessidade de se otimizar três ou mais objetivos como, por exemplo, no projeto de geradores em engenharia elétrica, de motores em engenharia mecânica e de estruturas na engenharia civil. As metaheurísticas multiobjetivo são técnicas bastante atrativas quando se lida com problemas de múltiplos objetivos conflitantes, pois conseguem explorar os *trade-offs* entre diferentes objetivos e alcançar as soluções em um tempo computacional razoável para problemas não lineares/grande escala (dimensão). A busca por novas técnicas para a otimização de sistemas resulta no desenvolvimento de novas metaheurísticas para suprir as lacunas que as técnicas anteriores não foram tão eficientes (como exemplo, técnicas com tempo computacional razoável), neste sentido, este trabalho apresenta duas novas metaheurísticas multiobjetivo. A primeira metaheurística desenvolvida utiliza como base o Algoritmo do Leão e os conceitos de dominância e decomposição, o qual será denominado MOLA (*Multi-objective Optimization Lion Algorithm*). A segunda metaheurística desenvolvida utiliza o Algoritmo do Leão e a Busca Local Iterativa trabalhando de forma cooperativa e o conceito de vetores suporte, o qual será denominado MOLAILS (*Multi-objective Optimization Lion Algorithm Iterated Search Local*). As novas metaheurísticas propostas foram submetidas a um conjunto de funções de teste para otimização multiobjetivo e seus desempenhos foram comparados com o de outros algoritmos de otimização por meio da distância geracional invertida (IGD), pelo hipervolume (HV) e por inferência não paramétricas. Os resultados dos testes em *benchmarks* mostraram que os algoritmos MOLA e o MOLSLA superam ou pelo menos tem desempenho semelhante ao *Multi-objective Grey Wolf Optimizer* (MOGWO), *Multiple Objective Particle Swarm Optimization* (MOPSO), *Multi-objective Salp Swarm Algorithm* (MSSA), *Multi-objective Evolutionary Algorithm Based on Decomposition* (MOEA/D) e Algoritmo Genético de Sorting Não Dominado III (NSGA-III). Os algoritmos propostos foram então aplicados em problemas de modelagem da área da engenharia a estimação de parâmetros da malha hidráulica de uma usina hidroelétrica e a a estimação de parâmetros do sistema massa-mola-amortecimento. Os resultados médios dos índices de desempenho mostraram que as metaheurísticas propostas MOLA e MOLAILS obtiveram resultados semelhantes ou superiores do algoritmo MOGWO.

Palavras-chave: Estimação de parâmetros. Metaheurísticas multiobjetivo. Sistemas dinâmicos.

ABSTRACT

In the area of engineering, when working with an optimization problem, there may be a need to optimize three or more objectives, for example, in the design of generators in electrical engineering, engines in mechanical engineering and structures in civil engineering. Multi-objective metaheuristics are very attractive techniques when dealing with problems with multiple conflicting objectives, as they can explore trade-offs between different objectives and reach solutions in a reasonable computational time for non-linear/large-scale problems. The search for new techniques for system optimization results in the development of new metaheuristics to fill the gaps that previous techniques were not as efficient (for example, techniques with reasonable computational time). In this sense, this work presents two new multi-objective metaheuristics. The first metaheuristic developed uses the Lion Algorithm and the concepts of dominance and decomposition as a basis, which will be called MOLA (Multi-objective Optimization Lion Algorithm). The second metaheuristic developed uses the Lion Algorithm and Iterative Local Search working cooperatively and the concept of support vectors, which will be called MOLAILS (Multi-objective Optimization Lion Algorithm Iterated Search Local). The proposed new metaheuristics were subjected to a set of test functions for multi-objective optimization and their performance was compared with that of other optimization algorithms using inverted generational distance (IGD), hypervolume (HV) and non-parametric inference. Benchmark test results showed that MOLA and MOLSLA algorithms outperform or at least have similar performance to Multi-objective Gray Wolf Optimizer (MOGWO), Multiple Objective Particle Swarm Optimization (MOPSO), Multi-objective Salp Swarm Algorithm (MSSA), Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) and Non-Dominated Sorting Genetic Algorithm III (NSGA-III). The proposed algorithms were then applied to engineering modeling problems, the estimation of parameters of the hydraulic network of a hydroelectric plant and the estimation of parameters of the mass-spring-damping system. The average results of the performance indexes showed that the proposed metaheuristics MOLA and MOLAILS obtained similar or higher results to the MOGWO algorithm.

Keywords: Dynamic systems. Multi-objective metaheuristics. Parameters estimation.

LISTA DE FIGURAS

Figura 1 – Exemplo de Mapeamento de Pontos entre ds Espaços de Decisão e Objetivos.	29
Figura 2 – Exemplo do conceito de dominância.	30
Figura 3 – Fronteira de Pareto ótima e conceito de dominância.	31
Figura 4 – Exemplo de um problema de três objetivos e com 15 pontos de referência em um plano de referência normalizado.	34
Figura 5 – Principais indicadores de qualidade para problemas de otimização em engenharia.	43
Figura 6 – Metaheurísticas utilizadas com frequência na área de engenharia.	45
Figura 7 – Principais índices de desempenho utilizados na área de engenharia.	46
Figura 8 – Principais testes estatísticos.	48
Figura 9 – Principais funções de <i>Benchmark</i>	52
Figura 10 – Fronteiras de Pareto da MaF1.	56
Figura 11 – Fronteiras de Pareto da MaF2.	56
Figura 12 – Fronteiras de Pareto da MaF3.	57
Figura 13 – Fronteiras de Pareto da MaF4.	57
Figura 14– Fronteiras de Pareto da MaF5.	58
Figura 15– Fronteiras de Pareto da MaF6.	58
Figura 16 – Fronteiras de Pareto da MaF7.	59
Figura 17 – Fronteiras de Pareto da MaF8.	59
Figura 18 – Fronteiras de Pareto da MaF9.	60
Figura 19 – Fronteiras de Pareto da MaF10.	60
Figura 20 – Fronteiras de Pareto da MaF11.	61
Figura 21 – Fronteiras de Pareto da MaF12.	61
Figura 22 – Fronteiras de Pareto da Ma13.	62
Figura 23 – Fronteiras de Pareto da MaF14.	62
Figura 24 – Fronteiras de Pareto da MaF15.	63
Figura 25 – Fluxograma do algoritmo MOLA.	68
Figura 26 – Exemplo de pontos de referência para $M = 3$ e $m = 4$	71
Figura 27 – Exemplo de checagem de não dominância.	75
Figura 28 – Formação de hiperplano a partir dos pontos extremos para um problema de três objetivos.	76

Figura 29 – Taxonomia de metaheurísticas hibridizadas.	78
Figura 30– Fluxograma do algoritmo cooperativo proposto – <i>MOLAILS</i>	80
Figura 31 – Geração de vetores de referência uniformemente distribuídos em um espaço de três objetivos.	82
Figura 32 – Classificações médias diferentes do MOLA.	98
Figura 33 – Classificações médias usando a métrica HV.	104
Figura 34 – Soluções obtidas por MOLA e MOLAILS para MaF2.	105
Figura 35 – Soluções obtidas por MOLA para MaF7.	106
Figura 36 – Soluções obtidas por MOLA e MOLAILS para MaF1.	107
Figura 37 – Processo de Identificação de Sistemas.	109
Figura 38 – Sistema Não Linear sob Estudo.	110
Figura 39 – Modelagem da Válvula Piloto/Proporcional.	111
Figura 40 – Modelagem da Válvula Distribuidora.	111
Figura 41 – Modelagem do Distribuidor.	112
Figura 42 – Modelagem da Turbina Hidráulica.	112
Figura 43 – Desempenho do sistema regulador de turbina hidráulica.	117
Figura 44 – Dados de validação do sistema regulador de turbina hidráulica.	119
Figura 45 – Representação do sistema massa-mola-amortecimento para 1 GDL. .	121
Figura 46 – Coordenadas paralelas de conjuntos de soluções de Pareto obtidas com o algoritmo MOLA.	126
Figura 47 – Coordenadas paralelas de conjuntos de soluções de Pareto obtidas com o algoritmo MOLAILS.	127

LISTA DE ABREVIATURAS

BIP	-Brazilian Interconnected Power
FIR	-Finite Impulse Response
HC	-Hill Climbing
IIR	-Infinite Impulse Response
HTRS	-The hydraulic turbine regulating system
ITAE	-Integral Time-Weighted Absolute Error
LFC	-Load Frequency Controller
LA	-Lion Algorithm
MAE	-Mean Absolute Error
MSE	-Mean Squared Error
MOEA/D	-Multi-objective Evolutionary Algorithm based on Decomposition
MOGWO	-Multi-objective Grey Wolf Optimizer
MOLA	-Multi-objective Optimization Lion Algorithm
MOLAILS	-Multi-objective Optimization Lion Algorithm Iterated LocalSearch
MOPSO	-Multiple Objective Particle Swarm Optimization
MSSA	-Multi-objective Salp Swarm Algorithm
PID	-Proporcional Integral Derivativo
PWM	-Pulse-width Modulation
RMSE	-Root Mean Squared Error
RMSLE	- Root Mean Square Algorithm Error

SUMÁRIO

1.	INTRODUÇÃO	16
1.1.	MOTIVAÇÃO	23
1.2.	OBJETIVOS	23
1.2.1.	Objetivos Específicos	24
1.3.	CONTRIBUIÇÕES DA TESE	24
1.4.	JUSTIFICATIVA.....	25
1.5.	ESTRUTURA DA TESE	25
2.	OTIMIZAÇÃO MULTIOBJETIVO UTILIZANDO METAHEURÍSTICAS	27
2.1.	OTIMIZAÇÃO MULTIOBJETIVA	27
2.1.1.	Dominância de Pareto	29
2.1.2.	Otimalidade de Pareto	30
2.1.3.	Conjuntos não-dominados e Fronteira Pareto Ótima.....	31
2.1.4.	Busca e Decisão.....	31
2.2.	OTIMIZAÇÃO EVOLUCIONÁRIA MULTIOBJETIVO.....	33
2.2.1.	Algoritmo Genético de Classificação Não Dominado III	33
2.2.2.	Algoritmo Evolutivo Multiobjetivo Baseado em Decomposição	35
2.3.	OTIMIZAÇÃO MULTIOBJETIVO POR INTELIGÊNCIA DE ENXAME	36
2.3.1.	Otimização Partícula de Enxames Multiobjetiva	37
2.3.2.	Otimização de Lobos Cinzentos Multiobjetivo	38
2.3.3.	Algoritmo de Enxame Salpa Multiobjetivo	40
3.	CRITÉRIOS DE DESEMPENHO UTILIZADOS NA PESQUISA.....	42
3.1.	MÉTRICAS DE QUALIDADE PARA PROBLEMAS MULTIOBJETIVOS E MUITOS OBJETIVOS	43
3.2.	METAHEURÍSTICAS MAIS UTILIZADAS	45
3.3.	ÍNDICES DE DESEMPENHO.....	46
3.4.	MÉTODOS DE COMPARAÇÃO DE METAHEURÍSTICAS.....	48
3.5.	FUNÇÕES DE <i>BENCHMARK</i>	51
4.	METAHEURÍSTICAS PROPOSTAS.....	64
4.1.	ALGORITMO DO LEÃO MULTIOBJETIVO	66
4.1.1.	Inicialização	69
4.1.2.	Geração da população residente e nômade.....	69
4.1.3.	Geração de pontos de referência	70

4.1.4.	Avaliação de fertilidade.....	71
4.1.5.	Acasalamento	72
4.1.5.1.	Cruzamento	72
4.1.5.2.	Mutação.....	73
4.1.6.	Defesa territorial	73
4.1.6.1.	Crescimento dos filhotes	73
4.1.6.2.	Luta de sobrevivência.....	74
4.1.7.	Aquisição territorial	74
4.1.7.1.	Criação da população auxiliar – Filhotes maduros	74
4.1.8.	Checando não dominância	75
4.1.9.	Associando cada indivíduo a um ponto de referência	75
4.2.	ALGORITMO DO LEÃO COM BUSCA LOCAL MULTIOBJETIVO.....	78
4.2.1.	Inicialização	81
4.2.2.	Geração da população residente e nômade.....	81
4.2.3.	Geração dos vetores de referência.....	81
4.2.4.	Processamento Paralelo LA e ILS.....	83
4.2.4.1.	Etapas do algoritmo LA	83
4.2.4.2.	Etapas do algoritmo ILS	83
4.2.4.2.1.	Memória de soluções.....	83
4.2.4.2.3.	Busca local.....	84
4.2.4.2.4.	Checagem não dominância	84
4.2.5.	Seleção Guiada por Vetor de Referência	84
4.2.6.	Adaptação do vetor de referência.....	86
5.	RESULTADOS OBTIDOS.....	88
5.1.	FUNÇÕES DE <i>BENCHMARK</i>	88
5.1.1.	Parâmetros de configuração.....	89
5.1.2.	Avaliação de desempenho	91
5.1.2.1.	Distância geracional invertida (IGD)	91
5.1.2.2.	Métrica Hipervolume (HV)	98
5.1.2.3.	Análise gráfica das respostas do MOLA e MOLAILS	105
5.1.3.	Ambiente experimental e tempo de computação.....	107
5.2.	MODELOS EM ENGENHARIA.....	109
5.2.1.	Estudo de caso 1: Modelagem da malha hidráulica e turbina hidráulica de uma usina elétrica.	110

5.2.1.1. Estimativa de parâmetros do HTRS	113
5.2.2. Estudo de Caso 2: Interação Fluido-Estrutura com um Grau de Liberdade	120
5.2.2.1. Estimativa do Problema Direto da Interação Fluido-Estrutura	122
5.2.3. Experimentos e análise de resultados da Interação Fluido-Estrutura	123
6. CONCLUSÃO.....	128
REFERÊNCIAS.....	130

1. INTRODUÇÃO

Nas últimas décadas o avanço computacional e a diversidade de métodos numéricos oriundos da matemática e ciência da computação possibilitaram a aplicação da otimização multiobjectivo em uma gama enorme de problemas em diversas áreas do conhecimento. Alguns desses problemas são: geração de energia (BERGAMINI et al., 2023), energia renovável (ABDELSATTAR et al., 2024), refino de petróleo e produção de hidrogênio (QIN et al., 2022).

Otimizar é a tarefa de encontrar uma ou mais soluções que corresponde a minimizar (ou maximizar) um ou mais objetivos especificados e que satisfaçam todas as restrições (se houver). Um problema de otimização de objetivo único envolve apenas uma função objetivo e geralmente resulta em uma única solução. Naturalmente, quando há apenas uma função a ser otimizada, a obtenção dos resultados costuma ser mais simples. Por outro lado, um problema de otimização que envolve múltiplos objetivos que devem ser otimizados ao mesmo tempo e que podem apresentar uma natureza conflitante entre si, é chamado de problema de otimização multiobjetivo (*Multi-objective Optimization Problems* - MOPs). Uma classe especial de MOPs são os problemas com quatro ou mais objetivos que são conhecidos na literatura como problemas de otimização com muitos objetivos (*Many-Objective Optimization Problems* - MaOPs) (CHENG et al., 2018).

Os problemas de otimização matemática geralmente são caracterizados por três elementos fundamentais. O primeiro é uma quantidade numérica, definida como função objetivo, usada para avaliar soluções candidatas. Em muitos problemas de otimização, há apenas uma função objetivo, e encontrar a melhor solução candidata é equivalente a encontrar o elemento com o melhor valor possível da função objetivo, intitulado como solução "ótima". A função objetivo representa um valor que se deseja minimizar (ou maximizar, dependendo do contexto). O segundo elemento é uma coleção de variáveis de decisão, que são quantidades que podem ser manipuladas para otimizar as funções objetivo. O terceiro elemento são as restrições, que são condições que devem ser sempre verdadeiras, não importa qual seja a solução, impondo restrições aos valores que as variáveis de decisão podem assumir (ABRAHAM et al., 2005; WRIGHT, 2016).

A formulação matemática de um problema de otimização é o primeiro passo para sua resolução. Corresponde à tradução do problema em equações e

inequações que representam seus três elementos: as funções objetivo, as variáveis de decisão e as restrições. Os métodos que serão aplicados para resolver o problema de otimização dependerão de sua formulação matemática. Uma formulação incompleta ou incorreta levará o método a produzir soluções inviáveis ou nenhuma solução.

Comum a quaisquer problemas de otimização multiobjetivo tem-se a não comensurabilidade e o conflito entre objetivos. Conseqüentemente, em tais problemas não existe uma solução única e sim um conjunto de soluções factíveis que podem otimizar todos os objetivos simultaneamente. Estas soluções são denominadas soluções eficientes ou soluções Pareto ótimas. Este conjunto é constituído de soluções não-dominadas, ou seja, soluções em que não existem outras melhores considerando os valores para os objetivos (FATHY; ABDELAZIZ, 2018).

Basicamente, existem três tipos de abordagens para resolver problemas multiobjetivos: *Métodos A Posteriori*: inicialmente é realizada a busca por soluções Pareto ótimas, em seguida o processo de decisão é realizado, no qual é escolhida a solução; *Métodos Progressivos/Iterativos*: caracterizam-se por eliminar a separação entre as etapas de busca e de decisão durante o processo de otimização. O processo de busca é guiado para regiões potencialmente promissoras, e o decisor direciona a busca das novas soluções eficientes; *Métodos A Priori*: há a participação do decisor antes da inicialização do processo de otimização, onde o decisor atribui ponderações aos critérios ou pelo menos hierarquiza os m objetivos. O problema, inicialmente com m objetivos, é transformado em outro, resultando em m problemas com um único objetivo (WONG; MING, 2019).

Com isto, várias metodologias têm sido empregadas para a otimização multiobjetivo. Vários métodos clássicos/exatos, como programação linear, método da soma ponderada e método de programação de metas, etc., foram propostos por pesquisadores desde 1950. Esses métodos clássicos têm algumas deficiências para resolver problemas de otimização: ficar preso no ótimo local; problemas resultantes da estrutura dos métodos clássicos; e problema de alto tempo computacional. Para superar essas questões, os pesquisadores têm investigado em diferentes campos (COLLETTE; SIARRY, 2003).

As últimas décadas testemunharam uma mudança profunda nas tecnologias de busca e otimização. Problemas de otimização em que a complexidade de se obter soluções considerando as técnicas exatas são inacessíveis, a utilização das

metaheurísticas tem ganhado popularidade e uso frequente. Atualmente, essas técnicas são consideradas métodos de ponta para solucionar problemas de otimização difíceis. Assim, sempre que surgem novos problemas de otimização, as metaheurísticas são uma das principais ferramentas no rol de soluções (ANERJEE et al., 2024).

Metaheurísticas são paradigmas da inteligência computacional, usadas especialmente para resolver problemas de otimização difíceis. São procedimentos que orientam e modificam as operações de heurísticas subordinadas, para produzir soluções aproximadas e aprimoradas para problemas de otimização com relação a algoritmos mais simples. Suas soluções são encontradas com maior precisão e confiabilidade do que as heurísticas devido a uma característica importante das metaheurísticas: processos estocásticos. Os processos estocásticos utilizam-se de operadores aleatórios que realizam a busca de soluções de modo aleatório no espaço de soluções. Esta característica se torna vantajosa quando comparada entre as metaheurísticas e as heurísticas, tornando-as mais ágeis (em questão de tempo computacional) para se obter uma solução (BOUSSAÏD et al., 2013; GENDREAU; POTVIN, 2010).

As metaheurísticas também vêm demonstrando agilidade e competência ao serem aplicadas para tratar de problemas de tempo polinomial não determinístico (NP - *Non-Deterministic Polynomial time*), ou seja, o tamanho das instâncias influencia o tempo de execução dos algoritmos. Quando se trata de problemas dinâmicos, está se lidando com problemas NP-difíceis devido à complexidade que os sistemas dinâmicos trazem ao reproduzir seu comportamento físico (MARROUCHE; CHLEBIKOVA, 2024).

Resolver problemas do mundo real geralmente envolve problemas multiobjetivo desafiadores e NP-difíceis que requerem técnicas de otimização, sem garantia de obtenção de uma solução ótima. Não há algoritmos eficazes e eficientes para todos os problemas NP-difíceis, portanto, a experimentação com vários algoritmos de otimização é necessária para encontrar o algoritmo que produz a melhor solução (TOMAR et al., 2024).

Diversos autores, ao longo dos anos, aproveitando o sucesso de metaheurísticas para resolução de problemas de otimização mono-objetivo e multiobjetivo, desenvolveram diferentes metaheurísticas inspiradas em algum fenômeno do “mundo real”, a Tabela 1 ilustra alguns trabalhos científicos.

Tabela 1 – Linha do tempo das principais propostas metaheurísticas.

Ano	Metodologia	Citação
Baseadas na evolução das espécies		
1985	<i>VEGA – Vector Evaluated Genetic Algorithms</i>	Schaffer (1985)
1993	<i>MOGA – Multi-Objective Genetic Algorithm</i>	Fonseca; Fleming (1993)
1993	<i>NPGA – Niche Pareto Genetic Algorithm</i>	Horn et al. (1993)
2000	<i>DEMO – Differential Evolution for Multi-objective Optimization</i>	O’Hagan et al. (2000)
2001	<i>SPEA II – Strength Pareto Evolutionary Algorithm II</i>	Zitzler et al. (2001)
2003	<i>MODE – Multi-objective Differential Evolution</i>	Babu; Jehan (2003)
2004	<i>IBEA – Indicator-Based Selection in Multi-objective Search</i>	Zitzler; Künzli (2004)
2014	<i>NSGA III – Non-dominated Sorting Genetic Algorithm III</i>	Deb et al. (2014)
2007	<i>MOEA/D – Multi-objective Evolutionary Algorithm based on Decomposition</i>	Zhang; Li (2007)
2008	<i>MOJADE – Self-Adaptive Multi Objective Differential Evolution</i>	Zhang; Sanderson (2008)
2017	<i>RVEA – Reference Vector Guided Evolutionary Algorithm</i>	Cheng (2017)
Baseadas na vida social de enxames/animais		
2002	<i>MOPSO – Multiple Objective Particle Swarm Optimization</i>	Coello Coello; Lechuga, (2002)
2011	<i>MOBA – Multi-objective Bat Algorithm</i>	Yang (2011)
2012	<i>MOABC – Multi-objective artificial bee colony algorithm</i>	Akbari et al. (2012)
2012	<i>MOCSSO – Multi-objective Cat Swarm Optimization</i>	Pradhan; Panda (2012)
2013	<i>MOCS – Multi-objective Optimization Cuckoo Search</i>	Yang; Deb (2013)
2015	<i>MOACO – Multi-objective Ant Colony Optimization</i>	Shi; Kong (2015)
2016	<i>MODA – Multi-objective Dragonfly Algorithm</i>	Mirjalili (2016)
2016	<i>MOGWO – Multi-objective Grey Wolf Optimizer</i>	Mirjalili et al. (2016)
2017	<i>MOALO – Multi-objective Ant Lion Optimizer</i>	Mirjalili et al. (2017)
2017	<i>MOWOA – Multi-objective whale optimization</i>	Kumawat et al. (2017)
2017	<i>MSSA – Multi-objective Salp Swarm Algorithm</i>	Mirjalili et al. (2017)
2018	<i>MOAAA – Multi-objective Artificial Algae Algorithm</i>	Babalik et al. (2018)
2018	<i>MOBFOA – Multi-objective Bacteria Foraging Optimization Algorithm</i>	Kaur; Kadam (2018)
2019	<i>MOCOA – Multi-objective Optimziation Coyote Algorithm</i>	Pierezan et al. (2019)
2019	<i>MDEHO – Multi-objective and Discrete Elephant</i>	Sadouki; Tari (2019)

Herding Optimization

2023 *MOLA – Multi-objective Optimziation Lion Algorithm* Bergamini et al. (2023)

Baseadas em Fenômeno físico/químico

1992 *AMOSAs – Multi-objective Simulated Annealing* Lee; wang (1992)

2010 *MOGSA – Multi-Objective Optimization Gravitational Search Algorithm* (Hassanzadeh; Rouhani (2010)

2016 *MOMVO – Multi-objective Multi-Verse Optimizer* Mirjalili et al. (2016)

2017 *MOVS – Multi-objective Vortex Search Algorithm* Özkiş; Babalik (2017)

2019 *MOLAPO – Multi-objective Lightning Attachment Procedure Optimization* Nematollahi (2019)

2019 *MOCSS – Multi-objective Charged System Search* Ranjbar et al. (2019)

Baseadas em comportamento humano

1997 *MOTS – Multi-objective Tabu Search* (Hansen, 1997)

2004 *MOVNS – Multi-objective Optimization Variable Neighbourhood Search* (Geiger, 2004),

2004 *MOGRASP – Multi-objective Greedy Randomized Adaptive Search Procedure* (Vianna; Arroyo, 2004),

2011 *MOHS – Multi-objctive Optimization Harmony Search* (Sivasubramani; Swarup, 2011)

2019 *MOILS – Multi-objective Optimization Iterated Local Search* (Schulz et al., 2019).

Com o aumento da complexidade dos problemas de engenharia que possuem mais de três objetivos a serem otimizados, estes tem chamado a atenção da comunidade científica pelas dificuldades encontradas com os algoritmos disponíveis na literatura para resolvê-los de forma adequada (YADAV et al., 2023).

Com isso, metaheurísticas passaram por adaptações ou modificações para melhoraram sua convergência. Nos últimos anos, está relatado na literartura um aumento significativo de algoritmos que não seguem puramente o paradigma de uma metaheurística tradicional única. Pelo contrário, combinam vários componentes algorítmicos, muitas vezes originários de algoritmos de outras áreas de pesquisa sobre otimização. A evolução das metaheurísticas tem sido constante ao longo dos anos e com isto diversas combinações têm surgido, como as metaheurísticas híbridas (TALBI, 2015).

A principal motivação por trás da hibridização de diferentes algoritmos é explorar o caráter complementar de diferentes estratégias de otimização, ou seja,

acredita-se que os algoritmos híbridos se beneficiem da sinergia. Na verdade, a escolha de uma combinação adequada de conceitos algorítmicos complementares pode ser a chave para obter o melhor desempenho na solução de muitos problemas de otimização difíceis. Infelizmente, o desenvolvimento de uma abordagem híbrida eficaz é, em geral, uma tarefa difícil que requer experiência de diferentes áreas de otimização. Além disso, a literatura mostra que não é trivial generalizar, ou seja, um determinado híbrido pode funcionar bem para problemas específicos, mas pode funcionar mal para outros.

De acordo com Lozano e García-Martínez (2010), o projeto de metaheurísticas híbridas utilizando algoritmos de busca especializados em busca local e busca global possibilita obter uma meta-heurística que compartilhe as características de intensificação e diversificação do espaço de busca. Alguns algoritmos desta categoria podem ser vistos na Tabela 2.

Tabela 2 – Linha do tempo das principais propostas metaheurísticas híbridas.

Ano	Metodologia	Citação
1998	<i>Multi-objective Optimization Genetic Algorithm Search Local</i>	Ishibuchi; Murata (1998)
2006	<i>Multi-objective Hybrid Tabu Scatter Search</i>	Beausoleil (2006)
2007	<i>Variable Neighborhood Particle Swarm Optimization for Multi-objective</i>	Liu et al. (2007)
2009	<i>Multi-objective Multi-State Hybrid Differential Evolution Algorithm</i>	Zeng et al. (2009)
2012	<i>Multi-objective hybrid evolutionary algorithms for radial basis function neural network design</i>	Qasem et al. (2012)
2017	<i>Hybrid Multi-objective Whale Optimization Algorithm Elman Neural Network</i>	Wang et al. (2017)
2019	<i>Evolutionary Algorithm Variable Neighborhood Search</i>	Sun et al. (2019)

Já no que se diz a respeito das metaheurísticas híbridas cooperativas, Feng et al. (2021) explicam estas metaheurísticas como duas ou mais metaheurísticas que trocam informações entre si, ou seja, as metaheurísticas trocam as melhores

soluções com isto os algoritmos podem evoluir mais rapidamente para a melhor solução sem que haja convergência precoce.

Tezel e Mert (2021) descrevem as metaheurísticas híbridas cooperativas como algoritmos de diferentes categorias que trocam informações sobre estados, modelos, subproblemas, soluções ou características do espaço de pesquisa. Assim, essa troca de informação resulta em soluções melhores ao comparar com as soluções das metaheurísticas em suas versões clássicas.

Vários trabalhos científicos (GUPTA et al., 2021; KISENGEU et al., 2020; NAGHAVIPOUR et al., 2022; e WU et al., 2022) mostraram que algoritmos heterogêneos nos quais diferentes métodos metaheurísticos são empregados garantem melhor soluções. Além disso, esses estudos revelaram que a combinação de vários métodos metaheurísticos, cada um usando uma estratégia diferente, melhora a robustez da pesquisa global. Esses estudos também apontam que, embora as estratégias de busca não relacionadas sejam fáceis de aplicar e forneçam bons resultados candidatos, o desempenho dessas estratégias pode ser melhorado por meio da cooperação.

Nesta tese são propostas duas novas metaheurísticas multiobjetivo: a primeira metaheurística multiobjetivo denominada Algoritmo de Otimização de Leão Multiobjetivo (do termo em inglês, *Multi-objective Optimization Lion Algorithm – MOLA*), baseada na metaheurística mono objetivo Otimização de Leão e na utilização dos conceitos de dominância de Pareto e pontos de referência para a seleção da população em cada iteração; a segunda metaheurística desenvolvida denominada Algoritmo de Otimização de Leão com Busca Local Multiobjetivo (do termo em inglês, *Multi-objective Optimization Lion Algorithm Iterated Search Local – MOLAILS*), utiliza o Algoritmo do Leão e a Busca Local Iterativa trabalhando de forma cooperativa e o conceito de vetores suporte, o qual será utilizado para a seleção da população em cada iteração. Os algoritmos são aplicados a um conjunto de funções de teste e seus resultados são comparados com os de outros algoritmos de otimização. Por fim, os algoritmos propostos são utilizados em problemas da engenharia.

1.1. MOTIVAÇÃO

Com o avanço da complexidade dos problemas de engenharia, e conforme a análise em diversos trabalhos envolvendo o uso de metaheurísticas multiobjetivos (clássicas e híbridas) em problemas do tipo NP, elas têm-se mostrado técnicas promissoras e eficientes quando se trata de analisar o comportamento dinâmico dos subsistemas conflitantes destes sistemas.

A busca por novos métodos para a otimização desses problemas tem levado os pesquisadores a proporem novas técnicas ou a combinação de técnicas existentes. E segundo o teorema *No Free Lunch* proposto por Wolpert e Macready (1997), para qualquer algoritmo de otimização qualquer melhoria no desempenho em uma classe de problemas é compensada por um desempenho inferior em outra classe, ou seja, não existe um algoritmo ótimo universal para todos os problemas de otimização. Baseado neste paradigma o presente trabalho propõe a construção de duas metaheurísticas multiobjetivo: uma a partir da metaheurística mono objetiva Algoritmo do Leão (RAJAKUMAR, 2012) utilizando não dominância e pontos de referência; a outra a partir de duas metaheurísticas mono objetivo Algoritmo do Leão (RAJAKUMAR, 2012) e Busca Local Iterativa (LOURENÇO et al., 2003) trabalhando de forma cooperativa e utilizando vetores de referência. Essas novas metaheurísticas multiobjetivo tem como objetivo melhorar a performance em termos de convergência e diversidade das soluções utilizando métricas como IGD e Hipervolume, e com tempo computacional menor quando comparadas com outras metaheurísticas da literatura.

1.2. OBJETIVOS

O objetivo deste trabalho é propor duas metaheurísticas multiobjetivo. A primeira baseada na versão mono objetivo da metaheurística Algoritmo do Leão utilizando não dominância e pontos de referência. A segunda será baseada na metaheurística híbrida cooperativa entre Algoritmo do Leão e Busca Local Iterativa utilizando vetores de referência. Essas metaheurísticas então serão utilizadas para obter modelos da engenharia.

1.2.1. Objetivos Específicos

- Implementar computacionalmente o algoritmo Multiobjetivo Otimização de Leão baseado respectivamente em seu algoritmo mono objetivo;
- Desenvolver computacionalmente o algoritmo multiobjetivo Otimização de Leão com Busca Local Iterativa que será utilizado no procedimento híbrido cooperativo
- Submeter os algoritmos multiobjetivos propostos em funções de *benchmark* disponibilizado na literatura;
- Avaliar o desempenho dos algoritmos propostos com um conjunto de algoritmos selecionados da literatura, comparando gráficos de convergência e utilizando medidas e inferência não paramétricas;
- Aplicar os algoritmos propostos em problemas de modelagem na área da engenharia.

1.3. CONTRIBUIÇÕES DA TESE

Baseado nos objetivos desta tese levanta-se as seguintes contribuições:

- i) a nova metaheurística multiobjetivo derivada de sua versão mono objetivo terá desempenho superior ou similar as metaheurísticas multiobjetivo presentes na literatura;
- ii) a nova metaheurística multiobjetivo híbrida cooperativa será capaz de ter um melhor desempenho do que as metaheurísticas multiobjetivo que a compõem.

Principais contribuições:

- i) proposição da metaheurística multiobjetivo denominada Algoritmo de Otimização de Leão Multiobjetivo (do termo em inglês, *Multi-objective Optimization Lion Algorithm – MOLA*);
- ii) proposição da metaheurística multiobjetivo híbrida cooperativa denominada Algoritmo de Otimização de Leão com Busca Local Multiobjetivo (do termo em inglês, *Multi-objective Optimization Lion Algorithm Iterated Search Local – MOLAILS*).

1.4. JUSTIFICATIVA

A área de metaheurísticas cresceu nas últimas duas décadas como uma solução para problemas de otimização do mundo real. Para problemas de otimização complexos, as técnicas metaheurísticas multiobjetivas são capazes de gerar soluções de boa qualidade em relativamente muito menos tempo do que as técnicas de otimização tradicionais. Além disso com o avanço da complexidade de problemas reais (elétrica, biomedicina, civil por exemplo) envolvem múltiplos objetivos que são conflitantes. A principal motivação de se utilizar as metaheurísticas multiobjetivo é a questão da flexibilidade e a cooperação, por serem consideradas de fácil implementação e fácil adaptação para se lidar com diferentes tipos de problemas de otimização para sistemas reais. Neste contexto, a utilização de técnicas de otimização cooperativa vem demonstrando ser algoritmos promissores ao encontrar soluções de Pareto, pois quando trocam informações aumentam a confiabilidade e melhoram as soluções. Posto isto a metaheurística *MOLA* é inédita, onde versões já existentes na literatura trabalham com outras versões (tipo de comportamento do leão por exemplo), assim como os operadores utilizados para a seleção da população para a próxima geração também são diferentes da utilizada na metaheurística proposta.

Quanto à metaheurística híbrida cooperativa multiobjetivo *MOLA/ILS*, tal combinação possui como objetivo obter uma metaheurística que explore e una as potencialidades de cada uma das metaheurísticas originais aplicadas individualmente a determinado problema. A metaheurística *MOLA/ILS* também é inédita, pois as populações geradas pelo LA (*Lion Algorithm*) são refinadas pelo ILS (*Iterated Local Search*) e para manter um tempo de processamento viável é utilizado como operador de seleção vetores de referência.

1.5. ESTRUTURA DA TESE

O presente trabalho está dividido em 6 capítulos. No Capítulo 1 faz-se uma introdução contendo a contextualização de metaheurísticas multiobjetivo clássicas e metaheurísticas híbridas. No Capítulo 2 tem-se uma descrição sobre otimização multiobjetivo e das metaheurísticas multiobjetivo existentes na literatura explanando sobre aquelas que serão utilizadas neste trabalho. No Capítulo 3 são apresentadas as métricas de qualidade mais utilizadas, as metaheurísticas mais utilizadas, os

métodos de comparação que vem sendo utilizado com maior frequência e quais as funções de *benchmark* mais apontadas na literatura. No Capítulo 4 são apresentadas as metaheurísticas propostas, com sua formulação matemática e a estratégia utilizada na construção das mesmas. No Capítulo 5 as metaheurísticas propostas são comparadas com metaheurísticas clássicas da literatura utilizando funções testes, a seguir as metaheurísticas *MOLA* e *MOLA/LS* são utilizadas nos estudos de caso reais. Por fim, no Capítulo 6 destacam-se as conclusões finais e os trabalhos futuros.

2. OTIMIZAÇÃO MULTIOBJETIVO UTILIZANDO METAHEURÍSTICAS

Em otimização multiobjetivo, em geral, não existem soluções ótimas no sentido de minimizarem/maximizarem individualmente todos os objetivos. A principal característica na otimização multiobjetivo é a existência de um conjunto de soluções eficientes/aceitáveis que são superiores às demais. Encontrar todas essas soluções eficientes é geralmente uma tarefa inviável, especialmente em problemas com muitos objetivos (Palakonda; Mallipeddi, 2017). Na prática, os otimizadores tentam entregar um conjunto com a maior quantidade e diversidade de soluções eficientes. Quando encontrar tais soluções é impraticável, os otimizadores tentam entregar uma boa aproximação dessas soluções. Tal conjunto de soluções é chamado de aproximação do conjunto de Pareto, ou conjunto de aproximação, e o conjunto de seus respectivos pontos no espaço objetivo é denominado aproximação da fronteira de Pareto.

Existem muitos métodos de otimização e cada um deles alcança bons resultados em determinados tipos de problema. A escolha do método depende de uma série de características do problema a ser otimizado, principalmente do comportamento da função que o representa, a qual muitas vezes é de difícil determinação. Para esta escolha, faz-se necessário também um bom conhecimento das ferramentas de otimização.

Algoritmos metaheurísticas são paradigmas da inteligência computacional usados especialmente para resolver problemas de otimização difíceis. Em geral, os algoritmos computacionais inteligentes visam gerar uma nova solução superior à existente a cada iteração. Idealmente, espera-se que algoritmos computacionais inteligentes gerem soluções superiores às soluções atuais com mínimo esforço.

Neste capítulo são apresentados os conceitos básicos sobre os temas utilizados neste trabalho. Nas próximas seções apresenta-se a teoria que envolve problemas multiobjetivos e as metaheurísticas como métodos de otimização.

2.1. OTIMIZAÇÃO MULTIOBJETIVA

Otimização multiobjetivo refere-se à otimização simultânea de múltiplos objetivos conflitantes. Nestes casos a solução do problema envolve não apenas

uma única solução ótima, mas um conjunto de soluções de Pareto ótimas ou soluções não dominadas, a fim de alcançar um benefício predefinido (seja ele por minimização ou maximização).

As soluções Pareto ótimas têm a característica de serem não dominadas, isto é, todas as soluções podem ser consideradas equivalentes não sendo possível ordená-las segundo algum critério de qualidade (Cuervas et al., 2018).

Com a evolução dos problemas de engenharia e ciência surgiram problemas com mais de três objetivos, esses problemas foram definidos como pertencentes à otimização de muitos objetivos (do termo em inglês *Many-Objective Optimization - MaOPs*) uma classe dos problemas multiobjetivos (do termo em inglês, *Multi-objective Optimization Problem – MOP*). Entretanto, a relação de dominância de Pareto que é considerado um dos critérios mais significativos na otimização multiobjetivo para esta classe de problemas resulta num baixo desempenho das metaheurísticas tradicionais baseados em Pareto. Além disso, pode ocorrer o comprometimento entre a convergência e diversidade das soluções de Pareto, juntamente com o aumento de tempo computacional ou espaço de busca (Cheng et al., 2018).

Problemas de otimização multiobjetivo de uma forma geral pode ser formulados pela Equação (2.1)

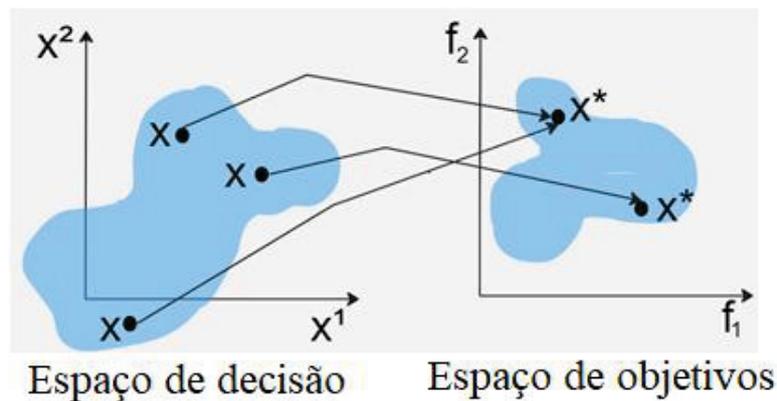
$$\begin{aligned}
 \min f(x) &= [f_1(x), f_2(x), \dots, f_o(x)] \\
 &\text{sujeito a:} \\
 g_i(x) &\geq 0, \quad i = 1, 2, \dots, m \\
 h_i(x) &= 0, \quad i = 1, 2, \dots, p \\
 x &= [x_1, x_2, \dots, x_n] \in \Omega \\
 L_i &\leq x_i \leq U_i, \quad i = 1, 2, \dots, n
 \end{aligned} \tag{2.1}$$

onde n é o número de variáveis de decisão, o é o número de funções objetivo, m é o número de restrições de desigualdade, p é o número de restrições de igualdade. Sendo g_i a i -ésima restrição de desigualdade, h_i indica a i -ésima restrição de igualdade, e x o vetor das variáveis de decisão, pertencente a região do espaço $\Omega \subseteq \mathbb{R}^n$, restrita pelos limites mínimos L_i e máximos U_i de cada variável x_i .

Em MOPs, há dois espaços de busca a serem considerados: o espaço de decisão, que contém as variáveis de decisão, e o espaço objetivo, apresentando os valores das funções objetivo correspondentes. Um ponto no espaço de decisão tem apenas um ponto correspondente no espaço objetivo e diversos pontos no espaço de decisão podem ser mapeados para o mesmo ponto no espaço objetivo. A Figura

1 ilustra essa ideia para espaços bidimensionais. Observe que não é obrigatório que ambos os espaços tenham a mesma dimensão. A correspondência entre os pontos nos diferentes espaços é representada pelas setas. O conjunto viável é delimitado pela área azul. O processo de busca de otimização ocorre no espaço de decisão, por meio da manipulação das variáveis de decisão, mas a análise da qualidade relativa das soluções é realizada no espaço objetivo.

Figura 1 – Exemplo de Mapeamento de Pontos entre ds Espaços de Decisão e Objetivos.



FONTE: Autora, 2024.

2.1.1. Dominância de Pareto

Uma solução x^* para um problema de otimização multiobjetivo é considerada eficiente, ótima de Pareto ou não dominada se e somente se não houver outra solução viável x para o problema, tal que x domine x^* . Há o interesse em encontrar o maior número possível de soluções não dominadas, uma vez que elas apresentam os melhores *trade-offs* disponíveis entre os valores das funções objetivo.

Os conceitos de comparação de duas soluções em problemas multiobjetivo foram inicialmente propostos por Francis Ysidro Edgeworth em 1881, e depois estendidos por Vilfredo Pareto em 1896.

Em otimização multiobjetivo, uma das maneiras de representar o conjunto ótimo de soluções é baseada na relação de dominância de Pareto. A definição de dominância de Pareto surgiu da percepção de que uma solução é melhor que outra, se a primeira for melhor em pelo menos um objetivo e não for pior em nenhum outro

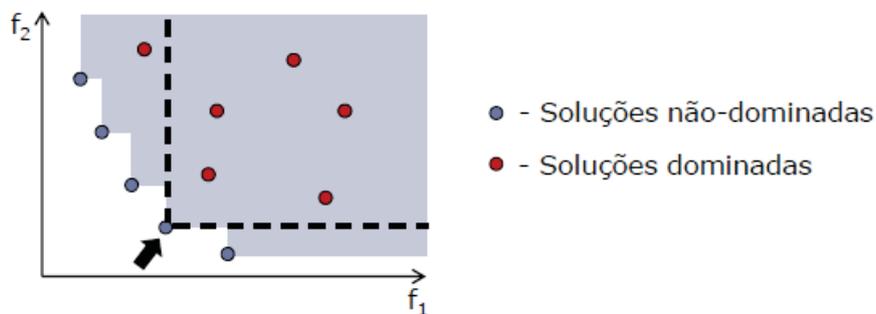
objetivo (Nassef et al., 2023). Sem perda de generalidade, a definição matemática da dominância de Pareto é dada pela Equação (2.2) (Coello, 2009).

$$x^* \text{ domina } x, \quad \text{se } f_i(x^*) < f_i(x) \text{ e } f_j(x^*) \leq f_j(x) \quad (2.2)$$

Se uma solução x^* não domina x e nem x domina x^* estas soluções são ditas indiferentes ou que possuem o mesmo grau de dominância.

A Figura 2 mostra um exemplo de dominância em relação a uma determinada solução.

Figura 2 – Exemplo do conceito de dominância.



Fonte Autora, 2024.

É importante observar que apesar da dominância ser definida no espaço das variáveis de decisão (x^* e x) a comparação entre as duas soluções é realizada no espaço das funções objetivo, ou seja, o critério de dominância é verificado utilizando-se os resultados da avaliação de cada um dos objetivos.

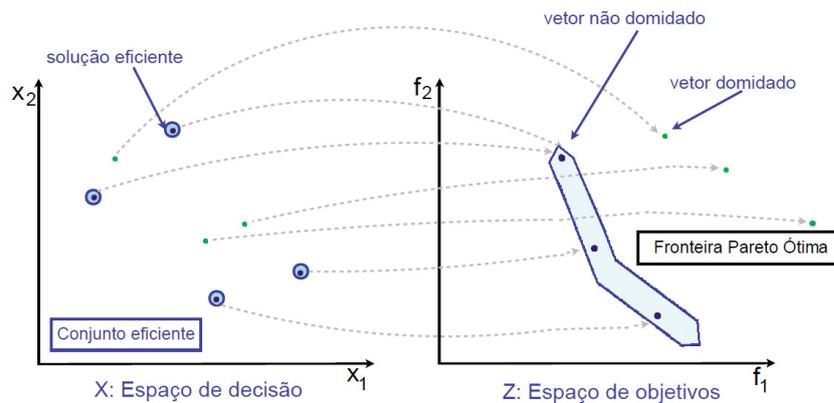
2.1.2. Otimalidade de Pareto

Uma solução $x^* \in R^n$ será considerada eficiente ou Pareto ótima, se somente se, não existir outra solução x que domine x^* . Em problemas multiobjetivo procura-se por soluções de Pareto ótimas. Neste caso, consistindo em um conjunto de soluções eficientes ou Pareto ótimas e não de apenas uma solução. Este conjunto de soluções eficientes ou Pareto ótimas é denominado "Conjunto Pareto Ótimo".

2.1.3. Conjuntos não-dominados e Fronteira Pareto Ótima

O conjunto contendo os valores das funções objetivos correspondentes às soluções eficientes ou Pareto ótimas é chamado de fronteira Pareto ótima. Na Figura 3, mostra-se um exemplo de fronteira Pareto ótima e o conceito de dominância. As soluções pertencentes a esta fronteira são as soluções Pareto ótimas. Note que estas soluções são indiferentes umas as outras.

Figura 3 – Fronteira de Pareto ótima e conceito de dominância.



Fonte: Autora, 2024.

2.1.4. Busca e Decisão

Quando se consideram problemas de otimização multiobjetivo, qualquer método sempre se restringe a encontrar um conjunto de soluções eficientes ou Pareto ótimas. O método, nada mais faz do que sugerir um conjunto de soluções ao decisor que deverá ponderar os aspectos que julgue mais importantes com relação aos objetivos conflitantes.

Assim, identificam-se duas etapas básicas: a busca de soluções eficientes ou Pareto ótimas (responsabilidade do método) e a tomada de decisões (responsabilidade do decisor).

Etapa 1: corresponde à busca de soluções na região factível em direção à fronteira Pareto ótima, tarefa essa que pode se tornar penosa devido ao tamanho e complexidade do espaço de busca.

Etapa 2: corresponde à escolha de um critério para determinar qual das várias soluções eficientes deverão ser eleitas para ser utilizada. Para tal ação, o projetista/engenheiro toma uma decisão externa ao processo de otimização.

Dependendo de como e quando o processo de otimização e a etapa de decisão são combinados, os métodos de resolução podem ser classificados em três categorias (Petchrompo et al., 2022).

i) Métodos a priori

Métodos desse tipo caracterizam-se pela participação do decisor antes de resolver o problema, ou seja, já se tem de antemão alguma informação sobre o tipo de solução mais adequada ao problema, então a busca é direcionada para encontrar este tipo de soluções (**decisão → busca**).

Há duas abordagens comumente empregadas:

1. Combinação: essa abordagem emprega a combinação dos objetivos envolvidos no problema de forma a deixá-lo como um problema mono-objetivo. Para tanto, é preciso determinar a importância de cada objetivo através da atribuição de seus respectivos pesos associados.
2. Decomposição: nesse caso os objetivos são elencados de acordo com as respectivas prioridades. A seguir o problema é resolvido para cada um deles, independentemente. No entanto, para a resolução do segundo objetivo, por exemplo, é imposta uma restrição que se refere ao valor ótimo encontrado no primeiro.

ii) Métodos a posteriori

Para esses métodos, a ordem “natural” entre as etapas de busca e de decisão é mantida, ou seja, é realizada inicialmente a busca por soluções Pareto ótimas, em seguida o processo de decisão, responsável pela escolha da solução mais adequada dentre as soluções do conjunto obtido no processo anterior (**busca → decisão**).

Ou seja, com o conjunto Pareto ótimo definido torna-se possível conhecer o comportamento do problema em relação aos objetivos analisados. Conhecendo-se as relações de dependência entre eles, a escolha final é facilitada.

iii) Métodos iterativos/progressivos

Esses métodos caracterizam-se por eliminar a separação entre as etapas de busca e de decisão. O processo de busca é guiado para regiões potencialmente

promissoras através da determinação de características desejáveis de uma solução. O resultado da consulta ao decisor é utilizado na busca de novas soluções eficientes, ou seja, é feito um direcionamento da busca, durante sua execução, para regiões que contenham soluções mais adequadas (**decisão ↔ busca**).

Nesta abordagem também se faz necessária certa experiência do projetista, já que as escolhas deverão ser tomadas de modo a orientar o processo de otimização a “caminhar” na direção da formação da fronteira Pareto ótima.

2.2. OTIMIZAÇÃO EVOLUCIONÁRIA MULTIOBJETIVO

Nas duas últimas décadas, uma série de algoritmos evolutivos multiobjetivos (MOEAs) foram sugeridos (Bilal et al., 2020; Grid et al., 2021; Von Lücken et al., 2014; Moriyama et al. 2021; Xu et al., 2023; e Zhang et al., 2020). A principal razão para isso é a capacidade desses algoritmos encontrarem várias soluções ótimas de Pareto em uma única simulação.

Algoritmos evolutivos tem como base a evolução darwiniana. Um sistema evolucionário artificial é um sistema computacional baseado na noção de evolução simulada. É composto por uma população de indivíduos de tamanho constante ou variável, um critério de aptidão e operadores geneticamente inspirados que produzem a próxima geração a partir da atual. A população inicial é normalmente gerada aleatoriamente ou heurísticamente, e os operadores típicos são cruzamento, mutação e recombinação. Em cada etapa, os indivíduos são avaliados de acordo com a função de aptidão dada (sobrevivência do mais apto). A próxima geração é obtida de indivíduos selecionados (pais) usando operadores geneticamente inspirados. A escolha dos pais pode ser orientada por um operador de seleção que reflete o princípio biológico da seleção do parceiro. Este processo de evolução simulada eventualmente converge para uma população quase ótima de indivíduos, do ponto de vista da função de aptidão.

Dentre as metaheurísticas da categoria evolução natural, neste trabalho será utilizada os algoritmos NSGA – III e o MOEA/D.

2.2.1. Algoritmo Genético de Classificação Não Dominado III

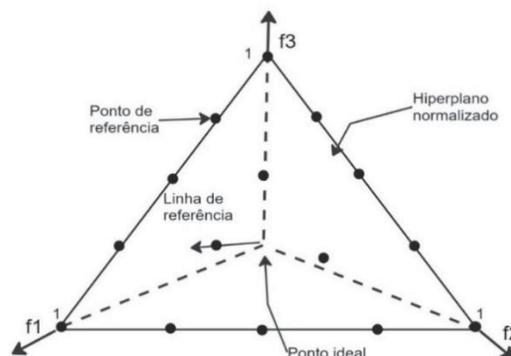
O Algoritmo Genético de Classificação Não-Dominado (do termo em inglês, *Non-Dominated Sorting in Genetic Algorithms – NSGA III*) foi proposto por Deb e

Jain (2014) é um dos algoritmos mais famosos da área de Otimização com Muitos Objetivos e foi também um dos grandes propulsores da utilização da técnica dos pontos de referência no ambiente de MOEAs. Ele estende o princípio de funcionamento de seu antecessor NSGA-II (Deb et al., 2002a), mas com algumas alterações significativas em seu mecanismo de seleção. Ao contrário do NSGA-II, a diversidade entre os membros da população no NSGA-III é obtida por um mecanismo que usa um conjunto de pontos posicionados em um hiperplano normalizado para orientar a pesquisa. Esses pontos (pontos de referência) são utilizados principalmente para garantir a convergência e a diversidade. Isto é feito através do reforço dos membros da população não dominada que estão mais próximos a este conjunto de pontos de referência.

A maior contribuição da metaheurística NSGA-III encontra-se na geração do hiperplano, onde é definido o mecanismo de diversidade baseado em pontos de referência. No fim do loop evolucionário (cruzamento e mutação), as melhores soluções de cada nicho são utilizadas para preencher as posições vagas para a próxima geração. A plataforma básica do NSGA-III é semelhante à do NSGA-II. Mas a diversidade no NSGA-III é garantida fornecendo um conjunto de pontos de referência bem espalhados. O trabalho apresentado por Deb e Jain, também concluiu que o NSGA-III foi capaz de encontrar um conjunto de soluções com boa convergência e diversidade para problemas envolvendo fronteiras que apresentavam características convexas, côncavas, disjuntas, com densidade variável de pontos através da fronteira e que são escalados de forma diferente (Deb; Jain, 2014).

Os pontos de referência do NSGA-III são posicionados em um hiperplano normalizado que é igualmente inclinado a todos os eixos e tem uma intersecção em 1 em cada eixo como pode ser visto na Figura 4.

Figura 4 – Exemplo de um problema de três objetivos e com 15 pontos de referência em um plano de referência normalizado.



Fonte: Adaptada de Das; Dennis, (1998).

2.2.2. Algoritmo Evolutivo Multiobjetivo Baseado em Decomposição

O algoritmo evolutivo multiobjetivo baseado em decomposição (do termo em inglês *Multi-objective Evolutionary Algorithm based on Decomposition* – MOEA/D) proposto por Zhang e Li (2007) faz uso de abordagens de agregação para decompor um MOP em vários subproblemas mono objetivo.

No MOEA/D os múltiplos objetivos são decompostos em subproblemas mono objetivo e a população é dividida em subpopulações, onde cada indivíduo é associado a um dos subproblemas de otimização. Cada subproblema é vinculado a um número de subproblemas vizinhos e são associados a uma única função de agregação e a um vetor de pesos diferente, que contribuem com seu processo de otimização. A função objetivo de cada subproblema é a função de decomposição, modelada de forma que incorpore todos os objetivos do problema original.

A decomposição feita pelo MOEA/D consiste de vetores de pesos diferentes, que podem encontrar pontos distintos na fronteira de Pareto. Desse modo, os diversos subproblemas são otimizados paralelamente, cada um com um vetor de pesos distintos. Uma desvantagem dessa abordagem é que a função de ponderação pode não ser contínua ou diferenciável.

O MOEA/D conta com três estratégias: decomposição, busca de vizinhança e recombinação. Na decomposição o problema é dividido em um conjunto de subproblemas escalares. Já na busca em vizinhança, é realizada a cada geração a troca de soluções entre subproblemas vizinhos, o que visa obter simultaneamente convergência e diversidade das soluções na frente de Pareto. Um subproblema é vizinho quando possui o conjunto de subproblemas mais similares a ele quanto ao desempenho frente aos objetivos. A similaridade pode ser medida usando métricas, como, por exemplo, a distância euclidiana. Em cada geração, uma solução é selecionada de um subproblema com base no critério de seleção, torneio ou roleta.

A solução selecionada é comparada com as soluções em sua vizinhança, e a melhor solução é escolhida como ponto de referência para gerar novas soluções no subproblema atual. Já pela recombinação se gera novas soluções candidatas combinando, por cruzamento das soluções de diferentes subproblemas. São então avaliadas usando seus valores de função objetivo e as soluções não dominadas são adicionadas à população.

Para as funções de agregação, os autores Zhang e Li (2007) propuseram a utilização de três funções de agregação distintas: a *Weighted Sum* (WS), a Tchebycheff (TCH) e a *Penalty-Based Boundary Intersection* (PBI). Essas funções de agregação, em problemas de minimização, cujos valores são menores indicam melhor qualidade da solução.

O método de agregação utilizando a função *Weighted Sum* e dado um conjunto de vetores de pesos W , esta função de escalarização apresenta resultados satisfatórios em um problema de minimização quando a fronteira Pareto é convexa. Na abordagem com a função de escalarização de Tchebycheff, esta função tem como vantagem à abordagem de problemas com fronteiras não-convexas quando comparada com a WS. Finalmente, utilizando a abordagem com a função *Penalty-Based Boundary Intersection* (PBI), esta função mantém a convergência e diversidade da população, enquanto as outras funções diversificam as soluções de acordo com a distribuição dos pesos.

Uma característica positiva do MOEA/D é que como cada subproblema é resolvido com interferência de um número reduzido de vizinhos, este algoritmo tem menor complexidade computacional quando comparado com o NSGA-II. Uma preocupação neste algoritmo é evitar que uma solução substitua indivíduos em todos os seus vizinhos, o que levaria à perda de diversidade e pode tornar difícil aos operadores de evolução gerar soluções melhores.

2.3. OTIMIZAÇÃO MULTIOBJETIVO POR INTELIGÊNCIA DE ENXAME

De forma similar as metaheurísticas multiobjetivo baseados na evolução, nas duas últimas décadas, um número grande de metaheurísticas multiobjetivo baseados no conceito de Inteligência de Enxame, foram também propostas (MOLINA *et al.*, 2020).

Estes algoritmos em geral são bioinspirados e mimetizam o comportamento coletivo proveniente da colaboração e interação entre agentes individuais (por exemplo, bactérias, formigas, cupins, abelhas, aranhas, peixes, pássaros) que se comunicam com outros agentes atuando em seus ambientes locais. A otimização de Inteligência de Enxame, aplica essa ideia ao problema para encontrar uma solução ótima para um determinado problema por meio de uma pesquisa em um espaço de solução (multidimensional). A configuração inicial é um enxame de partículas, cada uma representando uma possível solução para o problema. Cada partícula tem sua

própria velocidade que depende de sua velocidade anterior (o componente de inércia), a tendência para a melhor posição pessoal do passado (o componente nostalgia) e sua tendência para um ótimo de vizinhança global ou ótimo de vizinhança local (o componente social). As partículas, portanto, se movem através de um espaço multidimensional e eventualmente convergem para um ponto entre o melhor global e o melhor pessoal (ANANTHARAJ *et al*, 2015).

Dentre as metaheurísticas da categoria Inteligência de Enxame, neste trabalho serão utilizados os algoritmos MOPSO, MOGWO e MSSA.

2.3.1. Otimização Partícula de Enxames Multiobjetiva

A metaheurística Otimização Enxame de Partículas Multiobjetiva (do termo em inglês, *Multi-objective Particle Swarm Optimization* – MOPSO) foi desenvolvida por Coello Coello; Lechuga (2002). Essa metaheurística é inspirada no comportamento social e cooperativo de organismos biológicos, tais como cardumes e bandos de pássaros.

No MOPSO, cada solução candidata corresponde a uma partícula e o conjunto dessas partículas constitui um enxame de partículas em que ocorre um processo de troca de informação sobre boas regiões no espaço de busca. As partículas se movem no espaço de busca de alta dimensionalidade buscando por boas soluções, usando uma combinação de atração para a melhor solução encontrada até o momento pela própria partícula; e de uma atração para a melhor solução encontrada até o momento pelas partículas consideradas vizinhas. Cada partícula i do enxame S tem uma posição x_i , que consiste em um vetor de m dimensões cujos componentes representam as variáveis de decisão da função objetivo, uma velocidade v_i e uma posição y_i chamada de melhor posição pessoal ou *pbest*, que representa a melhor posição encontrada pela partícula até um momento em consideração.

O MOPSO utiliza o conceito de dominância de Pareto para determinar a direção de voo de uma partícula e mantém vetores não dominados previamente encontrados em um arquivo externo que é posteriormente usado por outras partículas para guiar seu próprio voo. O arquivo externo também tem uma segunda função, pois cada partícula do enxame é guiada pelo *gbest* que é a melhor posição visitada pelo enxame até o momento. No entanto, em problemas multiobjetivos não

existe um *gbest*, mas sim um conjunto de soluções em que não se pode dizer qual solução é a melhor.

Desta forma, o arquivo externo no qual cada partícula irá depositar suas experiências de voo após cada ciclo de voo, tem sua atualização realizada considerando um sistema de base geográfica definida em termos dos valores da função objetivo de cada indivíduo. Esta técnica é inspirada no arquivo externo usado no *Pareto Archive Evolution Strategy* (Knowles; Corne, 1999). O repositório citado anteriormente é utilizado pelas partículas para identificar um líder que irá orientar a busca. É implementado um mecanismo para que cada partícula possa escolher um guia diferente. O mecanismo é baseado na geração de hipercubos como um sistema de coordenadas onde as coordenadas de cada partícula são definidas de acordo com os valores de suas funções objetivo dividindo o espaço de busca explorado.

Se as soluções não dominadas no arquivo externo não atingirem o tamanho máximo, todos os indivíduos do conjunto de candidatos são inseridos no arquivo externo de acordo com o valor da classificação de aptidão individual e são armazenados; se as soluções não dominadas no arquivo externo atingirem o tamanho máximo, os indivíduos são selecionados de maneira aleatória e os restantes serão eliminados do arquivo externo. Em outras palavras, para manter o número de partículas no arquivo externo em um nível estável, quando certo número de partículas for excluído, o mesmo número de partículas será adicionado do conjunto candidato.

2.3.2. Otimização de Lobos Cinzentos Multiobjetivo

O algoritmo Otimização de Lobos Cinzentos Multiobjetivo (do termo em inglês, *Multiobjective Optimization Grey Wolf Optimization* – MOGWO) foi desenvolvido por MIRJALILI et al. (2016), ele mimetiza o comportamento da hierarquia social e da caça dos lobos cinzentos.

A hierarquia social é dividida em quatro tipos de lobos: os líderes são os alpha (α), o segundo nível na hierarquia são os beta (β), o terceiro nível na hierarquia são os delta (δ) e o nível de hierarquia mais baixa são os ômega (ω) que se submete a todos os outros lobos.

Na caça, os lobos cinzentos possuem a capacidade de reconhecer a localização da presa e cercá-la. Geralmente a caçada é guiada pelo α . O β e o δ

podem também participar na caçada ocasionalmente. As principais fases da caça da alcateia são divididas em: procura da presa, perseguição, abordagem da presa, cercando e assediando a presa até que ela pare de se mover, por fim, o ataque da alcateia contra a presa.

A etapa de procura das presas é fase de exploração das soluções pelo algoritmo, enquanto a etapa ataque às presas é a fase de intensificação da procura das soluções. Apesar de o MOGWO permitir que seus agentes de busca atualizem sua posição com base na localização do α , β e δ o cercamento da presa consegue explorar até certo limite sendo suscetível a estagnação em soluções locais. Com isso, a exploração é garantida pelo parâmetro A com valores aleatórios entre os limites inferior e superior $[-1 \ 1]$, onde obriga o agente de busca a divergir da presa. Outro componente do MOGWO que favorece a exploração é o vetor C . O vetor C gera valores aleatórios no intervalo $[0 \ 2]$ que são atribuídos para as presas, ao fornecer estocasticamente ($C > 1$) ou ($C < 1$) o efeito é um aumento ou uma diminuição na definição da distância entre lobo e presa.

O MOGWO tem semelhanças com o algoritmo MOPSO, primeiro possui um arquivo, que é responsável por armazenar as soluções de Pareto ótimo obtidas em primeira instância. O arquivo externo é a unidade de armazenamento que salva ou recupera as soluções de Pareto ótimo ao longo das iterações executadas. Se o arquivo estiver cheio, é necessário reorganizar a segmentação do espaço. Nesse caso, encontra-se o segmento mais cheio para retirar uma de suas soluções. Em seguida, a nova solução deve ser inserida no segmento menos congestionado para melhorar a diversidade das soluções Pareto ótimas.

A segunda semelhança é a estratégia de seleção de líder, escolha das soluções α , β e δ que guiam os outros lobos em busca da fronteira de Pareto. Como na otimização multiobjetivo, as soluções não podem ser facilmente comparadas devido aos conceitos de otimalidade de Pareto. O mecanismo de seleção de líder é escolher três soluções aleatórias para serem os lobos α , β e δ , no segmento menos congestionado. Se houver menos de três soluções no hipercubo de menor lotação, o segundo hipercubo menor também é encontrado e utilizado para escolher os outros líderes. Se caso o segundo hipercubo de menor lotação não conseguir completar a lista de líderes, então o líder δ deve ser escolhido a partir do terceiro hipercubo de menor lotação.

Com esse método, consegue se evitar que o MOGWO escolha líderes similares para os lobos α , β e δ . Consequentemente, a busca é sempre em direção

às áreas inexploradas ou não expostas no espaço de busca, uma vez que o mecanismo de seleção de líder favorece os hipercubos de menor lotação e oferece líderes de diferentes segmentos se não houver número suficiente de líderes (menos de 3) no hipercubo de menor lotação (Mirjalili et al., 2016).

2.3.3. Algoritmo de Enxame Salpa Multiobjetivo

O algoritmo Otimização de Enxame Salpa Multiobjetivo (do termo em inglês, *Multi-objective Salp Swarm Algorithm – MSSA*) foi proposta por MIRJALILI et al. (2017), mimetiza o comportamento de locomoção das salpas (do termo em inglês, Salps).

As salpas pertencem à família das *Salpidae* e possuem corpo transparente em forma de barril. Seus tecidos são muito semelhantes aos das medusas. Eles também se movem de forma muito semelhante às águas-vivas, nas quais a água é bombeada pelo corpo como propulsão para avançar. Um dos comportamentos mais interessantes das salpas é o seu comportamento de locomoção em cadeia. Em oceanos profundos, as salpas geralmente formam um enxame chamado cadeia de salpas. A principal razão deste comportamento é que isso é feito para conseguir uma melhor locomoção usando mudanças rápidas e coordenadas e forrageamento.

Similarmente a outras técnicas baseadas em enxames, a posição das salpas é definida em um espaço de busca n -dimensional onde n é o número de variáveis de um determinado problema. Supõe-se também que existe um repositório de fontes alimentares. Este repositório mantém as melhores soluções não dominadas obtidas durante a otimização e é muito semelhante ao arquivo externo do MOPSO. O repositório possui um tamanho máximo para armazenar um número limitado de soluções não dominadas. Entretanto, quando o repositório fica cheio e um salp não é dominado em comparação com os residentes do repositório, uma maneira de atualizar o repositório é remover uma das soluções semelhantes não dominadas do repositório. Assim, o melhor candidato a ser removido do arquivo é aquele que está nas regiões mais povoadas. Esta abordagem melhora a distribuição dos residentes do arquivo ao longo das iterações.

Para encontrar as soluções não dominadas com vizinhança povoada, o número de soluções vizinhas com distância máxima é contado. Esta distância é definida por $\vec{d} = \frac{\overline{max} - \overline{min}}{\text{tamanho do repositório}}$ onde max e min são dois vetores para

armazenar valores máximos e mínimos para cada objetivo, respectivamente. O repositório com uma solução em cada segmento é o melhor caso. Após atribuir uma classificação a cada residente do repositório com base no número de soluções vizinhas, uma roleta é utilizada para escolher uma delas. Quanto maior o número de soluções vizinhas (maior número de classificação) para uma solução, maior a probabilidade de removê-la do repositório.

A fonte de alimento pode ser escolhida aleatoriamente no repositório. Contudo, neste algoritmo ela é selecionada a partir de um conjunto de soluções não dominadas com a vizinhança menos populosa. Na manutenção do arquivo, as soluções com classificação mais elevada (vizinhança lotada) têm maior probabilidade de serem escolhidas. Por outro lado, quanto menos povoada a vizinhança (o número de classificação mais baixo) para uma solução não dominada no repositório, maior a probabilidade de ser selecionada como fonte de alimento.

Resumidamente o algoritmo MSSA primeiro inicializa a população de salpas, então calcula os valores objetivos para cada salpa e encontra as não dominadas. As soluções não dominadas são adicionadas ao arquivo se o repositório não estiver cheio. Se o repositório estiver cheio, a manutenção do repositório é executada para excluir as soluções com vizinhança lotada. Nesta etapa, as soluções são primeiro classificadas e depois selecionadas por meio de uma roleta. Depois de remover um número suficiente de residentes do repositório, as salpas não dominadas podem ser adicionadas ao repositório. Após a atualização do repositório, uma fonte de alimento é selecionada entre as soluções não dominadas no repositório com a vizinhança menos populosa. Da mesma forma que a manutenção do arquivo, isso é feito classificando as soluções e utilizando uma roleta. O próximo passo é atualizar a posição das salpas líderes/seguidoras, todas as etapas descritas anteriormente, exceto a inicialização, são repetidas até a satisfação de uma condição final.

3. CRITÉRIOS DE DESEMPENHO UTILIZADOS NA PESQUISA

Este capítulo visa apresentar algumas ferramentas essenciais para o desenvolvimento do atual trabalho, norteadas por quais ferramentas podem ser utilizadas e quais metaheurísticas multi/muitos objetivos a serem comparadas com as propostas deste trabalho. Portanto, uma revisão bibliográfica sobre quais ferramentas podem ser utilizadas na otimização multiobjetivo e muitos objetivos são apresentadas a seguir.

Posto isto alguns questionamentos devem ser levados em consideração, como:

- Quais as métricas de qualidade mais utilizadas?
- Quais as metaheurísticas mais utilizadas?
- Qual é o método de comparação que vem sendo utilizado com maior frequência?
- Quais as funções de *benchmark* mais apontadas na literatura?

Há diversos algoritmos usados para encontrar uma aproximação da fronteira de Pareto. É o caso dos algoritmos inspirados na natureza (Li et al., 2024; Yang et al., 2024; e Zhang et al., 2020). Para comparar tais algoritmos, existem vários indicadores na literatura que medem a qualidade das aproximações do conjunto de Pareto, mas nenhum destes indicadores se provou melhor do que os outros em todos os aspectos (Falcón-Cardona; Coello, 2020). Infelizmente, medir a qualidade de aproximações do conjunto de Pareto é um problema de otimização multiobjetivo em si. Tais conjuntos buscam produzir pontos no espaço objetivo tão próximos da fronteira de Pareto como possível e garantir diversidade suficiente para as diferentes necessidades do tomador de decisão, além de outros aspectos. Na literatura, é comum utilizar mais de um indicador de qualidade para comparar diferentes aproximações da fronteira de Pareto.

Para a obtenção dos dados a serem apresentados, os sites IEEE Xplore e Science Direct da Elsevier foram utilizados, e foram escolhidos trabalhos científicos pertencentes aos anos de 2021, 2022, 2023 e 2024.

Nas próximas subseções pretende-se mostrar resumidamente respostas para os questionamentos colocados anteriormente.

3.1. MÉTRICAS DE QUALIDADE PARA PROBLEMAS MULTIOBJETIVOS E MUITOS OBJETIVOS

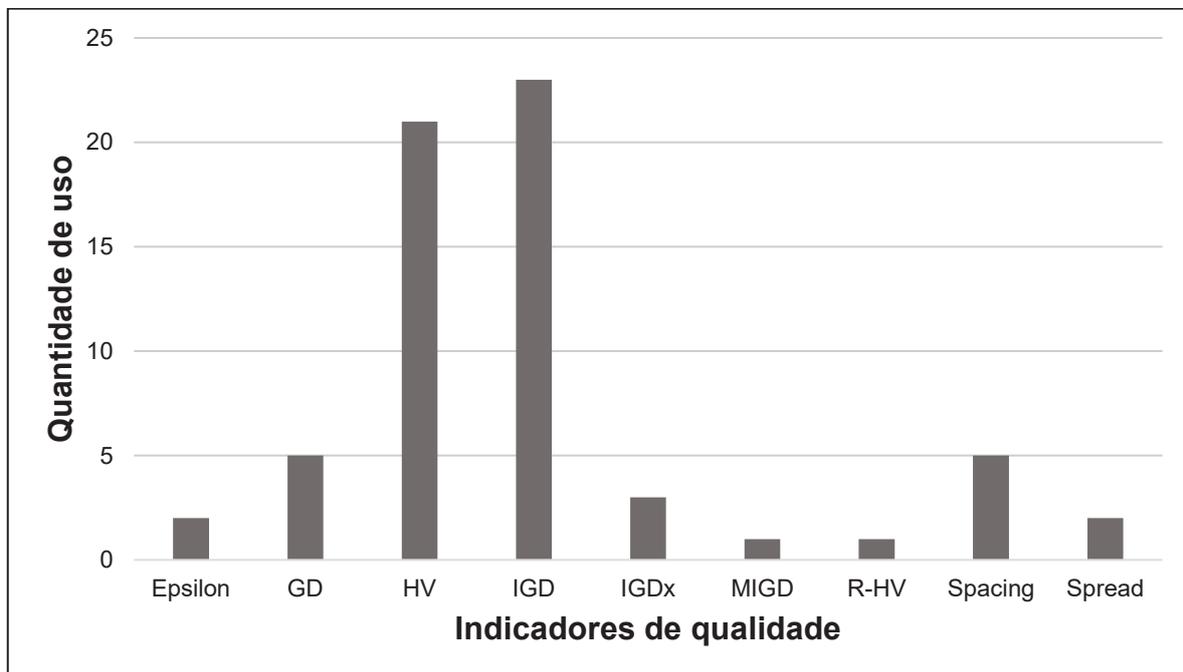
A análise dos algoritmos de otimização multiobjetivos e muitos objetivos são realizadas a partir de um ou mais indicadores que qualifiquem as soluções e permita compará-las entre si, ou seja, conseguem medir a qualidade das soluções.

Zitzler (1999) cita a utilização de alguns indicadores e neles podem-se encontrar:

- As soluções que estão à menor distância possível da fronteira Pareto ótima;
- As soluções com maior diversidade na fronteira de Pareto ótima;
- Maior dispersão de indivíduos sobre o espaço de busca.

A seguir apresenta-se algumas métricas disponíveis na literatura e utilizadas com maior frequência para problemas de otimização em engenharia e foram selecionados os trabalhos com aplicação dentro da área da engenharia elétrica.

Figura 5 – Principais indicadores de qualidade para problemas de otimização em engenharia.



Fonte: Autora (2024).

Da pesquisa realizada o atual trabalho irá utilizar as métricas de qualidade HV e IGD por serem as mais utilizadas na revisão bibliográfica realizada. As métricas de qualidade das soluções avaliam a convergência e a diversidade das

soluções obtidas (conjunto de Pareto ou soluções não dominadas encontradas pelo algoritmo) frente à fronteira de Pareto ótima (Von Lücken et al., 2014).

A diversidade tem por objetivo verificar a dispersão das soluções obtidas pelos algoritmos face à fronteira de Pareto ótima. Já a convergência tem como objetivo verificar o quão as soluções obtidas pelos algoritmos estão próximas da fronteira de Pareto ótima (Cheng et al., 2012).

A métrica hipervolume (do termo em inglês *Hipervolum* – HV) é usada para calcular o volume do espaço objetivo coberto por soluções pertencentes às soluções ótimas de Pareto obtidas no conjunto de referência. O valor de HV é calculado como:

$$HV = VOL(U(f_1(x_i), r_1^*) \times \dots \times (f_M(x_i), r_M^*)) \quad (3.1)$$

sendo um ponto de referência no espaço objetivo dominado por todas as soluções ótimas de Pareto no conjunto de soluções e $VOL(\bullet)$ indica a medida de Lebesgue. Matematicamente, cada solução i pertence ao conjunto de Pareto encontrado pelo algoritmo, é construído um hipercubo com referência a um ponto de referência ($r^* = (r_1^*, r_2^*, \dots, r_M^*)^T$). O ponto de referência irá ser utilizado no cálculo do volume em relação ao conjunto de Pareto, onde pode ser representado por um ponto contendo os piores valores da função objetivo ($(f_1(x_i), \dots, f_M(x_i))$). Por fim, como resultado da métrica (HV) será a união de todos os hipercubos encontrados. E quanto maior for o resultado melhor, pois indica que houve um alto espalhamento entre as soluções extremas do conjunto de Pareto encontrado pelo algoritmo e apresenta maior convergência.

A métrica distância geracional invertida (do termo em inglês *Inverted Generational Distance* – IGD) é utilizado para medir a distância média de soluções ótimas de Pareto verdadeiras para soluções de Pareto ótimo ou soluções não dominadas obtidas pelo algoritmo. O valor de IGD é calculado por:

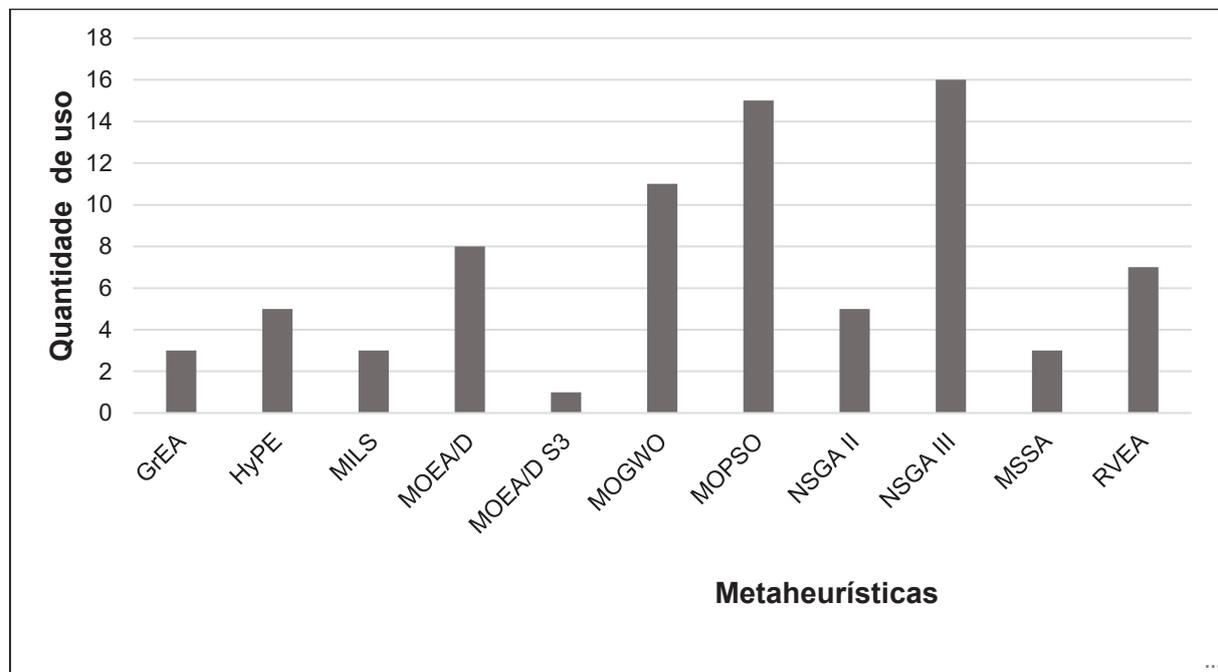
$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (3.2)$$

onde n é o número de soluções ótimas de Pareto verdadeiras e d_i indica a distância euclidiana entre a i -ésima solução ótima de Pareto verdadeira e as soluções ótimas de Pareto obtidas mais próximas no conjunto de referência (Ishibuchi et al., 2015).

3.2. METAHEURÍSTICAS MAIS UTILIZADAS

Ao longo das décadas os problemas de otimização na área computacional e das engenharias apresentaram um aumento significativo em relação ao grau de dificuldade para se tratar de problemas de otimização. Desta maneira, diversas metaheurísticas vem sendo propostas para suprir as lacunas (como: tempo computacional, o número de parâmetros, a convergência precoce e etc.) presentes em problemas multiobjetivos e de muitos objetivos. Em vista disto, serão apresentadas algumas metaheurísticas mais utilizadas na área da engenharia para tratar de problemas de modelagem e foi possível obter aproximadamente x trabalhos na literatura.

Figura 6 – Metaheurísticas utilizadas com frequência na área de engenharia.



Fonte: Autora (2024)

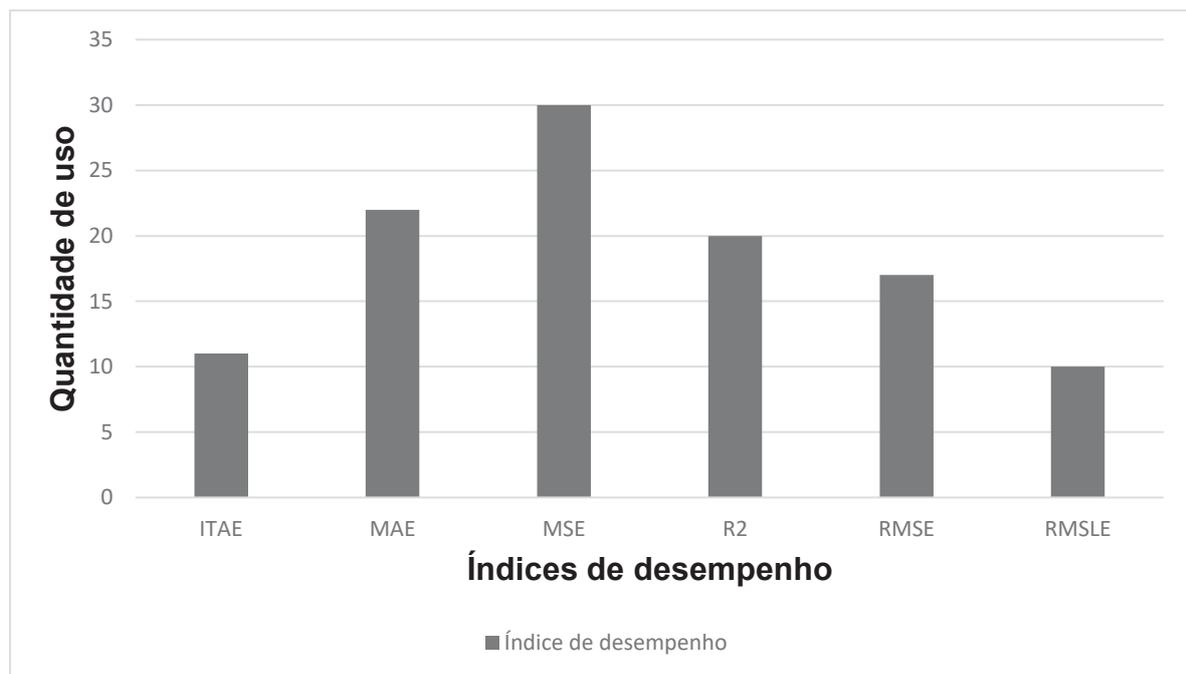
Da pesquisa realizada o atual trabalho irá utilizar parte das metaheurísticas NSGA III e RVEA para comporem os métodos de seleção de soluções para as metaheurísticas propostas ao utilizar os componentes chave para se escolher as soluções que serão denominadas de conjunto de Pareto ótima. Enquanto comparará as soluções obtidas com as metaheurísticas propostas com as metaheurísticas NSGA III, MOPSO, MOGWO, MOEA/D e, MSSA para funções de *benchmark*. E para os problemas de engenharia serão comparados os resultados

das metaheurísticas propostas com as já propostas que possuam os melhores resultados nas funções de *benchmark*.

3.3. ÍNDICES DE DESEMPENHO

Em um processo de estimação de parâmetros, a função objetivo é usada para avaliar a solução viável e fornecer a direção evolutiva correta para encontrar a solução ótima. A forma mais usual de medir a qualidade dos parâmetros estimados é comparar a simulação do modelo obtido com os dados medidos e então utilizar indicadores estatísticos para inferir um valor que possa ser analisado. Neste contexto, serão apresentados alguns índices de desempenho mais utilizadas na área da engenharia para tratar de problemas de identificação de sistemas não lineares.

Figura 7 – Principais índices de desempenho utilizados na área de engenharia.



Fonte: Autora, 2024.

Neste trabalho são utilizados 4 indicadores para medir a qualidade dos parâmetros estimados, são eles: o Erro Médio Absoluto (MAE); a raiz do erro quadrático médio (RMSE); o erro de log quadrático médio (RMSLE); e o Erro Absoluto de Tempo Integral (ITAE).

- Erro Médio Quadrático (do termo em inglês, *Mean Squared Error* - MSE): O MSE é uma medida da qualidade dos parâmetros estimados, e quanto mais

próximo de zero for o erro (para problemas de minimização) significa que o modelo estudado é válido. Além disso, é a diferença do erro ao quadrado entre as saídas do modelo estimado com a saída do sistema real. A sua formulação é dada pela Equação (3.3).

$$MSE_i = \frac{\sum_{k=1}^n (\delta_i(k) - \hat{\delta}_i(k))^2}{n} \quad (3.3)$$

onde n é o número máximo de amostras, $\delta_i(k)$ é a saída do sistema real e $\hat{\delta}_i(k)$ é a saída estimada por técnicas de otimização.

- Raiz do Erro Médio Quadrático (do termo em inglês, *Root Mean Squared Error* - RMSE): este critério resulta na raiz quadrática do erro médio. A Equação (3.4) ilustra sua formulação matemática

$$RMSE_i = \sqrt{\frac{1}{n} \sum_{i=1}^n (\delta_i(k) - \hat{\delta}_i)^2} \quad (3.4)$$

temos n é o número máximo de amostras, $\delta_i(k)$ é a saída do sistema real e $\hat{\delta}_i(k)$ é a saída estimada por técnicas de otimização

- Erro Logaritmo Médio Quadrática da Raiz (do termo em inglês *Root Mean Square Algorithm Error* - RMSLE): é uma extensão do RMSE e é usado principalmente quando as previsões têm grandes desvios, portanto o erro RMSLE não é tão afetado quanto com o RMSE, é definido como Equação (3.5)

$$RMSLE_i = \frac{\sum_{k=1}^n (\delta_i(k) - \bar{\delta}_i(k)) (\hat{\delta}_i(k) - \hat{\delta}_i k)}{\sqrt{\sum_{k=1}^n (\delta_i(k) - \bar{\delta}_i(k))^2 \sum_{k=1}^n (\hat{\delta}_i(k) - \hat{\delta}_i k)^2}} \quad (3.5)$$

onde δ_i e $\hat{\delta}_i$ são a média da saída medida e a da saída simulada, respectivamente.

- Integral do valor Absoluto do Erro Ponderado no Tempo (do termo em inglês *Integral Time Absolut Error* – ITAE): integra o erro absoluto multiplicado pelo tempo ao longo do tempo. O que isso faz são erros de peso que existem depois de um longo tempo com muito mais intensidade do que aqueles no início da resposta, é definido como na Equação (3.6).

$$ITAE = \delta(t) = \int_i^{\infty} t |\delta_i(t) - \hat{\delta}_i(t)| dt \quad (3.6)$$

sendo $\delta_i(k)$ é a saída do sistema real e $\hat{\delta}_i(k)$ é a saída estimada por técnicas de otimização.

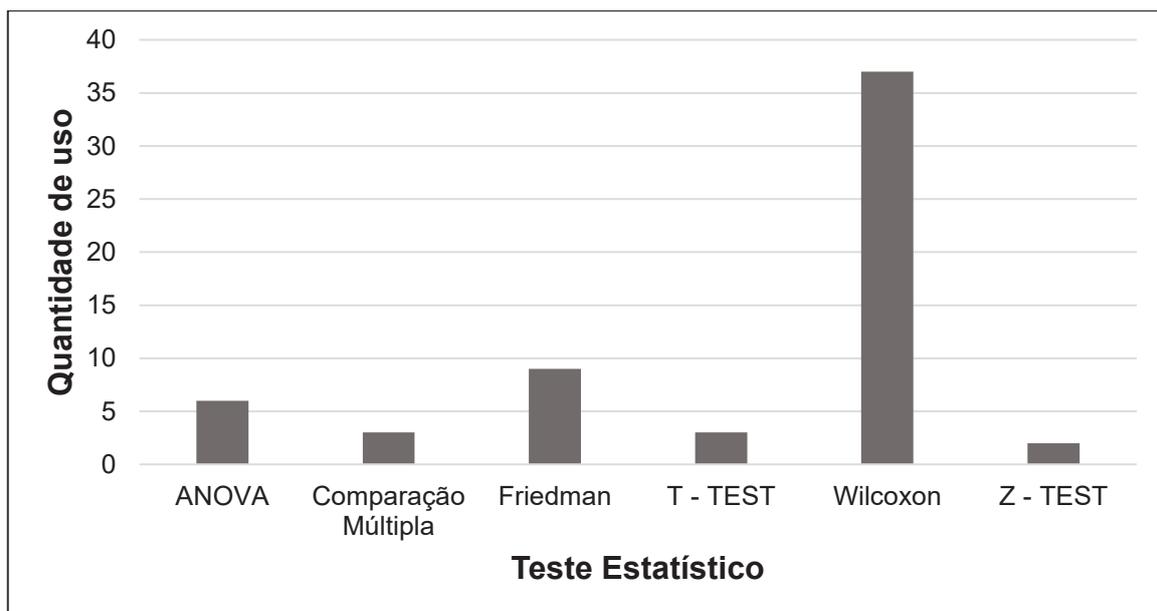
Portanto, pode se dizer que um bom modelo é validado quando for capaz de reproduzir as características aproximadas ou exatas do sistema identificado. Os testes dinâmicos validam o modelo com os parâmetros estimados utilizando os dados de medição. Nestes testes são realizadas uma análise visual comparativa com os dados de medição do sistema identificado, determinando se o modelo está ou não adequado.

3.4. MÉTODOS DE COMPARAÇÃO DE METAHEURÍSTICAS

Os testes estatísticos são ferramentas usadas para obter informação e significados de conjuntos de dados variáveis. Estes testes estão em várias formas, e são escolhidos dependendo das necessidades dos investigadores e das características das variáveis analisadas.

A Figura 8 mostra alguns testes estatísticos disponíveis na literatura e utilizados para realizar uma análise entre as metaheurísticas.

Figura 8 – Principais testes estatísticos.



Fonte: Autora, 2024.

No presente trabalho foi utilizado dois tipos de testes, o primeiro teste é de Kruskal Wallis (Ostertagová et al., 2014), que é uma extensão do teste de Wilcoxon-Mann-Whitney e que é utilizado em dados não paramétricos e compara três ou mais amostras de coluna única com cada amostra. Ele usa classificação de dados em vez de valores numéricos para calcular estatísticas de teste e algumas suposições são necessárias, tais como:

- Distribuições contínuas para a variável de teste devem ser exatamente as mesmas (exceto suas medianas) para as diferentes populações.
- Casos que representem as amostras aleatórias das populações e as pontuações na variável de teste são independentes um do outro.

O teste de hipóteses é um procedimento ou regra de decisão que permite aceitar ou rejeitar uma hipótese estatística, com um grau de risco conhecido e com base nas evidências (informações) fornecidas pela amostra/população.

Portanto se o teste de Kruskal-Wallis levar a resultados significativos, então pelo menos uma das amostras será diferente das outras amostras. O teste não identifica onde ocorrem às diferenças ou quantas diferenças realmente ocorrem. A hipótese nula e alternativa são ilustradas pela Equação (3.7).

$$\begin{aligned} H_0 &\rightarrow F_I(x) = F_J(x), \\ H_1 &\rightarrow F_I(x) \neq F_J(x) \end{aligned} \quad (3.7)$$

Sendo H_0 as medianas da população são iguais, H_1 as medianas da população são diferentes, $F_I(x)$, $F_J(x)$ são as distribuições contínuas da população $x_{I,J}$, $I = 1, 2, \dots, k$ e $J = 1, 2, \dots, n_i$, são os dados do I –ésimo grupo (nível) e J –ésimo observação. Portanto, se H_0 é verdadeira as medianas populacionais são iguais, por outro lado se H_1 for a verdadeira as medianas populacionais não são iguais.

Para o teste de comparações múltiplas o método Kruskal-Wallis utiliza outro tipo de procedimento em conjunto, como o post hoc de *Conover-Inman* (Kayri et al., 2022; Kumar Singla et al., 2021). Este procedimento é uma simplificação do método Fisher LSD (diferença mínima significativa) realizado nas linhas dos dados que se quer analisar as possíveis semelhanças. As amostras I e J são significativamente diferentes no nível de significância α se o valor absoluto da diferença entre suas classificações médias for maior que a diferença menos significativa, ou seja, se a seguinte desigualdade for satisfeita (Equação (3.8)).

$$|R_I - R_J| > t_{1-\frac{\alpha}{2}, N-k} \sqrt{\frac{N(N+1)}{12} \frac{N-1-H}{N-k} \left(\frac{1}{n_I} + \frac{1}{n_J} \right)} \quad (3.8)$$

onde $|R_I - R_J|$ é a média geral, $t_{1-\frac{\alpha}{2}, N-k}$ é o número de empates do I – ésimo grupo de m grupos empatados, R_I é a soma das classificações na I – ésima amostra, $J = 1, 2, \dots, k$.

$$\bar{R}_{I,J} = \frac{R_{I,J}}{n_{I,J}} \quad (3.9)$$

$N - k$ é o grau de liberdade e N pode ser obtido através da Equação (3.10).

$$N = \sum_{I=1}^k n_I \quad (3.10)$$

sendo N as observações do menor para o maior atribuindo, ou seja, independentemente de qual amostra elas vêm, e atribua a menor classificação de observação 1, a próxima menor classificação 2 até atingir a maior classificação de observação N . Por fim, para identificar a diferença entre as técnicas (algoritmo proposto x algoritmo escolhido da literatura) utilizou-se um nível de significância (α) igual a 3% ou 0,003.

Outro teste utilizado é o de comparações múltiplas. No presente trabalho foi utilizado o teste de comparações múltiplas $1 \times Z$, onde o algoritmo considerado como algoritmo de controle será o algoritmo proposto, então ele foi comparado com os $m = (W - 1)$ algoritmos escolhidos da literatura. Neste tipo de teste $Z \times Z$ podem ser considerados $m = \frac{W(W-1)}{2}$ comparações possíveis entre os algoritmos, então se tem as seguintes hipóteses (Equação (3.11)):

$$\begin{aligned} H : m &= (W - 1), & 1 \times Z \\ H : m &= \frac{W(W-1)}{2}, & Z \times Z \end{aligned} \quad (3.11)$$

Além disso, as hipóteses podem ser ordenadas de acordo com o p – valor. Nestes casos, o p – valor das hipóteses são obtidas pela conversão de classificação dos postos calculados aplicando-se a estatística de teste (z) para a comparação do i – ésimo algoritmo e do j – ésimo algoritmo. O valor de z resulta a probabilidade de distribuição normal para ser comparada com α . A Equação (3.12) ilustra matematicamente a estatística de teste(z):

$$z = \frac{R_i - R_j}{\sqrt{\frac{W(W+1)}{6n}}} \quad (3.12)$$

onde R_i e R_j os postos médios dos algoritmos comparados, W é o número máximo de algoritmos, n o número de observações.

Uma das vantagens de se utilizar o teste de comparações múltiplas é que ele consegue retornar informações sobre quais pares de médias podem ser significativamente diferentes e quais não são dos algoritmos.

3.5. FUNÇÕES DE *BENCHMARK*

Um novo algoritmo de otimização deve ter seu desempenho validado e comparado com outros algoritmos existentes utilizando um conjunto de funções de *benchmark* adequado. Idealmente, as funções de *benchmark* usadas devem ser diversas e não tendenciosas, apesar de não haver um conjunto de funções de *benchmark* unânime na literatura.

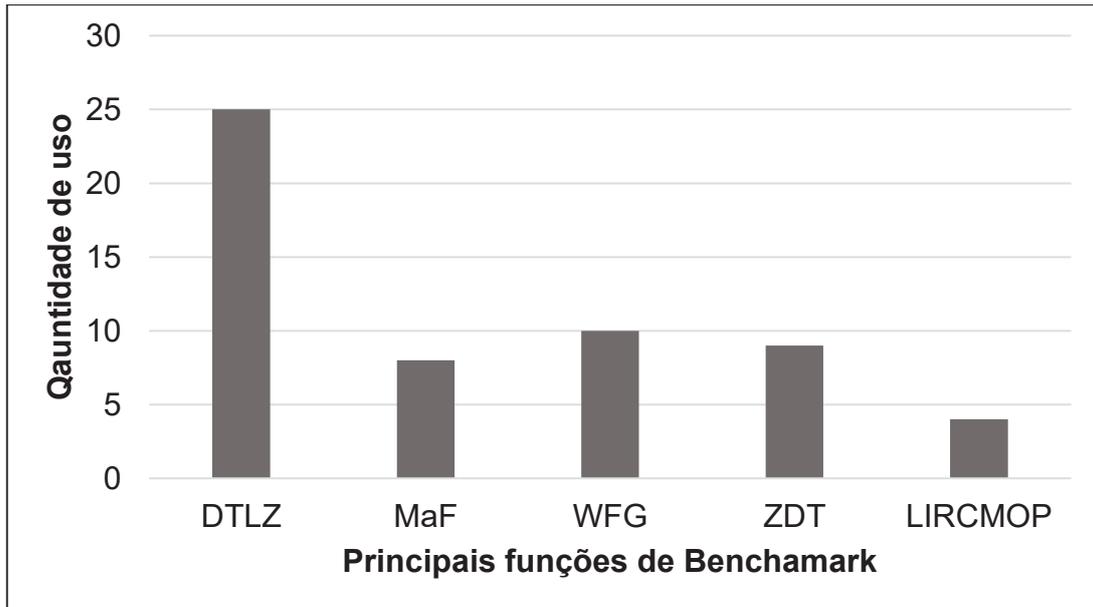
Basicamente, as funções de *benchmark* são uma medida de referência usada para avaliar a eficiência e as estratégias de um sistema ou processo. Uma forma bastante eficaz e utilizada para comparar o desempenho de diferentes metaheurísticas.

A literatura disponibiliza vários conjuntos de funções de *benchmark*, quando se trata de problemas de otimização multiobjetivo e com muitos objetivos. Os principais problemas com muitos objetivos que estão sendo desenvolvidos para tratar a escalabilidade e a eficácia na busca de soluções de Pareto ótimo em espaços de solução de alta dimensionalidade.

A Figura 9 mostra os conjuntos de funções de *benchmark* mais utilizados na revisão bibliográfica mencionada anteriormente e os trabalhos selecionados dentro da área das engenharias.

No presente trabalho será utilizado o *benchmark* MaF que utiliza 15 problemas de teste para otimização multiobjectivo originalmente fornecidos por Cheng et al (2018), e usados na Competição CEC'2018, que são mostradas na Tabela 3. Embora ela não seja a mais utilizada na literatura ela traz versões modificadas das funções que pertencem a DTLZ, WFG e outros conjuntos.

Figura 9 – Principais funções de *Benchmark*.



Fonte: Autora, 2024.

Os 15 problemas de *benchmark* do conjunto MAF possuem diversas propriedades que cobrem uma boa representação de vários cenários do mundo real, como ser multimodal, desconectado, degenerado e/ou inseparável, e ter uma forma de frente de Pareto irregular, um conjunto de Pareto complexo ou grande número de variáveis de decisão como destacado na Tabela 1.

A Figura 10 ilustra as fronteiras de Pareto para as quinze MaFs com 3 objetivos.

Tabela 3 – Funções de benchmark (onde, B = Biased; BD = Biased Deceptive; C = Concave; Co = Convex; Di = Disconnected; De = Degenerate; L= Linear; Ls = Large scale; M = Multimodal; Mx = Mixed; N = Nonseparable; U=Unimodal; Ps= Partially separable)

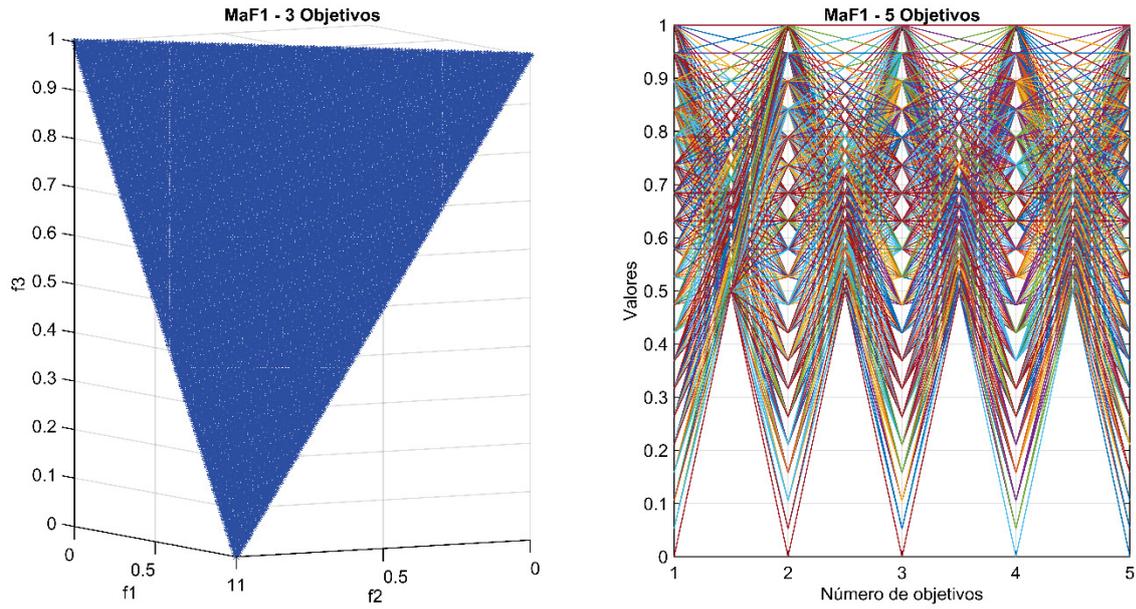
Função	Propriedades	Fórmula matemática
MaF1	L	$g(x, M) = \sum_{i=M}^{ x } (x_i - 0,5)^2$
MaF2	C	$g(x, M) = \sum_{j=M+(i-1) \cdot \lfloor \frac{D-M+1}{M} \rfloor}^D \left(\left(\frac{x_i}{2} + \frac{1}{4} \right) - 0,5 \right)^2$
MaF3	Co, M	$g(x, M) = 100 \left[x_M + \sum_{i=M}^{ x } (x_i - 0,5)^2 - \cos(20\pi(x_i - 0,5)) \right]$
MaF4	C, M	$g(x, M) = 100 \left[x_M + \sum_{i=M}^{ x } (x_i - 0,5)^2 - \cos(20\pi(x_i - 0,5)) \right]$
MaF5	C, B	$g(x, M) = \sum_{i=M}^{ x } (x_i - 0,5)^2$
MaF6	C, De	$g(x, M) = \sum_{i=M}^{ x } (x_i - 0,5)^2$
MaF7	Mx, D, M	$g(x, M) = 1 + \frac{9}{ x_M } \sum_{i=M}^{ x } x_i$
MaF8	L, De	$\min \begin{cases} f_1(x) = d(x, A_1) \\ f_2(x) = d(x, A_2) \\ \dots \\ f_M(x) = d(x, A_M) \end{cases}$

MaF9	L, De	$\min \begin{cases} f_1(x) = d(x, \overrightarrow{A_1 A_2}) \\ f_2(x) = d(x, \overrightarrow{A_2 A_3}) \\ \dots \\ f_M(x) = d(x, \overrightarrow{A_M A_1}) \end{cases}$
MaF10	Mx, B	$\min \begin{cases} f_1(x) = y_M + 2 \left(1 - \cos\left(\frac{\pi}{2} y_1\right)\right) \dots \left(1 - \cos\left(\frac{\pi}{2} y_{M-2}\right)\right) \left(1 - \cos\left(\frac{\pi}{2} y_{M-1}\right)\right) \\ f_2(x) = y_M + 4 \left(1 - \cos\left(\frac{\pi}{2} y_1\right)\right) \dots \left(1 - \cos\left(\frac{\pi}{2} y_{M-2}\right)\right) \left(1 - \sin\left(\frac{\pi}{2} y_{M-1}\right)\right) \\ f_{M-1}(x) = y_M + 2(M-1) \left(1 - \cos\left(\frac{\pi}{2} y_1\right)\right) \left(1 - \sin\left(\frac{\pi}{2} y_2\right)\right) \\ f_M(x) = y_M + 2M \left(1 - y_1 - \frac{\cos(10\pi y_1 + \pi/2)}{10\pi}\right) \end{cases}$
MaF11	Co, Di, N	$\min \begin{cases} f_1(x) = y_M + 2 \left(1 - \cos\left(\frac{\pi}{2} y_1\right)\right) \dots \left(1 - \cos\left(\frac{\pi}{2} y_{M-2}\right)\right) \left(1 - \cos\left(\frac{\pi}{2} y_{M-1}\right)\right) \\ f_2(x) = y_M + 4 \left(1 - \cos\left(\frac{\pi}{2} y_1\right)\right) \dots \left(1 - \cos\left(\frac{\pi}{2} y_{M-2}\right)\right) \left(1 - \sin\left(\frac{\pi}{2} y_{M-1}\right)\right) \\ f_{M-1}(x) = y_M + 2(M-1) \left(1 - \cos\left(\frac{\pi}{2} y_1\right)\right) \left(1 - \sin\left(\frac{\pi}{2} y_2\right)\right) \\ f_M(x) = y_M + 2M \left(1 - y_1 \cos^2(5\pi y_1)\right) \end{cases}$
MaF12	C, N, BD	$\min \begin{cases} f_1(x) = y_M + 2 \sin\left(\frac{\pi}{2} y_1\right) \dots \sin\left(\frac{\pi}{2} y_{M-2}\right) \sin\left(\frac{\pi}{2} y_{M-1}\right) \\ f_2(x) = y_M + 4 \sin\left(\frac{\pi}{2} y_1\right) \dots \sin\left(\frac{\pi}{2} y_{M-2}\right) \cos\left(\frac{\pi}{2} y_{M-1}\right) \\ f_{M-1}(x) = y_M + 2(M-1) \sin\left(\frac{\pi}{2} y_1\right) \cos\left(\frac{\pi}{2} y_2\right) \\ f_M(x) = y_M + 2 \cos\left(\frac{\pi}{2} y_1\right) \end{cases}$

MaF13	C, U, N, De	$\min \begin{cases} f_1(x) = \sin\left(\frac{\pi}{2}x_1\right) + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2 \\ f_2(x) = \cos\left(\frac{\pi}{2}x_1\right) \sin\left(\frac{\pi}{2}x_2\right) + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2 \\ f_3(x) = \cos\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right) + \frac{2}{ J_3 } \sum_{j \in J_3} y_j^2 \\ f_{4,\dots,M}(x) = f_1(x)^2 + f_2(x)^{10} + f_3(x)^{10} + \frac{2}{ J_4 } \sum_{j \in J_4} y_j^2 \end{cases}$
MaF14	L, Ps, Ls	$\min \begin{cases} f_1(x) = x_1^f \dots x_{M-1}^f (1 + \sum_{j=1}^M c_{1,j} \bar{g}_1(x_j^s)) \\ f_2(x) = x_1^f \dots (1 - x_{M-1}^f) (1 + \sum_{j=1}^M c_{2,j} \bar{g}_2(x_j^s)) \\ \dots \\ f_{M-1}(x) = x_1^f (1 - x_2^f) (1 + \sum_{j=1}^M c_{M-1,j} \bar{g}_{M-1}(x_j^s)) \\ f_M(x) = x_1^f (1 - x_1^f) (1 + \sum_{j=1}^M c_{M,j} \bar{g}_M(x_j^s)) \\ x \in [0,10]^{ x } \end{cases}$
MaF15	Co, Ps, Ls	$\begin{cases} x^s \leftarrow \left(1 + \cos\left(0,5\pi \frac{i}{x^s}\right)\right) (x_i^s - l_i) - x_1^f (u_i - l_i) \\ i = 1, 2, \dots, x^s \end{cases}$

- MaF1

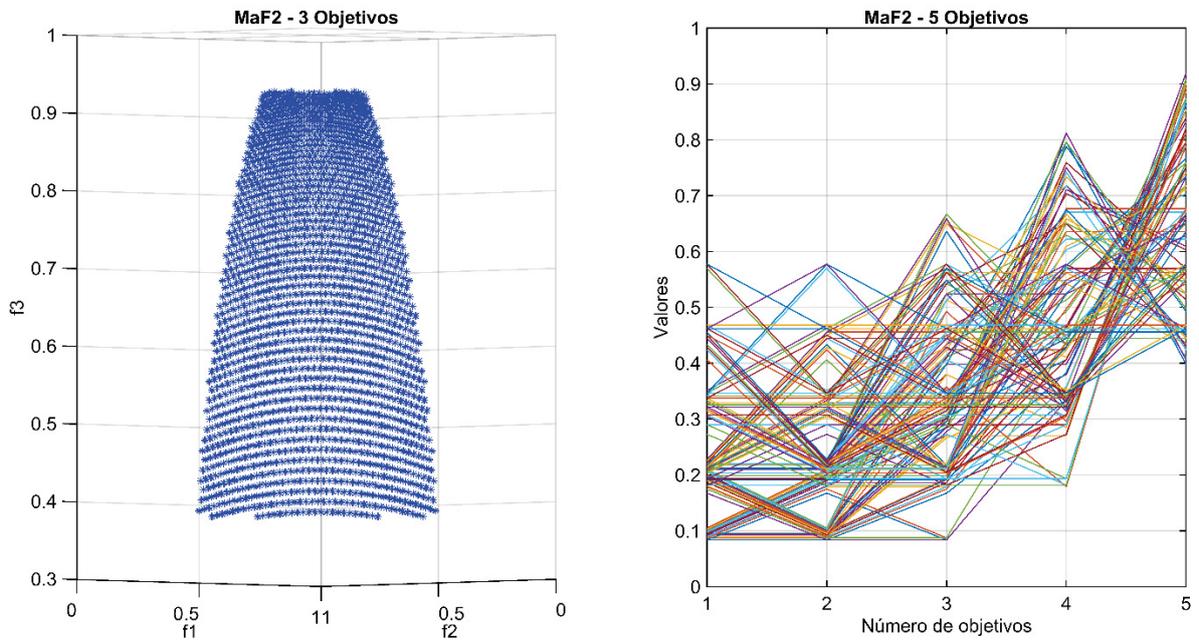
Figura 10 – Fronteiras de Pareto da MaF1.



Fonte: Autora, 2024.

- MaF2

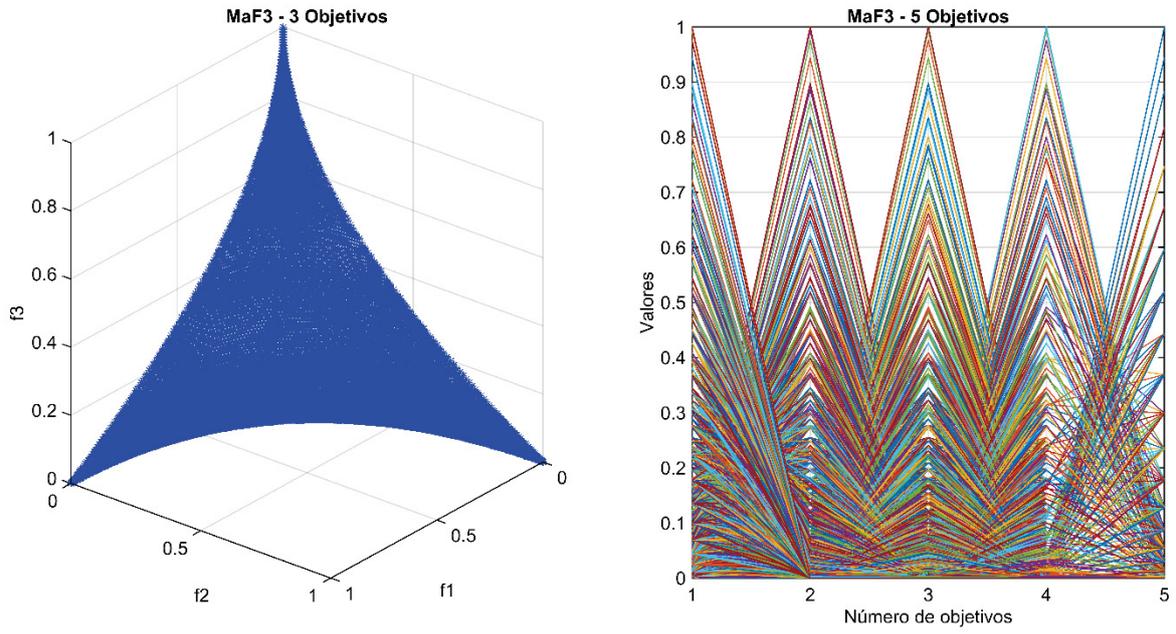
Figura 11 – Fronteiras de Pareto da MaF2.



Fonte: Autora, 2024.

- MaF3

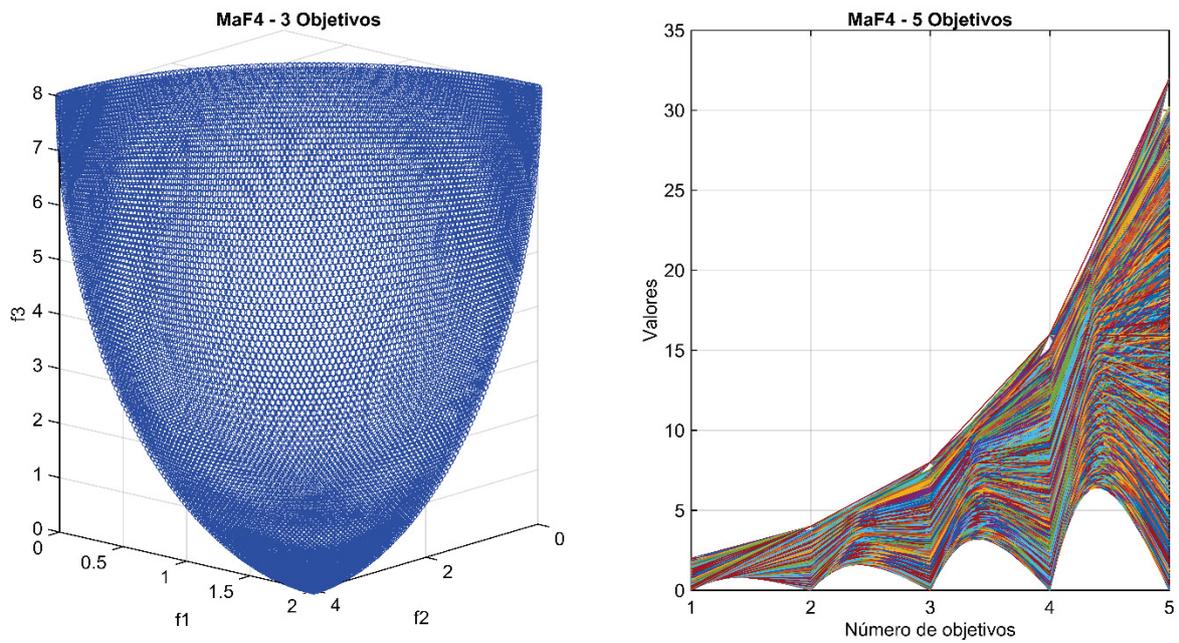
Figura 12 – Fronteiras de Pareto da MaF3.



Fonte: Autora, 2024.

- MaF4

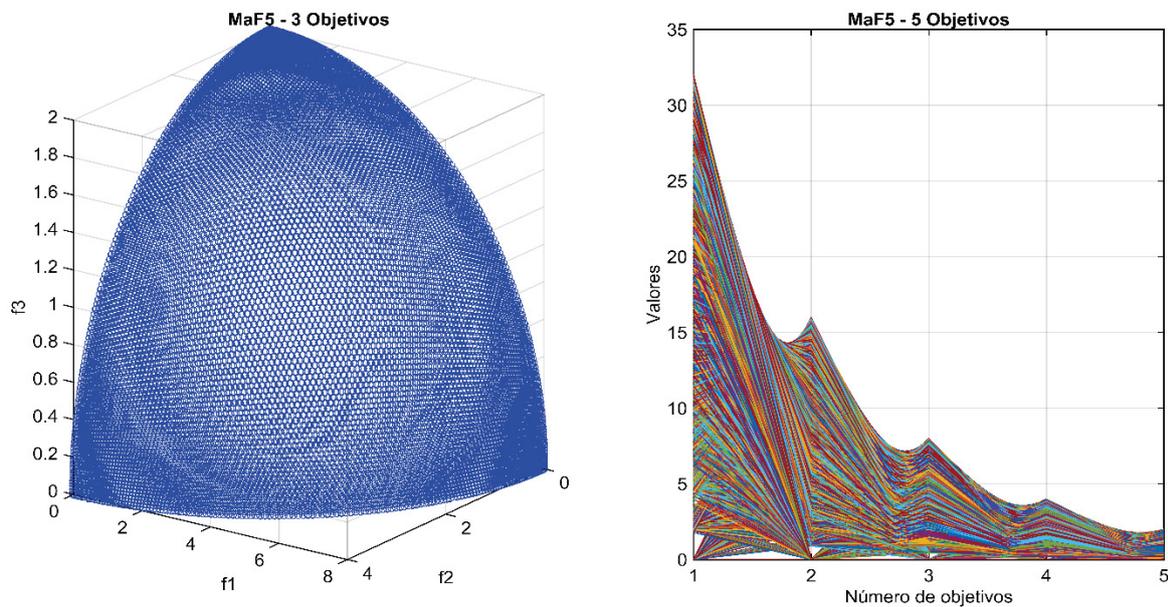
Figura 13 – Fronteiras de Pareto da MaF4.



Fonte: Autora, 2024.

- MaF5

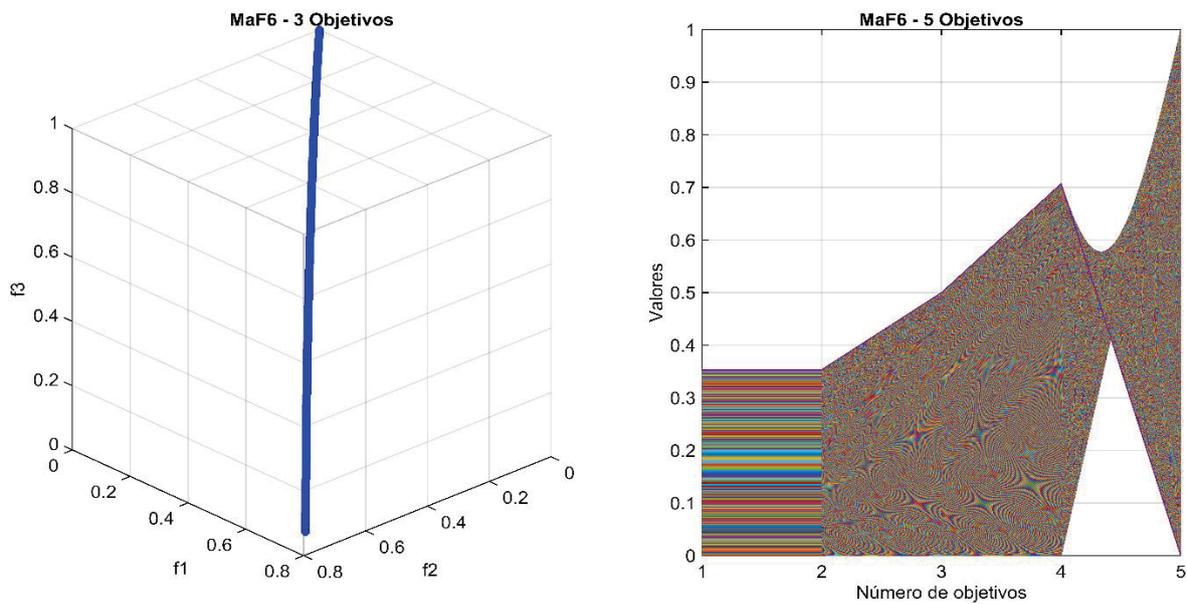
Figura 14– Fronteiras de Pareto da MaF5.



Fonte: Autora, 2024.

- MaF6

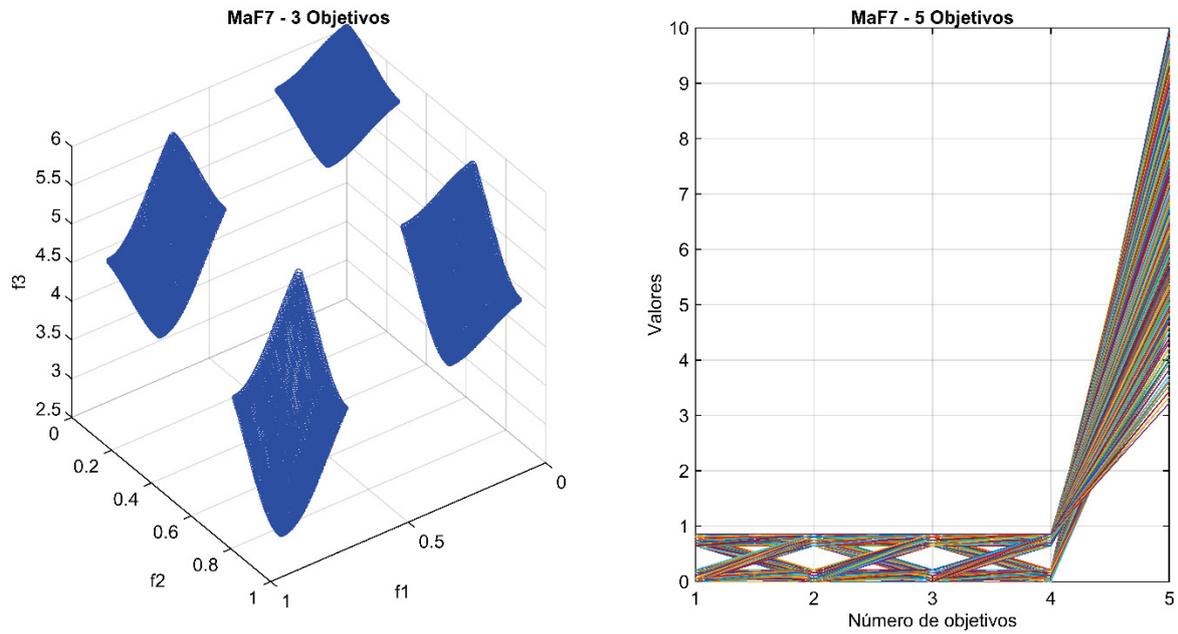
Figura 15– Fronteiras de Pareto da MaF6.



Fonte: Autora, 2024.

- MaF7

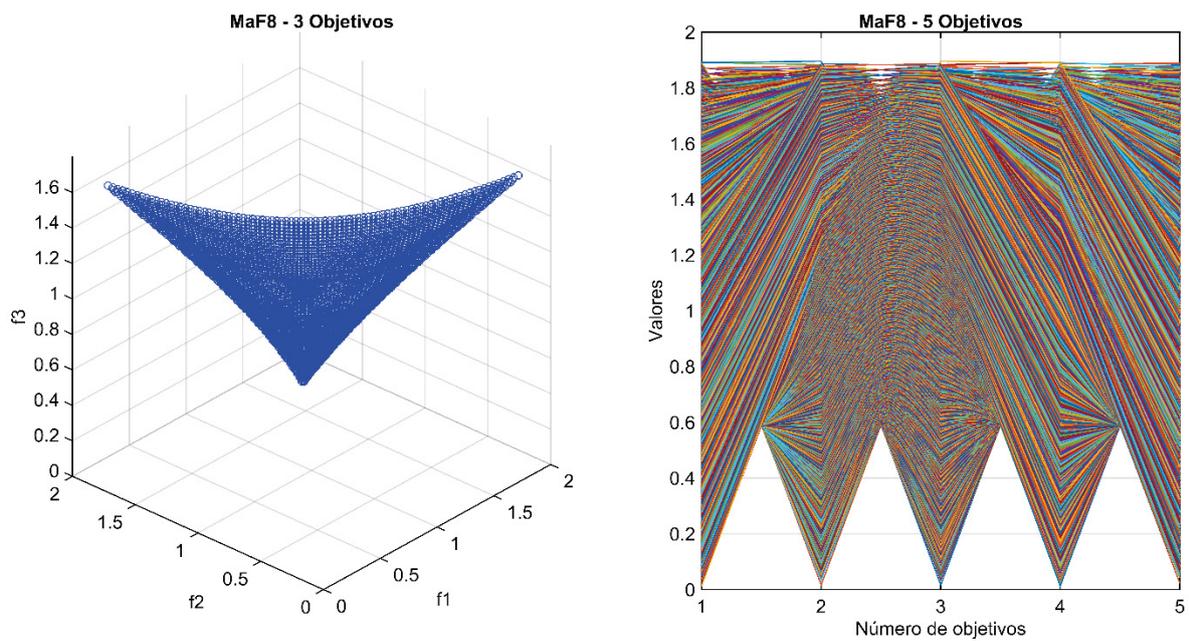
Figura 16 – Fronteiras de Pareto da MaF7.



Fonte: Autora, 2024.

- MaF8

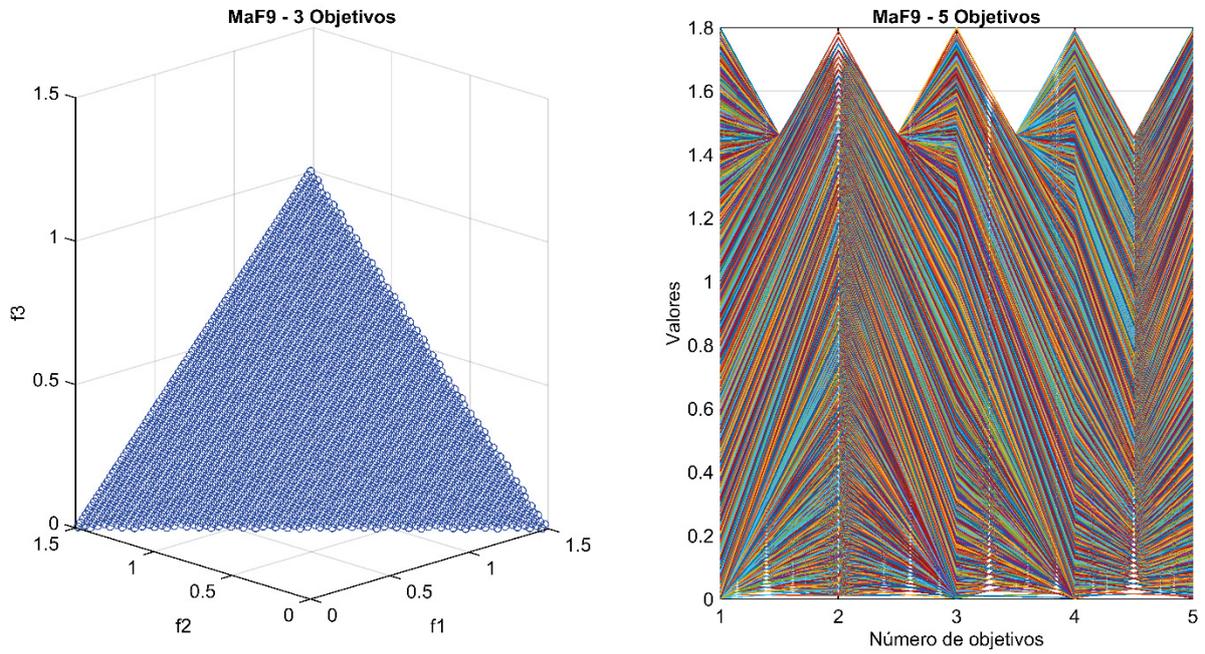
Figura 17 – Fronteiras de Pareto da MaF8.



Fonte: Autora, 2024.

- MaF9

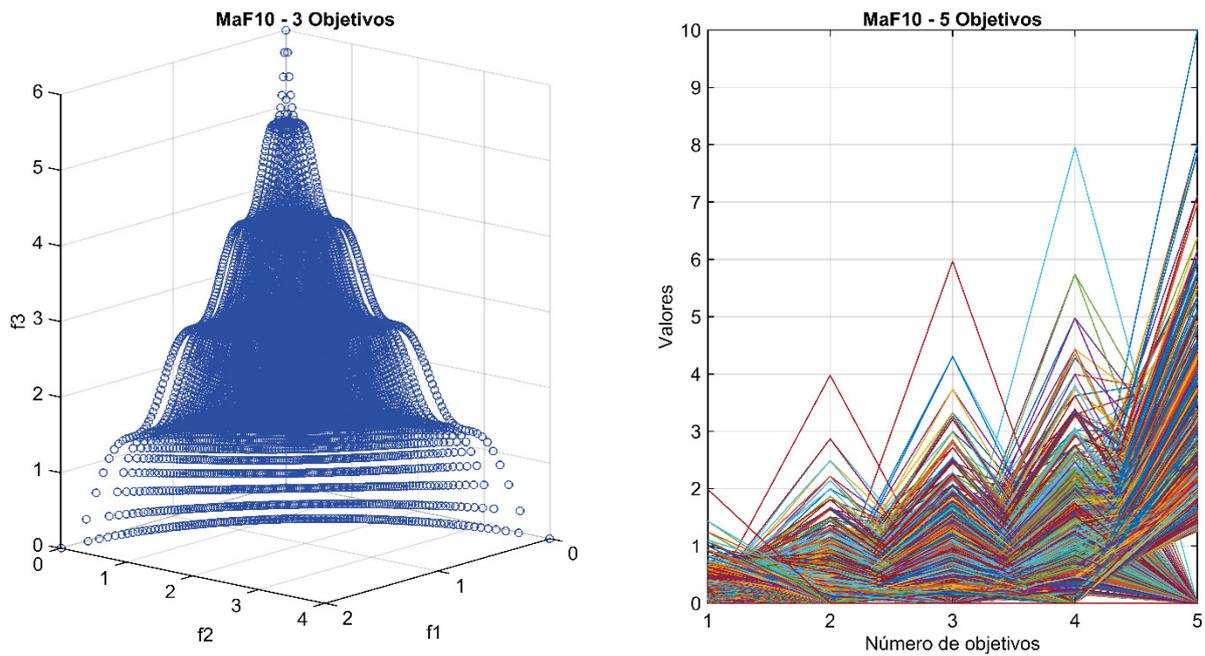
Figura 18 – Fronteiras de Pareto da MaF9.



Fonte: Autora, 2024.

- MaF10

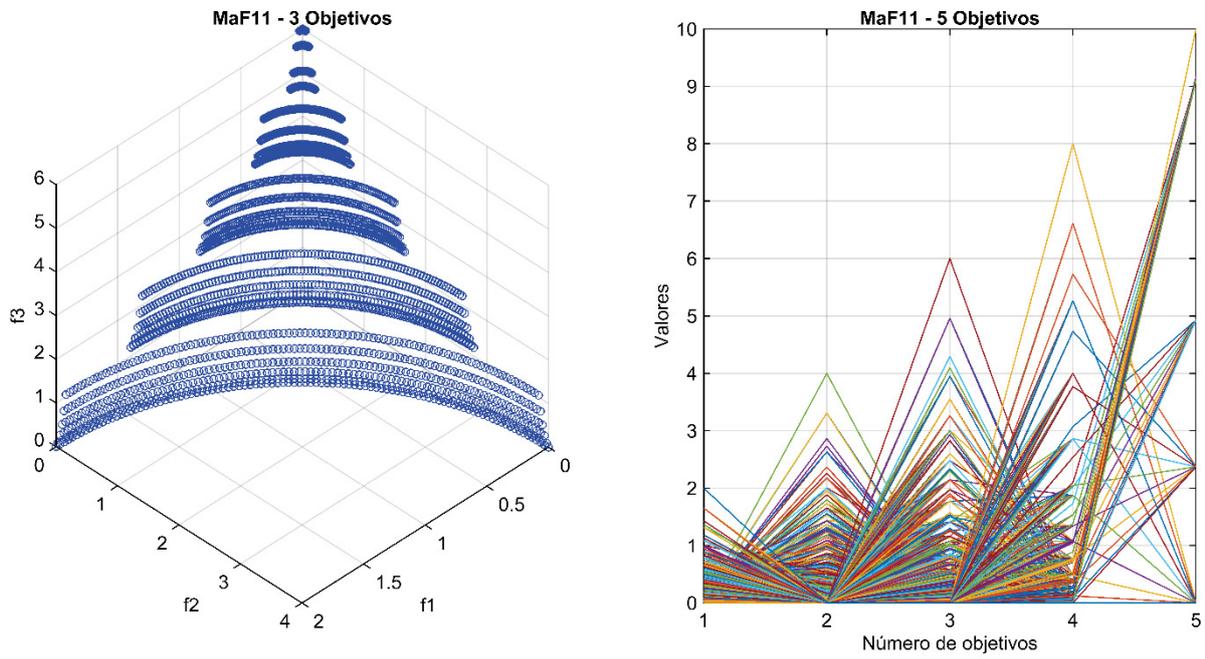
Figura 19 – Fronteiras de Pareto da MaF10.



Fonte: Autora, 2024.

- MaF11

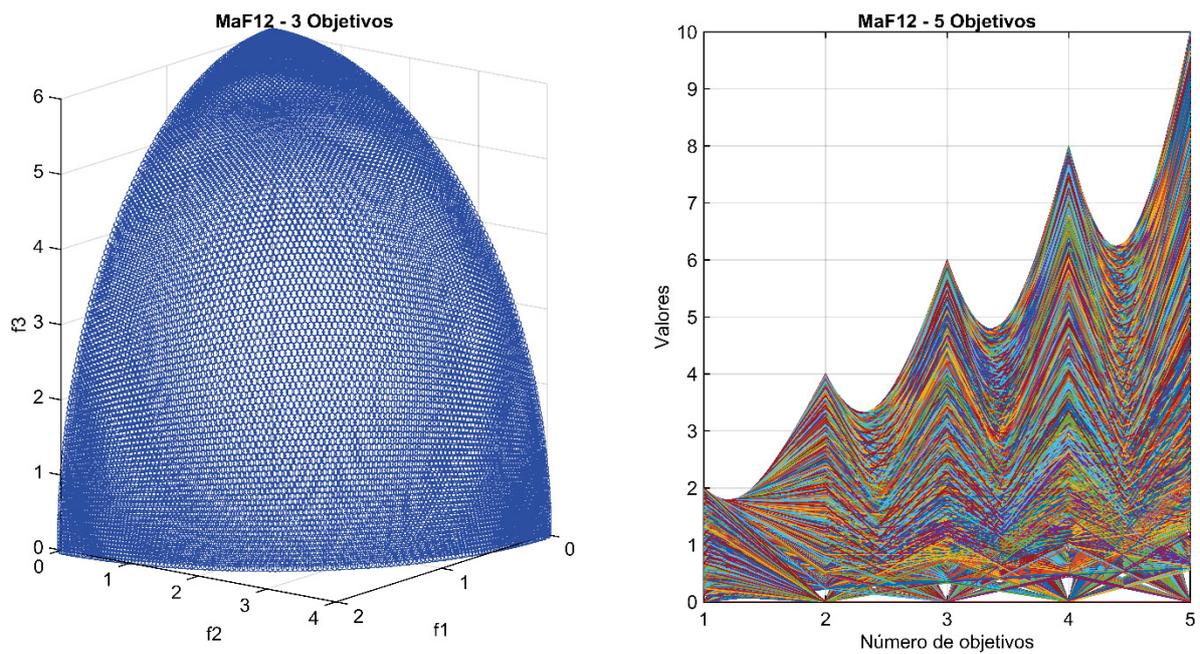
Figura 20 – Fronteiras de Pareto da MaF11.



Fonte: Autora, 2024.

- MaF12

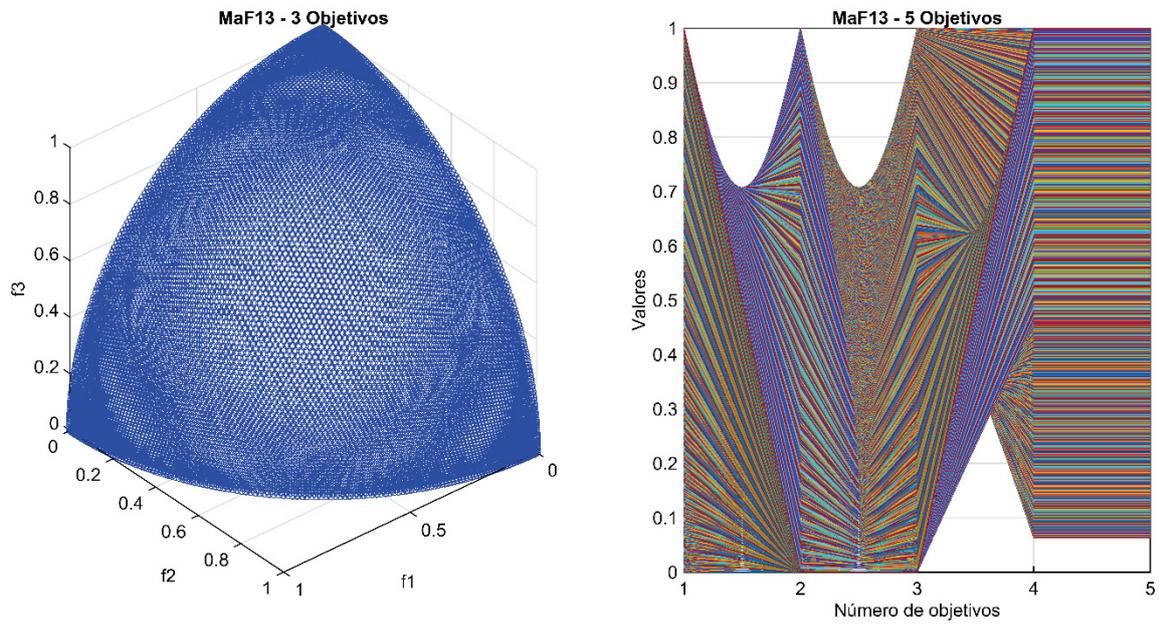
Figura 21 – Fronteiras de Pareto da MaF12.



Fonte: Autora, 2024.

- MaF13

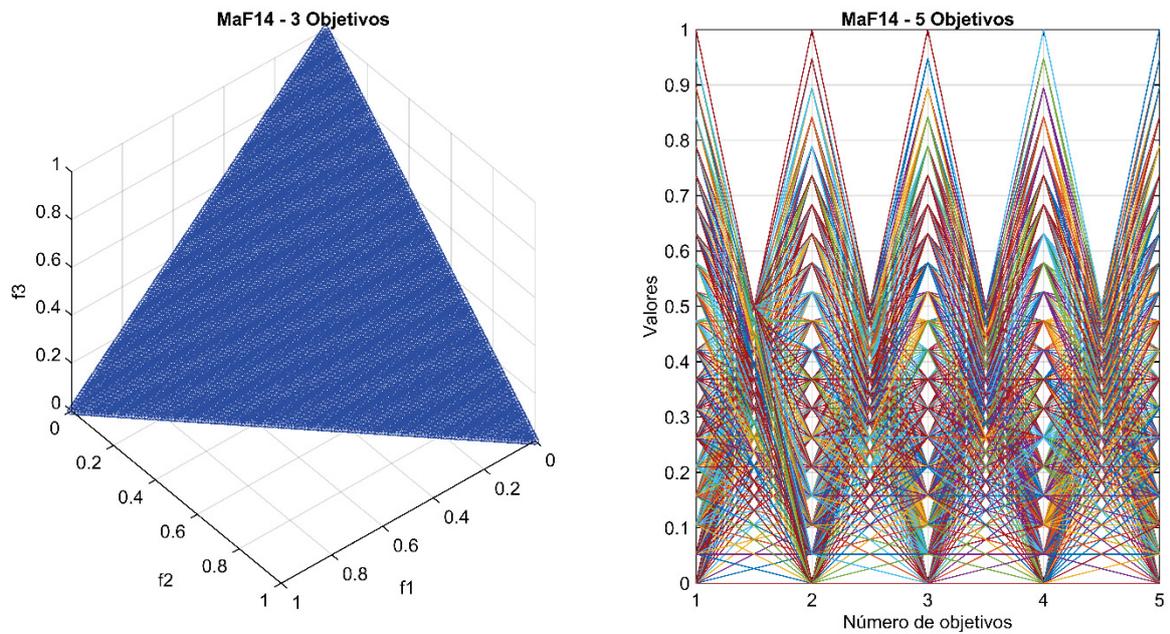
Figura 22 – Fronteiras de Pareto da Ma13.



Fonte: Autora, 2024.

- MaF14

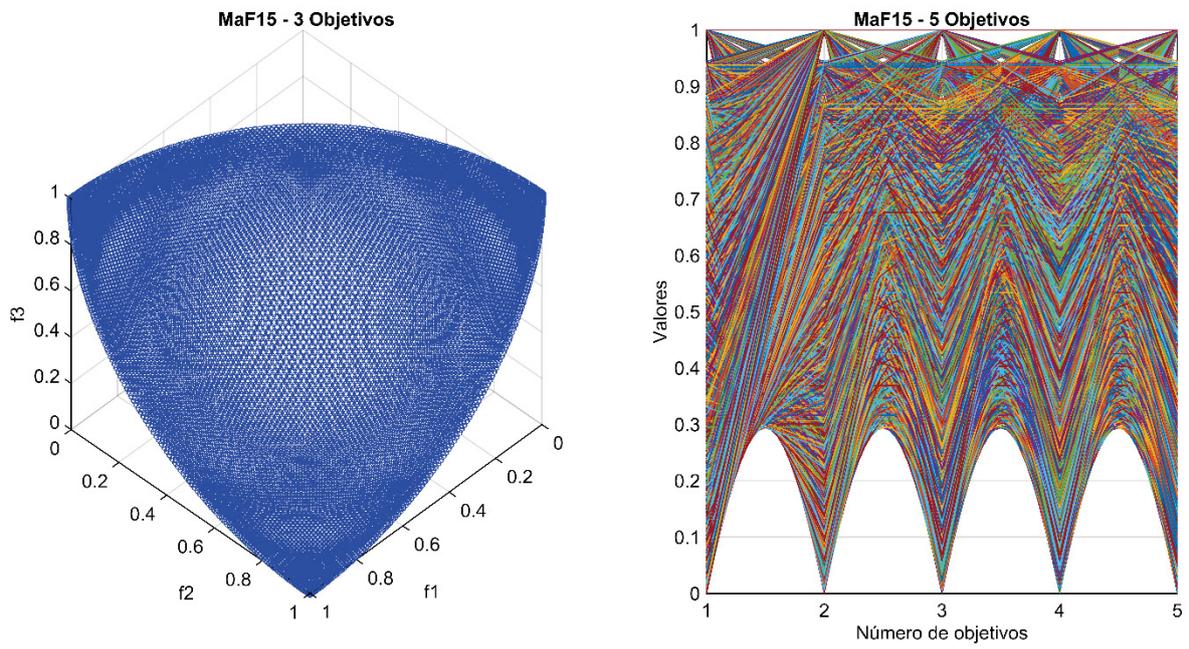
Figura 23 – Fronteiras de Pareto da MaF14.



Fonte: Autora, 2024.

- MaF15

Figura 24 – Fronteiras de Pareto da MaF15.



Fonte: Autora, 2024.

4. METAHEURÍSTICAS PROPOSTAS

Nos últimos anos, muitas metaheurísticas multiobjetivo foram desenvolvidas. A maioria das metaheurísticas multiobjetivo visa gerar simultaneamente, em uma única execução, um conjunto de soluções com que possuam uma boa aproximação de todo o conjunto ótimo de Pareto.

Neste capítulo são apresentadas as duas novas metaheurísticas multiobjetivo propostas. Inicialmente é apresentada a metaheurística multiobjetiva Algoritmo do Leão, baseada em sua versão mono objetivo devido o seu tempo computacional ser ágil em relação aos algoritmos consagrados da literatura, a qual será chamada (em inglês, *Multi-objective Optimization Lion Optimization Algorithm – MOLA*). A segunda metaheurística multiobjetiva proposta é uma metaheurística híbrida que utiliza o Algoritmo do Leão e a Busca Local Iterativa, que exploram e unem as potencialidades de cada uma das metaheurísticas, a qual será chamada (em inglês, *Multi-objective Optimization Lion Algorithm Iterated Search Local – MOLAILS*).

De acordo com o levantamento bibliográfico sobre os algoritmos mono objetivos na literatura, a Tabela 4 cita os artigos e as estratégias para se obter as soluções factíveis.

Tabela 4 – Linha do tempo das principais propostas metaheurísticas do leão.

Ano	Metodologia	Estratégia	Citação
2012	Lion pride optimizer (LPO)	Comportamento do orgulho leão baseado na luta de sobrevivência	Wang et al. (2012)
2012	Lion Algorithm (LA)	Fertilização da leoa, disputa territorial e aquisição territorial	Rajakumar (2012)
2016	Lion Optimization Algorithm (LOA)	Caça realizada pelas leoas	Yazdani; Jolai (2016)

O algoritmo mono objetivo proposto por Rajakumar (2012) tem se mostrado ser uma técnica promissora ao tratar de problemas da identificação de sistemas reais (Bergamini et al. (2017); Bergamini et al. (2019)).

Neste contexto, o presente trabalho utilizou a estratégia de acasalamento, disputa territorial e aquisição territorial para formular a proposta MOLA. Além disso, foi pesquisado sobre quais algoritmo multiobjetivo do leão havia sido propostos e que são apresentados na Tabela 5.

Tabela 5 – Linha do tempo das principais propostas metaheurísticas do leão multiobjetivos.

Ano	Metodologia	Citação
2017	Fgv MO-ADDOFL: Multi-objective-based adaptive dynamic directive operative fractional lion algorithm for data clustering	Chander et al. (2017)
2017	Solving Environmental Economic Load Dispatch with Line Losses in Power System using Lion Pride Optimization Algorithm	Singh Raghuwanshi et al. (2017)
2017	AFL-TOHIP: Adaptive fractional lion optimization to topology-hiding multi-path routing in mobile ad hoc network	Ambekar; Kolekar (2017)
2018	Lion pride optimization algorithm: A meta-heuristic method for global optimization problems	Kaveh; Mahjoubi (2018)
2018	M-LionWhale: multi-objective optimisation model for secure routing in mobile ad-hoc network	Chintalapalli; Ananthula (2018)
2019	MOFPL: Multi-objective fractional particle lion algorithm for the energy aware routing in the WSN	Bhardwaj; Kumar (2019)
2020	Chronological and exponential-based Lion optimisation for optimal resource allocation in cloud	Devagnanam; Elango (2020)
2020	Multi-Objective Optimization for Dynamic Resource Provisioning in a Multi-Cloud Environment using Lion Optimization Algorithm	Chaitra et al. (2020)
2021	Reliable packet transmission in WBAN with	Kathuria; Gambhir (2021)

4.1. ALGORITMO DO LEÃO MULTIOBJETIVO

Os leões são de todas as espécies de felinos selvagens, aqueles que exibem um comportamento social interessante e também mantem o animal mais forte (referido como leão orgulho) na liderança do grupo e exibem altos níveis de cooperação entre si. Os leões têm dois tipos de organização social: residente e nômade.

Um grupo de leões tem normalmente cerca de cinco fêmeas adultas, seus filhotes de ambos os sexos e um ou mais machos adultos. Eles procriam e compartilham um território com interações pacíficas. Eles são chamados de residentes nesta organização social (Mosser, 2008).

O segundo comportamento organizacional é denominado nômade. Neste grupo, os leões se movem aleatoriamente, em pares ou sozinhos. Essa organização social é formada por filhotes machos que são excluídos do grupo residente quando atingem a maturidade sexual ou é o vencido quando há uma disputa territorial. Observe que um leão pode mudar seu estilo de vida; os residentes podem se tornar nômades e vice-versa. Leões nômades podem tentar invadir o território do grupo residente. Na defesa territorial, trava-se uma guerra entre os leões residentes e os leões nômades. Se o leão nômade derrotar o leão orgulho, ele pode ser morto ou é expulso do grupo pelo leão nômade. O leão nômade então se torna o leão orgulho e mata os filhotes de leão vencido. O novo leão orgulho pode imediatamente forçar a leoa a copular para criar sua prole (Rajakumar, 2020).

Neste trabalho é proposto um algoritmo de otimização multiobjetivo do Leão (*Multi-objective Lion Algorithm - MOLA*) inspirado no comportamento social do leão, no que diz respeito à disputa e aquisição territorial. Ele usa essas estratégias para encontrar soluções de Pareto no espaço de busca. O procedimento básico do MOLA é descrito na Figura 25.

No MOLA, o Leão e a Leoa são soluções a serem determinadas, enquanto os Filhotes são soluções derivadas de soluções existentes (Bergamini et al., 2023). A defesa territorial é um processo de avaliação da solução existente (leão residente) e da solução recém-gerada (leão nômade), substituindo a solução existente por uma

nova solução se a nova solução for melhor que a solução existente, além de descartar as soluções derivadas da antiga solução existente.

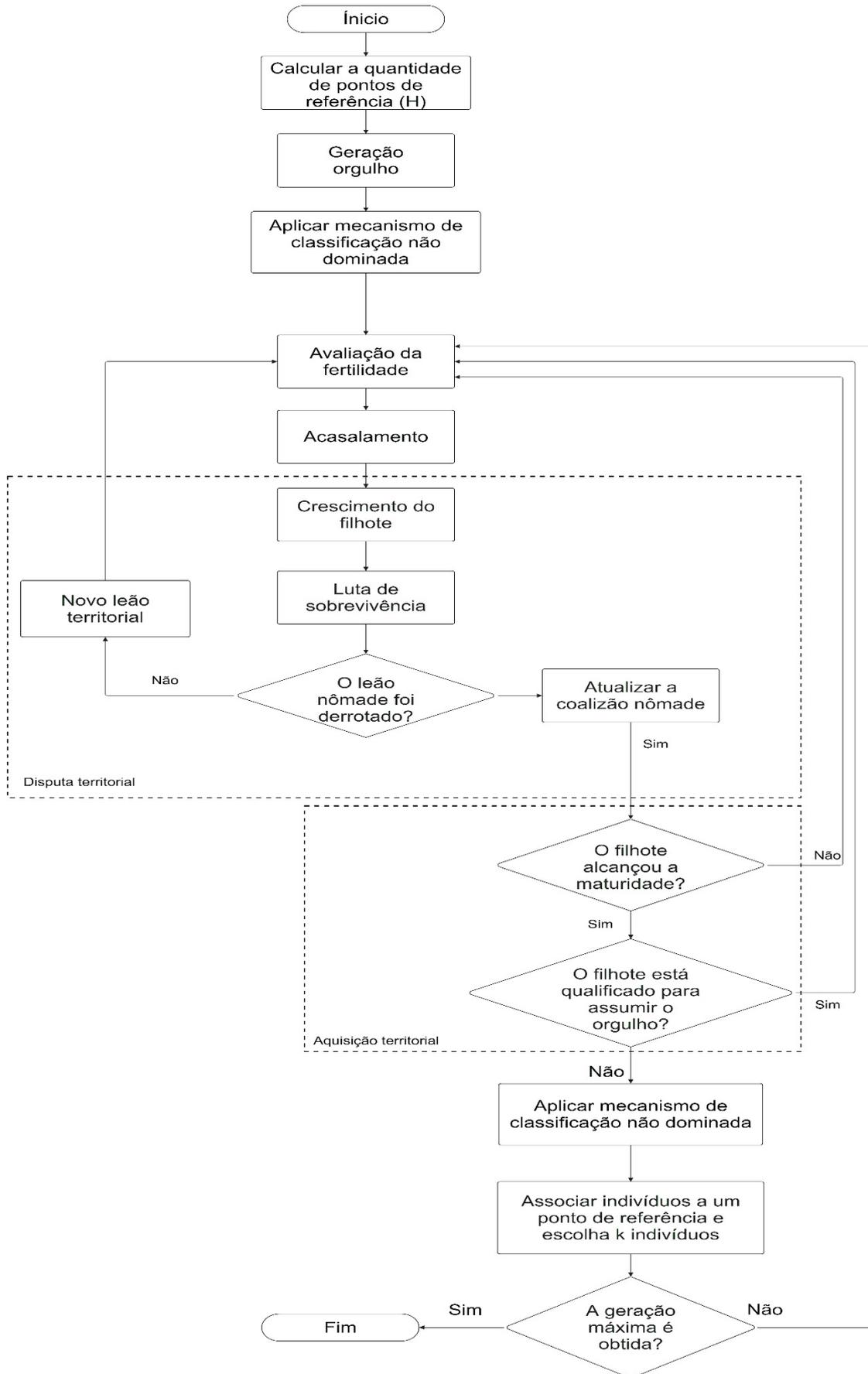
A aquisição territorial é o processo de manter apenas as melhores soluções masculinas e femininas derivadas da nova solução. Ao contrário de outros algoritmos baseados em população, o MOLA possui duas populações: a residente e a nômade, que inicialmente são um conjunto de soluções aleatórias. Essas soluções são então aprimoradas com base nas etapas de fecundação das leas, acasalamento, melhoramento dos filhotes, defesa territorial e tomada de território. Em cada problema, assume-se que os parâmetros dos leões e das leas (x_i) são variáveis contínuas. Cada parâmetro é limitado a um intervalo entre os valores limite inferior e superior. Estes parâmetros podem ser formulados como $LB_i < x_i < UB_i$, onde x_i é o parâmetro, i é o índice de parâmetro $i \in \{1, 2, \dots, Dim\}$, Dim é o número de parâmetros (dimensão do problema), LB_i e UB_i são limites inferior e superior. Cada um desses parâmetros pode ser considerado como uma dimensão do espaço de busca.

Assim, para um problema com parâmetro Dim ou dimensionalidade Dim , um espaço de busca S podendo ser considerado um conjunto R^{Dim} porque nenhum dos parâmetros é infinito. O algoritmo associará um vetor de posição $x = (x_1, x_2, \dots, x_{Dim})$ com um leão ou leoa residente ou nômade.

As modificações propostas ao Algoritmo do Leão original para lidar com problemas multiobjetivo estão resumidas abaixo. Os parâmetros de configuração do algoritmo são os mesmos apresentados no algoritmo LA descrito por Rajakumar (2020), como: o tamanho da população (leões residentes + leas residentes + leões nômades), número de leões residentes, número de leas residentes e número de leões nômades. Posto isto, os tamanhos populacionais de leões, leas e leões nômades são dados como uma porcentagem do tamanho total da população também definidos em Rajakumar (2020) e o número de objetivos é pré-definido de acordo com o problema escolhido pelo(a) codificador(a).

As subseções a seguir ilustram passo a passo as etapas da metaheurística proposta.

Figura 25 – Fluxograma do algoritmo MOLA.



Fonte: Autora, 2024.

4.1.1. Inicialização

Inicialmente, N indivíduos são gerados aleatoriamente no espaço de busca, conforme Equação (4.1).

$$x_{k,i} = LB_i + va(UB_i - LB_i) \quad (4.1)$$

onde $k \in \{1,2, \dots, N\}$, N é o tamanho da população, $i \in \{1,2, \dots, Dim\}$, Dim é o número de variáveis estimadas (dimensionalidade do problema) e va é um número aleatório selecionado da distribuição uniforme no intervalo $[0,1]$.

4.1.2. Geração da população residente e nômade

O processo de geração do orgulho irá definir a geração do leão nômade, onde o vetor de solução e a formulação do leão são semelhantes. O elemento vetorial de x^{nomad} , x^{male} e x^{female} é declarado como xL^{nomad} , xL^{male} e xL^{female} são contabilizados como números inteiros aleatórios com limites máximo e mínimo dados na Equação (4.3). Para $N > 1$, sendo $l = 1; 2 \dots, L$, o comprimento L do Leão pode ser visto na Equação (4.2).

$$L = \begin{cases} N; & N > 1 \\ m; & \text{caso contrário} \end{cases} \quad (4.2)$$

sendo, N e m número inteiros para decidir o tamanho de leões. Enquanto $N = 1$, a busca com leão codificado binário é desenvolvida no algoritmo e, portanto, a geração de elementos do vetor pode ser 0 ou 1.

$$g(x) \in (x_l^{min}, x_l^{max}) \quad (4.3)$$

$$m \% 2 = 0 \quad (4.4)$$

As Equações (4.3) e (4.5) conseguem garantir que o leão gerado esteja em formato binário dentro dos limites.

$$g(x_l) = \sum_{l=1}^L x_l 2^{\left(\frac{L}{2} - l\right)} \quad (4.5)$$

E a Equação (4.4), irá garantir o número de bits a serem utilizados para gerar o número de frações reais e decimais da solução para que o número par seja garantido.

Posto isto, para o presente trabalho foi utilizado a população inicial $(x_{k,i})$, onde 30% da população inicial será destinada aos leões residentes (x^{male}), 60% da população inicial será destinada para as leoas (x^{female}) e 10% da população inicial será destinada aos leões nômades (x^{nomad}). Nesta etapa, apenas valores são atribuídos a um leão nômade e para os demais leões nômades estes terão seus valores atribuídos na etapa de defesa territorial.

4.1.3. Geração de pontos de referência

Os pontos de referência são um método alternativo para encontrar vários pontos ótimos de Pareto para um problema geral de otimização multicritério não linear. Esses pontos capturam coletivamente o compromisso entre os vários objetivos conflitantes e produzem um conjunto de pontos distribuídos uniformemente no conjunto de Pareto, dado um conjunto de parâmetros distribuídos uniformemente.

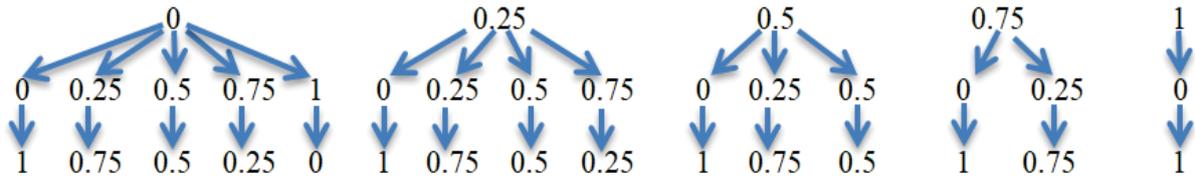
No presente trabalho é utilizada a proposta de Das e Dennis (1998) que colocam os pontos em um hiperplano normalizado, que é igualmente inclinado e intercepta todos os eixos objetivos em um em cada eixo.

Se forem consideradas m divisões ao longo de cada objetivo, o número total de pontos de referência (H) num problema de objetivo M é dado pela Equação (4.6).

$$H = \binom{M+m-1}{m} \quad (4.6)$$

Para problemas de $M = 3$ e $m = 4$ (quatro divisões para cada eixo objetivo) 15 pontos de referência serão criados. Então, $H = \binom{3+4-1}{4} = 15$. A Figura 26 ilustra os pontos de referências obtidos entre espaço e divisões de $\gamma = 0,25$.

Figura 26 – Exemplo de pontos de referência para $M = 3$ e $m = 4$.



Fonte: Autora, 2024.

Para gerar esses pontos de referência de modo uniforme e estruturado para um espaço objetivo de qualquer dimensão M usando qualquer número de divisões m fornecido pelo usuário, esse algoritmo cria cada um dos pontos de referência de modo recursivo a partir de pontos referência parciais com dimensões menores que M , até que finalmente todos os pontos de referência atinjam a dimensão completa M .

4.1.4. Avaliação de fertilidade

A avaliação da fertilidade pode ajudar as leas (x^{female}) a melhorarem suas soluções. Com isto, uma leoa residente (x_k^{female}) é escolhida aleatoriamente e uma cópia dela é realizada em ($x_k^{female+}$), então são realizadas tentativas de se melhorar a aptidão da leoa através de atualizações, conforme indicado nas Equações (4.7) à (4.9).

$$(x_l^{female+}) = \begin{cases} (x_{k1}^{female+}); & \text{se } l = k1 \\ (x_l^{female}), & \text{caso contrário} \end{cases} \quad (4.7)$$

$$x_{k1}^{female+} = \min [x_{k1}^{max}, \max (x_{k1}^{min}, \nabla_{k1})] \quad (4.8)$$

$$\nabla_{k1} = [x_{k1}^{female} + (0,1r_2 - 0,05)(x_{k1}^{male} - r_1x_{k1}^{female})] \quad (4.9)$$

sendo, $x_l^{female+}$ e $x_{k1}^{female+}$ as l^{th} e $k1^{th}$ elementos vetoriais de $x_k^{female+}$, respectivamente, $k1$ é um número inteiro aleatório gerado de distribuição uniforme no intervalo $[1, Dim]$, ∇_{k1} é a função de atualização, r_1 e r_2 são números inteiros aleatórios gerados no intervalo $[0,1]$.

Se $f(x_k^{female+})$ dominar $f(x_k^{female})$, x_k^{female} se torna $x_k^{female+}$ devido ao seu aprimoramento, onde $f(x_k)$ é a função multiobjetivo avaliada. Essa tentativa de melhora se repete certo número de vezes definido pelo usuário. Se não houver

dominância entre $f(x_k^{female+})$ e $f(x_k^{female})$ um número aleatório dentro o intervalo de $[0,1]$ é gerado, e se o valor gerado for maior que 0,5 então $f(x_k^{female})$ é substituído por $f(x_k^{female+})$. Se não houver $x_k^{female+}$ para substituir x_k^{female} durante todo o processo de atualização, pode-se inferir que x_k^{female} ainda é fértil o suficiente para produzir descendentes.

4.1.5. Acasalamento

O acasalamento é o processo de encontrar novas soluções por meio de cruzamento e mutação a partir das soluções existentes e inclui agrupamento de gênero para encontrar diversificação entre as soluções. Na literatura, numerosos trabalhos têm discutido as operações de cruzamento e mutação e sua necessidade em algoritmos evolutivos (Alhijawi e Awajan, 2023). Posto isto, as subseções a seguir mostram os processos de cruzamento e mutação.

4.1.5.1. Cruzamento

Nesta etapa, x^{male} e x^{female} geram os descendentes x^{cubs} (filhotes), escolhendo aleatoriamente um $x_k^{female} \in x^{female}$ e um $x_k^{male} \in x^{male}$ como pais. Para obter as soluções descendentes, o operador *Simulated Binary Crossover* (SBX) proposto por Deb et al. (1994) foi utilizado seguindo as seguintes etapas.

- i) Um número aleatório é definido $\mu \in [0,1]$;
- ii) β é calculado pela Equação (4.10)

$$\beta = \begin{cases} (2\mu)^{1/(n+1)} & \mu \leq 0.5 \\ \left(\frac{1}{2(1-\mu)}\right)^{1/(n+1)} & \text{caso contrário} \end{cases} \quad (4.10)$$

onde n é qualquer número real não negativo definido pelo usuário (chamado de índice de distribuição). O valor de β é determinado de modo que a área sob a curva de zero a β seja μ .

Os filhotes gerados são calculados usando as Equações (4.11) e (4.12).

$$x_k^{cubs} = \frac{1}{2}(x_k^{male} + x_k^{female}) + \frac{1}{2}\beta(x_k^{male} - x_k^{female}) \quad (4.11)$$

$$x_{k+1}^{cubs} = \frac{1}{2}(x_k^{male} + x_k^{female}) - \frac{1}{2}\beta(x_k^{male} - x_k^{female}) \quad (4.12)$$

4.1.5.2. Mutação

Após o cruzamento, o processo de mutação é aplicado aos filhotes (x_k^{cubs}). Para cada filhote, um número inteiro aleatório $i \in [1 Dim]$ é gerado, e nesta posição, uma distribuição uniforme é aplicada. A Equação (4.13) ilustra o passo da mutação.

$$x_{k,i}^{cubs} = x_{k,i}^{cubs} r_3 \quad (4.13)$$

onde r_3 é um número aleatório gerado por uma distribuição uniforme no intervalo $[0,1]$.

4.1.6. Defesa territorial

A defesa territorial é a etapa em que são avaliadas as soluções existentes (leões orgulho) e a solução recém-gerada (leão nômade). A solução existente é substituída por uma nova solução, se a nova solução for melhor que a solução existente. As subseções a seguir ilustram essa etapa.

4.1.6.1. Crescimento dos filhotes

O crescimento dos filhotes é realizado fazendo uma nova mutação nos filhotes. Em cada filhote, para todas as posições, é aplicada uma distribuição normal. Distribuição normal é uma distribuição de probabilidade simétrica em relação à média, mostrando que dados próximos à média têm ocorrência mais frequente do que dados distantes da média. A Equação (4.14) mostra o crescimento dos filhotes.

$$x_k^{cubs} = N(x_k^{cubs}, \delta) \quad (4.14)$$

onde $N(\mu, \delta)$, é uma distribuição normal sendo μ a média e δ é o desvio padrão.

4.1.6.2. Luta de sobrevivência

A comparação entre $f(x_k^{male})$ e $f(x_k^{nomad})$ serve para verificar quem é mais forte. Se $f(x_k^{male})$ dominar $f(x_k^{nomad})$, aumente a idade de x_k^{cubs} em 1 e atualize x_k^{nomad} . A atualização do x_k^{nomad} é realizada por uma mutação, e a mutação é realizada usando distribuição uniforme e distribuição normal. Se $f(x_k^{nomad})$ dominar $f(x_k^{male})$, x_k^{nomad} se torna x_k^{male} e vice-versa, e os x_k^{cubs} são mortos. Se $f(x_k^{male})$ não é dominado por $f(x_k^{nomad})$ e $f(x_k^{nomad})$ não é dominado por $f(x_k^{male})$ um número aleatório r entre $[0, 1]$ é gerado, se este número for maior que 0,5 troca-se x_k^{male} por x_k^{nomad} . As Equações (4.15) a (4.19) sintetizam o exposto anteriormente.

$$f(x_k^{male}) < f(x_k^{nomad}) \Rightarrow \text{aumente a idade de } x_k^{cubs} \text{ em } 1 \quad (4.15)$$

$$x_1^{nomad} = LB_i + va(UB_i - LB_i) \quad (4.16)$$

$$x_k^{nomad} = N(x_k^{nomad}, \delta) \text{ para } k \neq 1 \quad (4.17)$$

$$f(x_k^{male}) > f(x_k^{nomad}) \Rightarrow \begin{cases} x_k^{male} = x_k^{nomad} \text{ e } x_k^{nomad} = x_k^{male} \\ \text{matar os } x_k^{cubs} \end{cases} \quad (4.18)$$

$$f(x_k^{male}) = f(x_k^{nomad}) \Rightarrow x_k^{male} = x_k^{nomad} \text{ se } r > 0,5 \quad (4.19)$$

4.1.7. Aquisição territorial

A aquisição territorial é a etapa de manter apenas as melhores soluções, derivadas dos leões, leas e filhotes maduros e definir das soluções existentes quem serão os residentes. Inicialmente, no processo de Aquisição Territorial verifica se os filhotes atingiram a idade de maturidade (em torno de 3 anos). Quando os filhotes atingirem essa idade, eles passam a ser considerados leões e podem começar a lutar com os leões residentes para provar sua força.

4.1.7.1. Criação da população auxiliar – Filhotes maduros

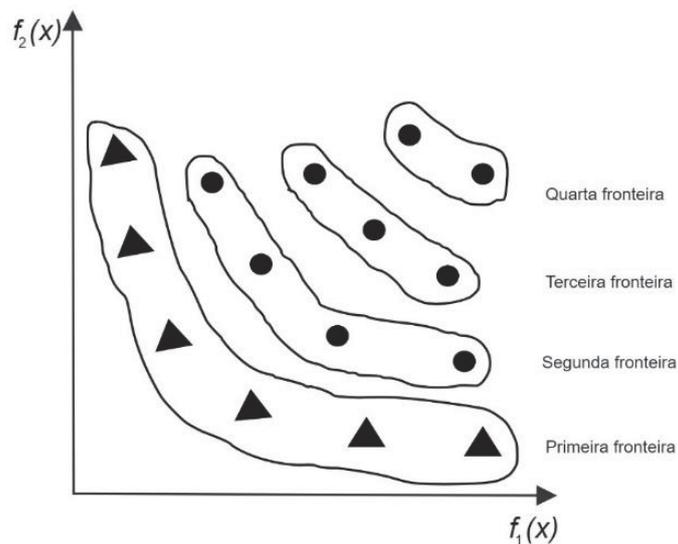
Nesta etapa, inicialmente é verificada se os filhotes atingiram a idade de maturidade. Caso negativo uma população auxiliar será constituída pelos leões e leas residentes. Caso os filhotes atinjam a idade de maturidade a população auxiliar será composta por leões, leas e filhotes.

A população auxiliar será analisada conforme a subseção a seguir.

4.1.8. Checando não dominância

A população auxiliar criada na subseção anterior passa por uma análise de dominância semelhante à proposta no NSGA III por Deb e Jain (2014). Nesse processo são criadas frentes não dominadas. Inicialmente a primeira frente possui as soluções não dominadas da população. As demais frentes são criadas retirando as soluções não dominadas da frente anterior e o procedimento é repetido, criando as frentes subsequentes. A Figura 27 ilustra o procedimento de checagem de não dominância.

Figura 27 – Exemplo de checagem de não dominância.



Fonte: Autora, 2024

4.1.9. Associando cada indivíduo a um ponto de referência

Inicialmente, determina-se os valores máximo e mínimo de cada função objetivo utilizando a população auxiliar. Com isso, tem-se o vetor $y = (f_1^{min}(x), f_2^{min}(x), \dots, f_n^{min}(x))$ e o vetor $w = (f_1^{max}(x), f_2^{max}(x), \dots, f_n^{max}(x))$ de tamanho Dim construído.

Esses n pontos ($n = Dim$) do vetor y e w são usados para constituir um hiperplano n – dimensional. As intercepções do hiperplano construído pelos pontos extremos e pelos eixos são os pontos dados pela Equação (4.20).

$$a_k = w_k - y_k \quad (4.20)$$

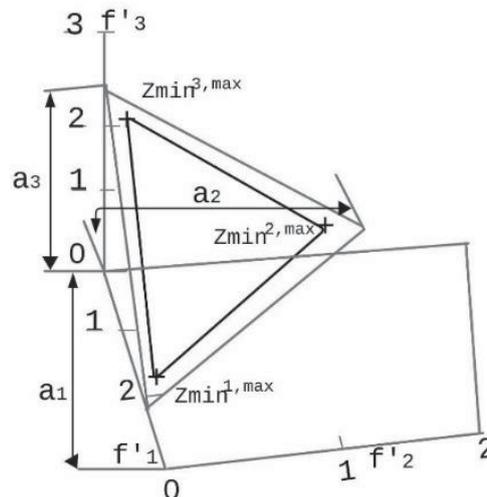
Então a k – ésima função objetivo pode então ser normalizada conforme Equação (4.21)

$$f_k^n(x) = \frac{f_k(x) - y_k}{a_k} \quad (4.21)$$

onde $f_k^n(x)$ é o k -ésimo objetivo normalizado da função objetivo.

A Figura 28 ilustra o procedimento para o caso de um problema com três objetivos. Onde $Zmin^{i,max}$ é o mínimo dos máximos do objetivo i da população auxiliar, e $f_i' = f_i(x) - y_i^{min}$.

Figura 28 – Formação de hiperplano a partir dos pontos extremos para um problema de três objetivos.



Fonte: Adaptada de DEB e JAIN (2014).

Ao analisar a Figura 28, os pontos de referência definidos pelo usuário são mapeados no hiperplano normalizado construído a partir da Equação (4.21). A normalização e a criação do hiperplano são realizadas a cada geração utilizando os pontos extremos que já foram encontrados no início da simulação do algoritmo, esse procedimento mantém adaptativamente a diversidade no espaço de busca abrangido pelos membros de f_k^{min} a cada geração. Dessa maneira, o MOLA soluciona problemas com uma frente Pareto-ótima cujos valores objetivos podem ser escalados de modo distinto.

Portanto agora será necessário associar cada solução da população a um ponto de referência. Para isso, foi utilizado o método da abordagem de interseção

de limites baseada em penalidades (PBI) (Coello Coello; Lechuga, 2002) que é definido por uma reta perpendicular (linha de referência) que liga os pontos de referência do hiperplano à origem. Em seguida, a distância perpendicular das soluções da população a cada uma das linhas de referência será calculada. O ponto de referência, cuja linha de referência está mais próxima de um indivíduo da população no espaço objetivo normalizado pode ser considerado “associado” a uma solução da população. Seja x uma solução da população e T o ponto de interseção de x com a linha de referência. A distância xT é calculada e definida como c_2 , a distância ZT é calculada e definida como c_1 sendo a distância do ponto de referência Z ao ponto de interseção T . Um parâmetro de penalidade θ é considerado como uma entrada do usuário para quantificar a penalidade aplicada à solução pelo desalinhamento de x com o vetor de referência. O PBI é calculado conforme mostrado nas Equações (4.22) a (4.24)

$$\min_x f^n(x|Z, b) = c_1 + \theta c_2 \quad (4.22)$$

sujeito a:

$$c_1 = \frac{|(Z - f^n(x))b|}{|b|} \quad (4.23)$$

$$c_2 = \left| f^n(x) - \left(Z - c_1 \frac{b}{|b|} \right) \right| \quad (4.24)$$

onde, b é o vetor peso $R^{Dim \times 1}$ tal que $\sum_{i=1}^{Dim} b(i) = 1$.

Após realizado o processo de associação, pode-se observar que um ponto de referência pode ter nenhuma, uma ou várias soluções da população associados a ele.

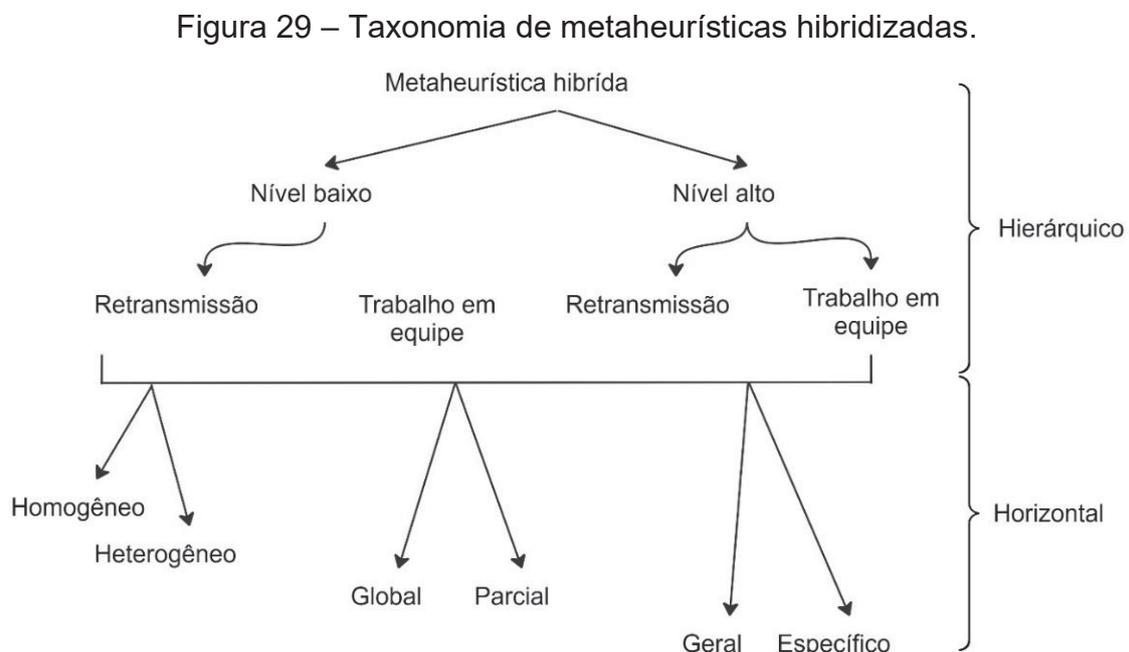
Para os pontos de referência que possuem mais de uma solução associada a ele, é necessário realizar um processo para que o mesmo possua somente uma solução associada. Para isto, a menor distância perpendicular das soluções à linha de referência é a escolhida.

Os casos que não há nenhuma solução associado a um ponto de referência, para que a população permaneça com o mesmo tamanho são geradas novas soluções de forma aleatória dentro do espaço de busca. Pois o tamanho da população é igual ao número de pontos de referência.

4.2. ALGORITMO DO LEÃO COM BUSCA LOCAL MULTIOBJETIVO

Quando se tem metaheurísticas construídas que não seguem os paradigmas tradicionais, isto é, combinam diferentes componentes algoritmos até mesmo provenientes de outras áreas de pesquisa tem-se as chamadas metaheurísticas híbridas (Blum et al., 2013).

A Figura 29 ilustra a taxonomia de hibridização proposta por Talbi (2002). Neste contexto se tem duas estruturas: a hierárquica e a horizontal. Na primeira, quatro classes de hibridização são esquematizadas em diferentes níveis. Na segunda, por sua vez, as classes estão num mesmo nível e qualificam as classes da primeira abordagem.



Fonte: Adaptada de Talbi (2002).

Conforme a estrutura hierárquica, metaheurísticas híbridas podem ser divididas em duas sub-classes: de baixo ou alto nível. Na hibridização de baixo nível, um procedimento específico de uma metaheurística (mais externa) é realizado por outra metaheurística (mais interna), estabelecendo uma relação de composição. Na classe alto nível, não há esta relação direta de composição entre as metaheurísticas. Estas são, portanto, autônomas. No terceiro nível da abordagem hierárquica, tem-se duas novas sub-classes: retransmissão e trabalho em equipe. Na primeira, duas ou mais metaheurísticas são aplicadas sequencialmente, cada

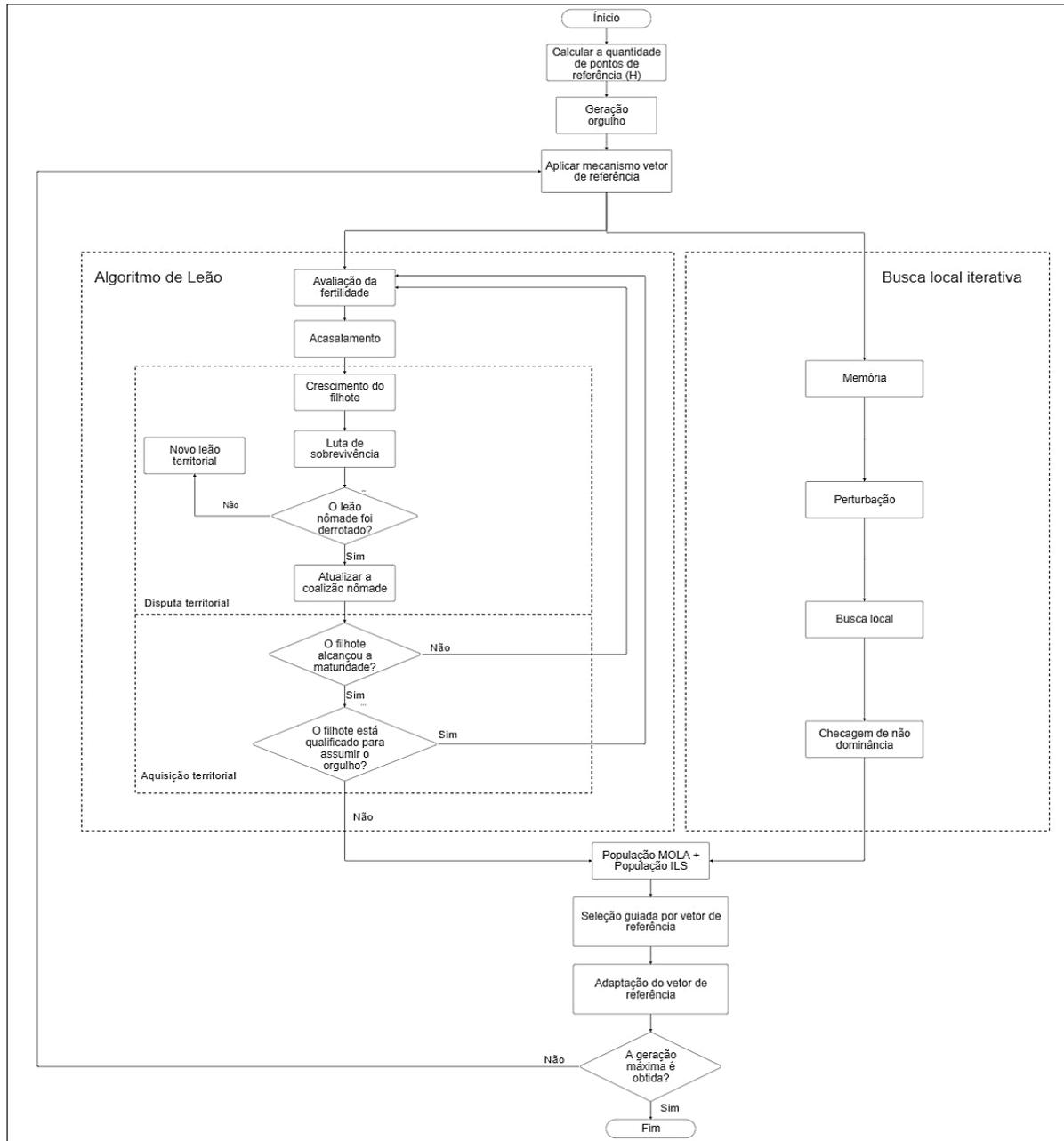
uma utilizando como entrada a saída da anterior; na segunda, por outro lado, tem-se duas ou mais metaheurísticas executando em paralelo, caracterizando-se como agentes cooperativos, onde cada agente efetua sua própria busca no espaço solução.

A estrutura horizontal é dividida em: homogêneo e heterogêneo. Essas classes são adjetivos que qualificam variados pontos de vista das hibridizações, inclusive aquelas da abordagem hierárquica. Uma hibridização é dita homogênea se todas as partes envolvidas utilizam a mesma metaheurística, mesmo que com parâmetros distintos. Se as metaheurísticas envolvidas forem diferentes, diz-se que a hibridização é heterogênea. De outro ponto de vista, pode-se classificar uma hibridização como global ou parcial: a primeira refere-se às hibridizações nas quais todas as partes exploram todo o espaço de busca; a segunda, por sua vez, refere-se às hibridizações onde o problema é decomposto em sub-problemas com espaço de busca distintos e cada um é otimizado por uma metaheurística diferente. Note, entretanto, que a decomposição em sub-problemas não cria problemas de otimização diferentes do original, mas apenas problemas menores. Por fim, distingue-se ainda a classe geral e a especialista: na primeira, todas as metaheurísticas resolvem o mesmo problema de otimização; na segunda, as metaheurísticas resolvem problemas de otimização com formulações distintas.

Neste trabalho é proposta uma nova de metaheurística híbrida, que segundo a taxionomia apresentada se enquadra como sendo de alto nível, trabalho em equipe, heterogênea, global e geral. O algoritmo de otimização multiobjetivo do Leão com Busca Local Iterativa (*Multi-objective Optimization Lion Algorithm Iterated Search Local – MOLAILS*) é inspirado no comportamento social do leão, no que diz respeito à disputa e aquisição territorial e no refinamento dessas soluções utilizando o algoritmo da Busca Local Iterativa. Essas estratégias são utilizadas de forma cooperativa para encontrar soluções de Pareto no espaço de busca.

O procedimento básico do *MOLAILS* é descrito na Figura 30. O algoritmo *MOLAILS*, inicializa pela definição das variáveis utilizadas no Algoritmo do Leão (LA) e no algoritmo da Busca Local Iterativa (ILS). Em seguida é definido o conjunto de pontos de referência que serão utilizados para o cálculo dos vetores de referência.

Figura 30– Fluxograma do algoritmo cooperativo proposto – MOLAILS.



Fonte: Autora, 2024.

A seguir, assume-se que os parâmetros dos leões e das leoas (x_i) são variáveis contínuas. Cada parâmetro é limitado a um intervalo entre os valores limite inferior e superior. Estes parâmetros podem ser formulados como $LB_i < x_i < UB_i$, onde x_i é o parâmetro, i é o índice de parâmetro $i \in \{1, 2, \dots, Dim\}$, Dim é o número de parâmetros (dimensão do problema), LB_i e UB_i são limites inferior e superior. Cada um desses parâmetros pode ser considerado como uma dimensão do espaço de busca. Com isso, gera-se a população inicial. Essa população então passa pela etapa de associação com os vetores de referência, gerando a definição dos leões e

leões residentes. Esses leões e leões então são utilizados nas etapas respectivamente do LA e do ILS.

As etapas seguintes são realizadas de forma paralela pelo LA e ILS, quando esses algoritmos geram uma população essas são guardadas em um arquivo que então é utilizado o procedimento de vetor de referência e a sua adaptação, gerando a população para o próximo passo.

As subseções a seguir ilustram passo a passo as etapas da metaheurística proposta.

4.2.1. Inicialização

Inicialmente, N indivíduos são gerados aleatoriamente no espaço de busca, conforme Equação (4.1).

4.2.2. Geração da população residente e nômade

Da população inicial $(x_{k,i})$, 30% da população inicial será destinada aos leões residentes (x^{male}) , 60% da população inicial será destinada para as leões (x^{female}) e 10% da população inicial será destinada aos leões nômades (x^{nomad}) . Nesta etapa, apenas valores são atribuídos a um leão nômade e para os demais leões nômades estes terão seus valores atribuídos na etapa de defesa territorial.

4.2.3. Geração dos vetores de referência

Os vetores de referência utilizados podem ser considerados como vetores unitários dentro do primeiro quadrante, sendo a origem do ponto inicial. Em outras palavras, o vetor unitário pode ser facilmente gerado dividindo um vetor arbitrário por sua norma. Entretanto, na prática, os vetores unitários de referência uniformemente distribuídos são fundamentais para uma cobertura uniformemente distribuída do espaço objetivo. Então para gerar vetores de referência uniformemente distribuídos, um conjunto de pontos uniformemente distribuídos em um hiperplano unitário é criado utilizando o método canônico de projeto da malha simplex (do termo em inglês *Simplex-Lattice*). Este método canônico pode ser representado matematicamente pela Equação (4.25).

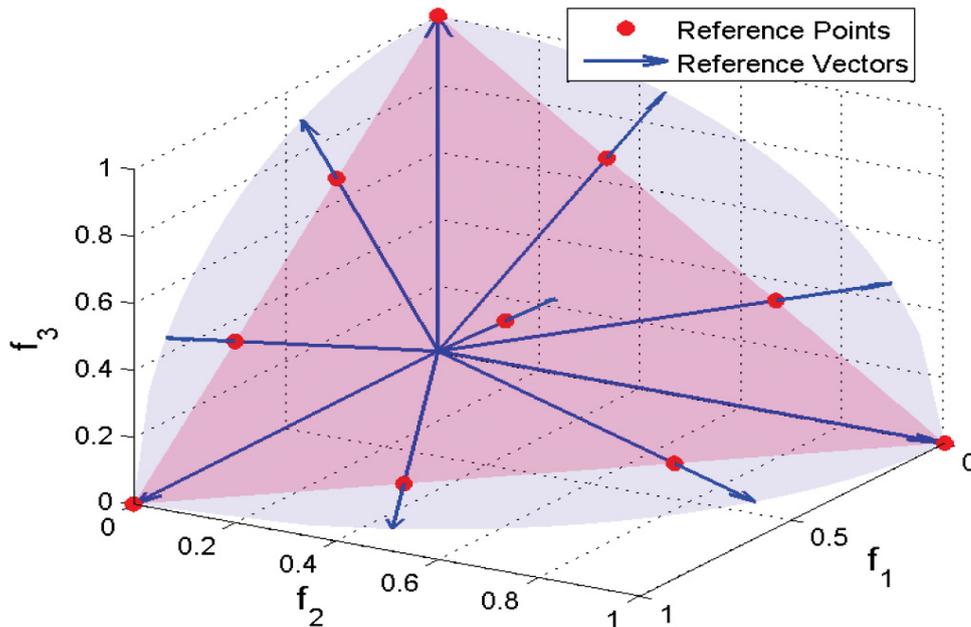
$$\begin{cases} u_i = (u_i^1, u_i^2, \dots, u_i^M) \\ u_i^j \in \left\{ \frac{0}{H}, \frac{1}{H}, \dots, \frac{H}{H} \right\}, \sum_{j=1}^M u_i^j = 1 \end{cases} \quad (4.25)$$

sendo $i = 1, 2, \dots, N$, N é o número de pontos distribuídos uniformemente, M é o número de objetivos e H é um número positivo do projeto da malha simplex, definido como na Equação (4.2). Portanto, os vetores unitários de referência (v_i) podem ser obtidos como definido na Equação (4.26)

$$v_i = \frac{u_i}{\|u_i\|} \quad (4.26)$$

Como exemplo, a Figura 31 ilustra o mapeamento dos pontos de referência de um hiperplano para uma hiperesfera. Neste caso, $H \binom{M+m-1}{m}$ com $M = 3$ e $m = 3$ gera dez pontos de referência uniformemente distribuídos em um hiperplano, depois estes pontos são mapeados em uma hiperesfera gerando os dez vetores de referência.

Figura 31 – Geração de vetores de referência uniformemente distribuídos em um espaço de três objetivos.



Fonte: Adaptada de Cheng et al. (2016).

4.2.4. Processamento Paralelo LA e ILS.

As etapas dos algoritmos LA e ILS são descritas a seguir.

4.2.4.1. Etapas do algoritmo LA

As etapas de avaliação de fertilidade, acasalamento, defesa territorial e aquisição territorial são idênticas às apresentadas nas seções 4.1.4 a 4.1.7.

4.2.4.2. Etapas do algoritmo ILS

Aqui irá inicializar os processos do algoritmo de busca local iterativa. As próximas sub-seções descreverão estes processos.

4.2.4.2.1. Memória de soluções

A memória (x_o) fica responsável por armazenar soluções como pontos de referências para orientar a busca em direções promissoras ou apenas para se comparar novas soluções que irão ser encontradas ao longo dos processos do algoritmo ILS. Na memória serão armazenados as soluções encontradas pelo algoritmo MOLA que será $x_o \leftarrow [x^{male}, x^{female}]$.

4.2.4.2.2. Perturbação

As soluções iniciais passam pelo processo de perturbação. Este processo tentará minimizar as chances de ocorrer convergência prematura e permitirá uma exploração melhor do espaço de busca das soluções. A Equação (4.27) ilustra o processo de perturbação nas soluções iniciais.

$$x'_o = x_o + step_size \quad (4.27)$$

onde x_o é a população obtida pelo MOLA e $step_size$ é o valor de perturbação igual a 0,01.

4.2.4.2.3. Busca local

O processo de busca local irá verificar se há outras soluções ao seu redor, ou seja, irá realizar uma busca local na vizinhança da solução escolhida na tentativa de melhorar as soluções. A Equação (4.28) mostra matematicamente o processo de busca local (*point*).

$$point = N(x_o, 0,1) \quad (4.28)$$

onde $N(\mu, \delta)$, é uma distribuição normal sendo μ a média (x_o é a população obtida pelo MOLA) e δ (0,1) o desvio padrão.

4.2.4.2.4. Checagem não dominância

O processo de checagem de não dominância é descrito na seção (4.1.8). Resumidamente, verifica se tal solução é dominada por qualquer outra solução e se não for dominada por nenhuma outra solução pode ser dita de conjunto de Pareto e a que for dominada é descartada.

4.2.5. Seleção Guiada por Vetor de Referência

Na seleção guiada o espaço objetivo é particionado em vários subespaços utilizando os vetores de referência, e a seleção pode ser realizada separadamente dentro de cada subespaço. A partição do espaço objetivo equivalente a adicionar uma restrição ao subproblema especificado em cada vetor de referência, sendo capaz de ajudar a equilibrar a convergência e a diversidade em abordagens baseadas em decomposição. Posto isto, a estratégia de seleção guiada por vetor de referência consiste em quatro etapas:

- a. Translação do valor objetivo: o ponto inicial dos vetores de referência utilizados serão a origem das coordenadas. Os valores das funções objetivos dos indivíduos da população x_t , definidos como $F_t = f_{t,1}, f_{t,2}, \dots, f_{t,|x_t|}$, t é o índice de geração, são transladados conforme Equação (4.29)

$$F'_{t,i} = f_{t,i} - z_t^{min} \quad (4.29)$$

$i = 1, \dots, |x_t|$, $f_{t,i}$, $F'_{t,i}$ são as funções objetivos dos indivíduos i antes e depois da translação $z_t^{min} = (z_{t,1}^{min}, z_{t,2}^{min}, \dots, z_{t,m}^{min})$ representa os valores mínimos das funções objetivos calculados a partir de F_t .

b. Partição da população: após a translação das funções objetivos, a população x é particionada em N subpopulações $x_{t,1}, x_{t,2}, \dots, x_{t,N}$ associando cada indivíduo ao seu vetor de referência mais próximo. Assim, o valor do cosseno entre um vetor objetivo e um vetor de referência pode ser calculado pela Equação (4.30):

$$\cos \theta_{t,i,j} = \frac{F'_{t,i} \cdot v_{t,j}}{\|F'_{t,i}\|} \quad (4.30)$$

sendo, $\theta_{t,i,j}$, que são os ângulos entre o vetor objetivo ($F'_{t,i}$) e o vetor de referência ($v_{t,j}$). A Equação (4.31) representa um indivíduo $I_{t,i}$, que será alocado em uma subpopulação $\bar{x}_{t,k}$ se e somente o ângulo entre $F'_{t,i}$ e $v_{t,k}$ for mínimo (ou seja, o valor do cosseno é máximo) entre todos os vetores de referência.

$$\bar{x}_{t,k} = \{I_{t,i} | k = \underset{\text{para } j \in \{1,2, \dots, N\}}{\text{argmax}} \cos \theta_{t,i,j}\} \quad (4.31)$$

$I_{t,i}$ é denotado com I – ésimo indivíduo de x_t e $i = 1, 2, \dots, |x_t|$.

c. Cálculo da distância penalizada pelo ângulo: Uma vez a população (x_t) particionada em N subpopulações $\bar{x}_{t,1}, \bar{x}_{t,2}, \dots, \bar{x}_{t,N}$, um elitista pode ser selecionado de cada subpopulação para criar x_{t+1} . Então, ao transformar o vetor de referência $F'_{t,i}$ em subpopulação j , o critério de convergência pode ser representado pela distância entre $F'_{t,i}$ para o ponto ideal $\|F'_{t,i}\|$ e o critério de diversidade pode ser representado pelo ângulo agudo entre $F'_{t,i}$ e $v_{t,j}$. Por fim, para equilibrar os dois critérios (convergência e diversidade) a distância penalizada pelo ângulo pode ser dada pelas Equações (4.32 – 4.34):

$$d_{t,i,j} = (1 + x(\theta_{t,i,j})) \cdot \|F'_{t,i}\| \quad (4.32)$$

Para $x(\theta_{t,i,j})$ a função de penalidade relacionada ao $\theta_{t,i,j}$, terá:

$$x(\theta_{t,i,j}) = M \cdot \left(\frac{t}{t_{max}}\right) \alpha \cdot \left(\frac{\theta_{t,i,j}}{\Upsilon v_{t,i,j}}\right) \quad (4.33)$$

$$\Upsilon v_{t,i,j} = \min_{i \in \{1,2, \dots, N\}, i \neq j} \langle v_{t,i}, v_{t,j} \rangle \quad (4.34)$$

onde M é o número de objetivos, N é o número do vetor de referência, t_{max} é o valor máximo de gerações pré-definido, $\Upsilon v_{t,i,j}$ é o menor valor do ângulo entre o vetor de referência $v_{t,i}$ e o outros vetores de referência da geração corrente e α é o parâmetros de controle de probabilidade de seleção $x(\theta_{t,i,j})$ do cálculo de penalização pelo ângulo.

Por fim, um das vantagens sobre o cálculo de distância dos ângulos é que não precisa saber a distância exata que a solução candidata encontra-se do ponto ideal, pois o ângulo entre a solução candidata e o vetor de referência é constante. Após realizar o cálculo de penalização, a população passa pelo processo de seleção elitista, onde serão selecionados os pais para poder gerar os novos descendentes.

4.2.6. Adaptação do vetor de referência

Dado um conjunto de vetores unitários de referência uniformemente distribuídos, espera-se obter um conjunto de soluções Pareto ótimas uniformemente distribuídas que são os pontos de intersecção entre cada vetor de referência e a fronteira de Pareto. No entanto, isso acontece apenas se os valores das funções de todos os objetivos puderem ser facilmente normalizados no mesmo intervalo, por exemplo, $[0, 1]$. Infelizmente, na prática, podem existir MaOPs onde diferentes objetivos são normalizados para diferentes intervalos.

Os autores Cheng et al. (2016), propuseram uma alternativa de se realizar a normalização dos objetivos à medida que a busca pelo conjunto de Pareto avança-se. Portanto, ao invés de normalizar as funções objetivo, adaptaram os vetores de referência de acordo com as faixas das funções objetivo, conforme a Equação (4.35).

$$V_{t+1,i} = \frac{v_{0,i} \circ (z_{t+1}^{max} - z_{t+1}^{min})}{\|v_{0,1} \circ (z_{t+1}^{max} - z_{t+1}^{min})\|} \quad (4.35)$$

onde $i = 1, \dots, N$, $V_{t+1,i}$ é o i -ésimo vetor de referência adaptado para a próxima geração $t+1$, $v_{0,i}$ é o i -ésimo vetor de referência uniformemente distribuído, que é gerado no estágio de inicialização do algoritmo, e z_{t+1}^{max} e z_{t+1}^{min} são os valores máximo e mínimo de cada função objetivo na geração $t+1$, respectivamente. O operador \circ é o produto de Hadamard que multiplica elemento a elemento de dois vetores (ou matrizes) de mesmo tamanho.

5. RESULTADOS OBTIDOS

Este capítulo apresenta a comparação do desempenho das metaheurísticas propostas, inicialmente utilizando um conjunto de funções de *Benchmark* e métricas de qualidade elas tem seus desempenhos comparadas a outras metaheurísticas multiobjetivos amplamente utilizados na literatura: MOPSO, MOGWO, NSGA-III, MOEA/D e MSSA. Em seguida as metaheurísticas propostas são utilizadas em problemas mais realísticos, serão tratados problemas de engenharia. Tais estudos de caso são constituídos por equações algébricas, algébrico-diferenciais ou integro-diferenciais e descrevem problemas de grande interesse no meio acadêmico e industrial.

5.1. Funções de *Benchmark*

Existem várias funções de *benchmark* adequadas para testar algoritmos de otimização metaheurística. Neste trabalho, foram utilizados 15 problemas de teste para otimização multiobjetivo conforme originalmente apresentado em Cheng et al (2018) e usado na Competição CEC'2018, que são mostrados na Tabela 2.

Essas funções têm a possibilidade de serem utilizadas com vários objetivos, assim pode-se testar tantos algoritmos multiobjectivos (número de objetivos até 3) como de muitos objetivos (número de objetivos maior que 3). Neste trabalho serão utilizados problemas com 3 e 5 objetivos. Quanto ao número de variáveis (dimensão do problema) utilizadas são as propostas na competição.

Nesse conjunto de funções as fronteiras de Pareto Ótimas são disponibilizadas no trabalho de Cheng et al (2018) e na Competição CEC'2018. Os códigos que representam essas fronteiras e utilizados neste trabalho estão disponíveis na plataforma PlatEMO ([PlatEMO/PlatEMO/Problems at master · BIMK/PlatEMO · GitHub](https://github.com/BIMK/PlatEMO)).

A Tabela 6 complementa a Tabela 1, apresentada no capítulo 3, com as seguintes informações. A primeira coluna é o nome da função, a segunda contém os números de funções objetivo, a terceira define o número de variáveis otimizadas, a quarta mostra as propriedades e a última coluna contém o respectivo espaço de busca de cada função.

Tabela 6 – Problemas testes da competição CEC'2018.

Função	Objetivos	Dimensão (D)	Propriedades	Espaço de busca
MaF1	3, 5	12, 14	L	$[0, 1]^D$
MaF2	3, 5	12, 14	C	$[0, 1]^D$
MaF3	3, 5	12, 14	Co, M	$[0, 1]^D$
MaF4	3, 5	12, 14	C, M	$[0, 1]^D$
MaF5	3, 5	12, 14	C, B	$[0, 1]^D$
MaF6	3, 5	12, 14	C, De	$[0, 1]^D$
MaF7	3, 5	12, 14	Mx, D, M	$[0, 1]^D$
MaF8	3, 5	12, 14	L, De	$[-10\ 000, 10\ 000]^2$
MaF9	3, 5	12, 14	L, De	$[-10\ 000, 10\ 000]^2$
MaF10	3, 5	12, 14	Mx, B	$\prod_{i=1}^D [0, 2i]$
MaF11	3, 5	12, 14	Co, D, N	$\prod_{i=1}^D [0, 2i]$
MaF12	3, 5	12, 14	C, N, BD	$\prod_{i=1}^D [0, 2i]$
MaF13	3, 5	12, 14	C, U, N, De	$[0, 1]^2 \times [-2, 2]^{D-2}$
MaF14	3, 5	20, 100	L, Ps, Ls	$[0, 10]^D$
MaF15	3, 5	20, 100, 200	Co, Ps, Ls	$[0, 10]^D$

5.1.1. Parâmetros de configuração

Os algoritmos utilizados nesta comparação são: MOLA, MOLAILS, MOPSO, MOGWO, NSGA-III, MOEA/D e MSSA. Cada algoritmo tem o tamanho da população igual a 400, foram submetidos a 50 execuções, com 200 iterações/gerações em cada execução. A configuração utilizada por cada algoritmo é mostrada na Tabela 7.

Tabela 7 – Configurações do algoritmo de otimização.

Configuração	MOLA	MOLAILS	MOPSO	MOGWO	MOEA/D	MSSA	NSGAIII
Coefficiente de Aprendizagem Global			2,0				
Coefficiente de Aprendizagem Pessoal			1,0				
Idade inicial	0,0	0,0					
Idade de maturidade	3,0	3,0					
Frequência para chamar vetor de referência		0,1					
Número de			7	10			

5.1.2. Avaliação de desempenho

Os algoritmos multiobjetivos baseados em indicadores de desempenho tornaram-se populares nos últimos anos, uma vez que o uso de medidas de indicadores fornece uma quantificação direta para guiar o processo de busca. A frente de Pareto aproximada pode ser avaliada de acordo com o indicador.

Neste trabalho serão utilizados os indicadores de *Distância geracional invertida (IGD)* e o Hipervolume (HV), pois consideram tanto diversidade quanto convergência.

5.1.2.1. Distância geracional invertida (IGD)

Quanto menor o valor de IGD, melhor a qualidade das soluções de Pareto obtidas com relação ao conjunto verdadeiro da Frente de Pareto ótimo. Na Tabela 8, encontram-se os valores do critério IGD para os algoritmos propostos e outros algoritmos presentes na literatura. Posto isto, as melhores médias apresentadas para as MaFs com 3 objetivos foram destacadas na cor vermelha e a cor azul representa as melhores médias para as MaFs com 5 objetivos.

Observa-se que o algoritmo MOLA e MOLAILS tiveram desempenho semelhante aos outros algoritmos clássicos da literatura (NSGA III, MOPSO, MOEA-D, MOGWO e MSSA). Porém, o algoritmo MOLA apresentou uma métrica de IGD melhor do que os outros algoritmos nas funções de teste com 3 e 5 objetivos. Os resultados em termos de ranking podem ser vistos na Tabela 9, sendo vermelha indicando o algoritmo em primeiro lugar do rank com 3 objetivos e a cor azul ilustra o algoritmo em primeiro lugar do rank com 5 objetivos.

Tabela 8 – Valores da métrica distância geracional invertida (IGD) obtidos pelos sete algoritmos no teste MaF.

Função	M	MOLA		MOLAILS		MOPSO		MOGWO		MOEA/D		MSSA		NSGA III	
		Med.	Std	Med.	Std	Med.	Std	Med.	Std	Med.	Std	Med.	Std	Med.	Std
MaF1	3	7,60e-4	0,0054	0,0474	0,0064	0,1230	0,0167	0,1993	0,0140	0,0080	1,08e-4	0,0734	0,0042	0,025	4,83e-3
	5	0,0030	0,0209	0,0030	0,0115	0,3261	0,0435	0,1072	1,03e-5	0,0070	1,08e-4	0,1976	0,0092	0,1262	0,0060
MaF2	3	0,0044	0,0311	0,0203	7,09e-4	0,1385	0,0077	0,0042	5,4e-4	0,0040	3,42e-4	0,1955	0,0103	0,734	7,31e-4
	5	0,0099	0,0701	0,0075	0,0013	0,5401	0,0163	0,0076	3,4e-5	0,0048	2,53e-4	0,6362	0,0702	1,2924	0,0078
MaF3	3	0,0119	0,0267	1,41e+3	1,68e+3	5e+5	3,2e+4	0,0047	3,3e+5	0,0039	1,00e-4	2,3e+4	1,03e+4	0,024	5,53e-5
	5	0,0185	0,1306	205,36	0,3089	3e+5	4e+4	0,0102	1,1e-4	0,0045	1e-4	3,7e+4	6,7e+4	0,0609	3,5e-4
MaF4	3	0,0011	0,0081	0,1047	0,0058	846,15	134,17	0,016	1,03e-3	230,22	76,78	305,65	195,68	0,1657	0,0042
	5	0,0443	0,3129	20,17	1,1725	3,9e+3	3e+3	0,010	3,12e4	151,31	78,05	2,26e+3	398,32	2,1891	0,4382
MaF5	3	0,0042	0,0298	4,2719	1,2735	0,8917	0,3813	1,9e+3	3,4e+3	2,1997	1,0664	0,4857	0,1220	3,6441	1,6281
	5	0,0362	0,2561	22,73	1,78e-6	7,5279	1,0811	5,6e+4	7,2e+3	0,005	1,20e-4	5,4414	1,2073	3,1623	0,3197
MaF6	3	1,77e-4	0,0013	0,9637	0,1735	0,0199	0,0035	0,005	1,97e-6	0,005	5,50e-5	0,0288	0,0072	0,004	4,56e-4
	5	1,97e-4	0,0014	1,8896	0,5998	2,1717	7,2176	0,008	1,07e-6	0,0062	4,29e-5	0,2435	0,3395	0,0083	0,0023
MaF7	3	9,49e-4	0,0067	0,1049	0,1089	0,1411	0,0240	0,0030	0,00021	0,0063	2,04e-4	0,0132	0,0417	0,0329	6,34e-4
	5	0,0054	0,0381	0,0063	0,0299	0,6526	0,0920	0,0043	1,24e-4	0,0043	2,57e-4	0,4453	0,4094	0,2399	0,0049
MaF8	3	0,0076	0,0540	0,0062	0,0016	0,0768	0,0081	0,0071	0,0037	0,2861	0,0685	0,0678	0,0040	0,0443	0,0012
	5	0,0099	0,0701	0,2731	0,0345	0,1267	0,0103	0,0120	0,0055	0,2991	1,1217	0,1153	0,0059	0,1230	0,0151
MaF9	3	0,0067	0,0475	0,0528	0,0461	4,1047	0,0625	0,0910	0,1083	0,0725	0,3206	0,0867	0,0746	0,5101	0,0165
	5	0,0138	0,0976	0,0071	0,0345	9,8698	9,5268	0,1763	0,3438	0,0075	0,0109	0,7719	0,1636	0,4742	0,1985
MaF10	3	0,0151	0,1069	0,3062	0,0227	2,2068	0,0909	0,9e-4	2,3e-4	2,4051	0,0963	1,5863	0,0632	0,3653	0,2018
	5	0,0300	0,2120	0,0247	0,0163	2,5531	0,0748	0,0101	0,0012	2,8189	0,0818	1,6773	0,0814	1,9909	0,2024
MaF11	3	0,0046	0,0327	0,1063	0,0399	0,3860	0,0385	0,017	5,32e-4	1,2171	0,1245	0,3930	0,0613	0,1406	0,0036
	5	0,0152	0,1078	0,5225	0,0134	0,9247	0,0577	0,0574	0,0027	1,2033	0,1194	3,0240	1,1672	2,8732	0,006
MaF12	3	0,0035	0,0250	0,0155	0,0428	0,4826	0,0611	0,0193	0,0210	1,7563	0,6163	0,5997	0,1601	1,219	0,0105
	5	0,0197	0,1395	0,8675	0,0522	1,8243	0,1660	0,0344	0,0205	0,0212	0,0122	3,4885	0,5467	0,4007	0,007
MaF13	3	0,0027	0,0194	0,6935	0,2136	0,2441	0,0498	0,0027	0,0021	1,991	0,1327	0,1462	0,0342	0,1795	0,0094
	5	0,0044	0,0311	1,0722	0,1049	0,5531	0,1470	1,0904	7,1041	0,0040	2,25e-4	1,0096	2,2582	0,3228	0,023
MaF14	3	0,0241	0,1704	0,0317	0,2280	5,9826	4,2509	0,0035	1,87e-4	0,0098	0,0072	1,3e+3	1,45e+3	0,9004	0,014
	5	0,0529	0,3741	1,0133	0,0256	18,990	6,7586	0,0090	0,0118	0,0132	0,0066	1,04e+3	955,1	3,7871	0,057
MaF15	3	0,0107	0,0758	0,0202	0,1544	5,9826	4,2509	7,55e-4	0,0016	0,0094	0,0109	3,0335	1,5684	0,6012	0,031
	5	0,0443	0,3135	0,0673	0,0992	31,850	3,4810	0,0060	0,0030	0,0471	0,0080	47,370	17,16	0,7347	0,021

Tabela 9 – Ordem crescente em relação ao IGD de melhor desempenho obtido entre os algoritmos.

Função	3 objetivos										5 objetivos				
	MOLA	MOLAAILS	MOPSO	MGWO	MOEAD	MSSA	NSGAI/III	MOLA	MOLAAILS	MOPSO	MGWO	MOEAD	MSSA	NSGAI/III	
MaF1	1	4	6	7	3	5	2	1,5	1,5	7	4	3	6	5	
MaF2	3	4	5	2	1	6	7	4	2	5	3	1	6	7	
MaF3	4	6	5	3	1	7	2	3	5	6	2	1	7	4	
MaF4	1	3	7	2	5	6	4	2	4	7	1	5	6	3	
MaF5	1	6	3	7	4	2	5	2	6	5	7	1	4	3	
MaF6	1	7	5	3	3	6	2	1	6	7	3	2	5	4	
MaF7	1	6	7	2	3	4	5	3	4	7	1,5	1,5	6	5	
MaF8	3	1	6	2	7	5	4	1	6	5	2	7	3	4	
MaF9	1	2	7	5	3	4	6	3	1	7	4	2	6	5	
MaF10	2	3	6	1	7	5	4	3	2	6	1	7	4	5	
MaF11	1	3	5	2	7	6	4	1	3	5	2	6	7	4	
MaF12	1	2	4	3	7	5	6	1	5	6	3	2	7	4	
MaF13	1,5	6	5	1,5	7	3	4	2	6	4	7	1	5	3	
MaF14	3	4	6	1	2	7	5	3	4	6	1	2	7	5	
MaF15	3	4	7	1	2	6	5	2	4	6	1	3	7	5	
Rank	27,5	54	80	42,5	60	77	65	32,5	59,5	89	41,5	44,5	86	66	

Fazendo uma análise dos resultados da métrica IGD para as funções de *benchmark* com três objetivos, a metaheurística MOLA se comportou de forma melhor que os outros algoritmos nas funções (MaF1, MaF4, MaF5, MaF6, MaF7, MaF9, MaF11 e MaF12) e de forma semelhante na função (MaF13) empatando com o MOGWO. No ranking geral o MOLA obteve a primeira colocação se destacando entre os demais algoritmos, pois ele está sempre os três primeiros colocados, exceção somente na MaF4 que ficou em quarto lugar.

Classificado em segundo lugar está a metaheurística MOGWO que se comportou de forma melhor que os outros algoritmos nas funções (MaF10, MaF14 e MaF15) e de forma semelhante na função (MaF13) empatando com o MOLA. O MOGWO possui várias colocações em segundo e terceiro lugar, porém tem colocações em último lugar.

Em terceiro lugar no rank está a metaheurística cooperativa MOLAILS comportando-se de forma melhor que os outros algoritmos na função (MaF8). O MOLAILS, porém, tem colocações intermediárias nas outras funções o que leva ele a esta colocação.

Embora a metaheurística MOEA/D tenha mais vitórias que o MOLAILS, ela se classificou em quarto lugar, pois ela possui várias colocações em último lugar. Já as metaheurísticas NSGA III, MSSA e MOPSO não obtiveram nenhuma vitória, ficando o algoritmo NSGA III em quinto lugar, o algoritmo MSSA em sexto lugar e o MOPSO em sétimo lugar do rank.

Agora, fazendo uma análise dos resultados para as funções de *benchmark* com cinco objetivos, a metaheurística MOLA se comportou de forma melhor que os outros algoritmos nas funções (MaF6, MaF8, MaF11 e MaF12) e de forma semelhante na função (MaF1) empatando com o MOLAILS. No ranking geral o MOLA obteve a primeira colocação se destacando entre os demais algoritmos, pois ele está sempre os três primeiros colocados, exceção somente na MaF2 que ficou em quarto lugar.

Classificado em segundo lugar está a metaheurística MOGWO que se comportou de forma melhor que os outros algoritmos nas funções (MaF4, MaF10, MaF14 e MaF15) e de forma semelhante na função (MaF7) empatando com o MOEA/D. O MOGWO nas outras funções possui colocações em segundo e terceiro lugar, porém tem uma colocação em último lugar.

Em terceiro lugar no rank está a metaheurística MOEA/D que se comportou de forma melhor que os outros algoritmos nas funções (MaF2, MaF3, MaF5 e

MaF13) e de forma semelhante na função (MaF7) empatando com o MOGWO. O MOEA/D possui colocações em segundo e terceiro lugar, porém tem colocações em último lugar, o que o leva a está colocação.

A metaheurística cooperativa MOLAILS se classificou em quarto lugar tendo se comportado de forma melhor que os outros algoritmos na função (MaF9) e de forma semelhante na função (MaF1) empatando com o MOLA. Já as metaheurísticas NSGA III, MSSA e MOPSO não obtiveram nenhuma vitória, ficando o algoritmo NSGA III em quinto lugar, o algoritmo MSSA em sexto lugar e o MOPSO em sétimo lugar do rank.

Para uma análise da semelhança entre as metaheurísticas, foram utilizados os testes estatísticos Kruskal – Wallis (p_{valor}) e comparações múltiplas (z_{teste}) e a Tabela 10 apresenta os resultados da distância geracional invertida. Os algoritmos com índice menor que 0,03 (ou 3%) apresentam discrepâncias e as metaheurísticas com índice superior a 0,03 apresentam amostras semelhantes ou não há diferença entre as metaheurísticas. Para maior compreensão, os valores obtidos com referência no p_{valor} estão destacados em negrito e podem ser vistos na Tabela 10.

Tabela 10 – Teste estatístico da hipótese nula usando os resultados do IGD.

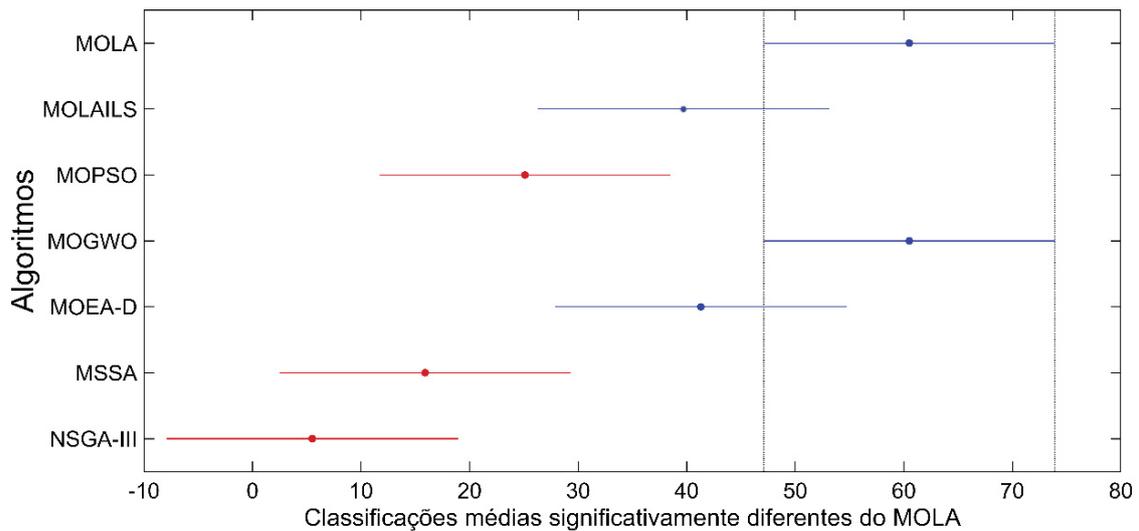
p_{valor}	3 objetivos					5 objetivos						
	MOLAIS	MOPSO	MOGWO	MOEA-D	MSSA	NSGA - III	MOLAIS	MOPSO	MOGWO	MOEA-D	MSSA	NSGA - III
MaF1	0,4606	0,0093	0,9282	0,0004	0,0979	0,9098	0,0979	0,0093	0	0,0004	0,4606	0,9098
MaF2	0,9098	0,4606	0	0,0093	0,0979	0,0004	0,9092	0,4588	0	0,0126	0,0760	0,0004
MaF3	0,4005	0,0004	0,1058	0,0115	0,9098	0,2831	0,4007	0,0004	0	0,0979	0,0093	0,1030
MaF4	0,2812	0,0011	0	0,0075	0,0904	0,0904	0,4606	0,0004	0	0,0316	0,0347	0,9098
MaF5	0,6213	0,0019	0	0,3467	0	0	1	0,1059	0	0,9999	0,0047	0,2201
MaF6	0,0093	0,8987	0	0,0979	0,4812	0,9098	0,0280	0,6679	0	0	0,0643	0,0004
MaF7	0,0012	0,0012	0	0	0	0,2490	0,0013	0,0001	0,8334	0	0,1036	0,9918
MaF8	0,0064	0	0,9282	0	0,0002	0,2950	0	0,0026	0,9282	0	0,0532	0,0266
MaF9	0,1822	0,0040	0	0,0006	0,9517	1	0,0126	0	0	0,0001	0,0004	0,7174
MaF10	0,9457	0,3690	0	0,3467	0,0457	0,3690	0,0301	0,9457	0	0,0007	0,3690	0,0310
MaF11	0	0,2506	0	0,1082	0,0226	0,2963	0,9896	0,0439	0	0,0624	0	0
MaF12	0,0502	0,0017	0	0	0	0	1	0,1000	0,1000	0	0	0
MaF13	0	0,0115	0	0	0,2945	0,0004	0,0025	0,0629	0,0011	0	0,9969	0
MaF14	0	0,2946	0,0004	0,0001	0,2945	0	0,0004	0,1632	0,0042	0,0035	0	0
MaF15	0	0,0119	0	0,0173	0,0222	0	0,0007	0	0,1000	0,0941	0	0

Ao analisar a similaridade dos resultados para as funções de *benchmark* com três objetivos utilizando a métrica IGD (Tabela 6), o MOLAILS apresenta similaridade com MOLA nas funções (MaF6, MaF7, MaF8, MaF11, MaF13, MaF14 e MaF15). O MOPSO apresenta similaridade com MOLA nas funções (MaF1, MaF3, MaF4, MaF5, MaF7, MaF8, MaF9, MaF12, MaF13 e MaF15). O MOGWO apresenta similaridade com MOLA nas funções (MaF2, MaF4, MaF5, MaF6, MaF7, MaF9, MaF10, MaF11, MaF12, MaF13, MaF14 e MaF15). O MOEA/D apresenta similaridade com MOLA nas funções (MaF1, MaF2, MaF3, MaF4, MaF7, MaF8, MaF9, MaF12, MaF13, MaF14 e MaF15). O MSSA apresenta similaridade com MOLA nas funções (MaF5, MaF7, MaF8, MaF11, MaF12 e MaF15). O NSGA III apresenta similaridade com MOLA nas funções (MaF2, MaF5, MaF12, MaF13, MaF14 e MaF15). Portanto o MOGWO é o algoritmo que apresenta a maior similaridade com o MOLA e os algoritmos MSSA e NSGA III representam a menor similaridade com o MOLA.

Agora ao analisar a similaridade dos resultados para as funções de *benchmark* com cinco objetivos utilizando a métrica IGD (Tabela 6), o MOLAILS apresenta similaridade com MOLA nas funções (MaF6, MaF7, MaF8, MaF9, MaF13, MaF14 e MaF15). O MOPSO apresenta similaridade com MOLA nas funções (MaF1, MaF3, MaF4, MaF7, MaF8, MaF9 e MaF15). O não MOGWO apresenta similaridade com MOLA nas funções (MaF1, MaF2, MaF3, MaF4, MaF5, MaF6, MaF9, MaF10, MaF11, MaF13 e MaF14). O MOEA/D apresenta similaridade com MOLA nas funções (MaF1, MaF2, MaF6, MaF8, MaF9, MaF10, MaF12, MaF13 e MaF14). O MSSA apresenta similaridade com MOLA nas funções (MaF3, MaF5, MaF9, MaF11, MaF12, MaF14 e MaF15). O NSGA III apresenta similaridade com MOLA nas funções (MaF2, MaF6, MaF8, MaF11, MaF12, MaF13, MaF14 e MaF15). Portanto o MOGWO é o algoritmo que apresenta a maior similaridade com o MOLA enquanto os algoritmos MOLAILS, MOPSO e MSSA apresentam a menor similaridade.

A Figura 32 mostra um exemplo das classificações médias obtidas a partir dos dados do IGD com 5 objetivos (Tabelas 8 e 10), onde ao menos três grupos (MOPSO, MSSA e NSGA III) têm classificações médias significativamente diferentes ao algoritmo MOLA. Os algoritmos MOLAILS, MOGWO e MOEA/D apresentaram nenhuma/pouca similaridade em relação ao MOLA.

Figura 32 – Classificações médias diferentes do MOLA.



Fonte: Autora, 2024.

5.1.2.2. Métrica Hipervolume (HV)

A métrica HV mede o tamanho do espaço objetivo dominado por soluções no conjunto de referência e limitado por r^* . O Hipervolume pode medir a aproximação em termos de diversidade e convergência. Quanto maior o valor de HV, melhor é a qualidade das soluções de Pareto obtidas no conjunto de referência para aproximar da verdadeira solução ótima de Pareto. A Tabela 11 mostra os valores do critério HV obtidos pelo MOLA e outros algoritmos, destacando as melhores médias obtidas para 3 objetivos (cor vermelha) e para 5 objetivos (cor azul).

Observa-se que o algoritmo MOLA e MOLAILS tiveram desempenho semelhante aos outros algoritmos clássicos da literatura (NSGA III, MOPSO, MOEA-D, MOGWO e MSSA). Porém, o algoritmo MOLAILS apresentou uma métrica de Hipervolume melhor do que os outros algoritmos nas funções de teste com 3 e 5 objetivos. Os resultados em termos de ranking podem ser vistos na Tabela 12, sendo o melhor rank para 3 objetivos na cor vermelha e para 5 objetivos na cor azul.

Tabela 11 – Valores da métrica de Hipervolume (HV) obtidos pelos sete algoritmos no teste MaF.

Função	M	MOLA		MOLAIS		MOPSO		MOGWO		MOEA/D		MSSA		NSGA III	
		Média	Std	Média	Std	Média	Std	Média	Std	Média	Std	Média	Std	Média	Std
MaF1	3	0,3117	6,06e-4	0,2694	0,0104	0,1749	0,0173	0,2057	0,1275	0,0464	0,4198	0,2339	0,0051	0,3119	5,49e-4
	5	0,4409	4,02e-4	0,5088	0,0036	9,80e-4	8,6e-4	0,2096	0,2002	9,09e-4	5,9e-4	0,0082	7,2e-4	0,0147	8,59e-4
MaF2	3	0,3707	6,5e-4	0,2196	7,86e-4	0,3315	0,0044	0,3296	0,0041	0,1964	0,0281	0,3277	0,0060	0,2174	5,17e-4
	5	0,0102	1,21e-4	0,0488	3,15e-4	2,41e-4	1,4e-4	0,7824	0,0239	1,66e-4	8,17e-4	6,66e-5	1,4e-4	0,0454	7,25e-4
MaF3	3	1,2236	0,1373	1,8139	0,0887	0,8180	0,0761	0,9605	0,0664	0,5446	0,0481	0,0776	0,0070	1,2881	2,14e-4
	5	3,0702	0,3943	4,0048	0,1020	0,8627	0,5031	0,8650	0,1320	0,7897	0,7723	0,1170	0,0054	0,609	3,50e-4
MaF4	3	40,36	10,97	48,29	0,0593	11,424	0,4341	10,031	0,6771	13,437	0,2824	26,30	0,5597	47,86	0,500
	5	38,118	7,55	42,70	0,0781	11,899	0,4970	20,891	0,3998	11,850	0,3071	18,218	0,3764	3,22e+3	1,1e+3
MaF5	3	18,64	12,54	58,55	149,20	30,226	5,6332	0,9615	0,0476	5,1217	2,4823	36,68	3,4805	3,6441	1,6281
	5	3,01e+4	1,46e+3	4,79e+3	1,50e+3	3,36e+3	2,7e+3	2,25e+4	1,9e+3	3,71e+3	1,7e+3	7,8e+3	5,35e+3	3,69e+4	1,57e+3
MaF6	3	0,0172	3,22e-5	0,1621	0,0404	0,1214	0,0028	0,2087	0,3920	0,0422	0,0214	0,1207	0,0083	0,1333	1,97e-4
	5	0,00912	6,18e-5	0,0172	0,0010	0,0069	0,0027	0,0153	0,0986	0,0070	0,0012	0,0048	0,0039	0,0090	7,46e-5
MaF7	3	0,1333	1,97e-4	1,6020	0,0878	1,3845	0,0426	1,4322	0,0559	1,7572	0,0218	0,1369	0,4154	1,6900	0,0018
	5	2,0652	0,0333	2,4030	0,0639	0,9891	0,2475	0,1334	3,23e ⁻⁴	0,6256	0,2793	0,4669	0,4754	2,3918	0,0170
MaF8	3	1,6321	0,0521	1,6867	0,0051	1,8647	0,0236	1,7781	0,6913	1,8557	0,6073	1,8864	0,0140	1,9812	0,0034
	5	3,2956	0,5490	3,1361	0,1011	4,4770	0,0959	0,2839	0,2420	3,2880	0,4372	4,6656	0,0371	0,1230	0,0151
MaF9	3	3,7175	3,16e-4	3,5797	0,1390	0,9207	1,3409	0,3523	0,1958	3,5638	0,5055	3,5334	0,2833	0,1230	0,0151
	5	6,2752	0,8969	6,8601	0,4404	0,6654	1,7627	0,4466	0,0371	7,6309	0,4169	10,473	2,9739	5,7212	1,6942
MaF10	3	1,8334	0,4700	52,168	0,1260	1,2383	0,8004	0,2421	0,2411	11,6050	1,0320	14,257	1,6240	1,06e+3	1,5822
	5	877,25	235,17	4,89e+3	1,8110	935,86	130,86	911,17	99,080	850,438	483,27	756,19	96,917	1,15e+3	1,447
MaF11	3	59,57	0,0893	59,55	0,0631	50,177	1,381	32,261	1,2926	74,6263	0,0334	50,553	0,9665	1,09e+3	1,5025
	5	6,05e+3	35,86	6,01e+3	1,0442	4,35e+3	183,6	4,35e+3	183,6	5,59e+3	95,442	3,11e+3	1,1e+3	1,24e+3	1,352
MaF12	3	34,495	0,2507	38,75	0,4472	1,02e+3	156,6	1,0e+3	240,08	2,15e+3	206,8	23,879	2,0872	3,18e+3	59,377
	5	4,22e+3	248,65	4,33e+3	49,28	2,10e+3	267,78	3,1e+3	1,02e+3	2,05e+3	1,0e+3	1,10e+3	469,21	23,606	0,2383
MaF13	3	0,6815	0,0204	3,9682	0,9437	0,3605	0,0576	0,3155	0,3286	0,4292	0,2721	0,5433	0,0574	0,5217	0,0132
	5	0,2960	0,0417	2,5170	0,1736	0,0528	0,0432	0,7132	0,1087	0,8001	0,0157	0,2286	0,1189	0,2148	0,0303
MaF14	3	0,6234	0,0104	0,7182	0,0202	0,3101	0,0590	0,8114	0,0713	0,6903	0,0409	0,0122	0,0176	0,4523	0,0111
	5	0,4709	0,0023	0,5030	0,0088	0,4798	0,1317	0,5443	0,0090	0,7010	0,0336	0,2352	0,0209	0,4899	0,0019
MaF15	3	0,6081	0,0029	0,7837	0,0079	0,0085	0,0296	0,0889	0,0641	0,8003	0,0122	0,0122	0,0176	0,7944	0,0026
	5	0,8623	0,1002	0,8809	0,1037	0,1055	0,0288	0,8891	0,1030	0,7716	0,1331	0,1304	0,0198	0,7161	0,1022

Tabela 12 – Ordem crescente em relação ao melhor desempenho HV obtido entre os algoritmos.

Função	3 objetivos					5 objetivos								
	MOLA	MOLAIS	MOPSO	MOGWO	MOEAD	MSSA	NSGAIII	MOLA	MOLAIS	MOPSO	MOGWO	MOEAD	MSSA	NSGAIII
MaF1	3	4	7	6	1	5	2	2	1	7	3	6	5	4
MaF2	1	5	2	3	7	4	6	4	2	5	1	6	7	3
MaF3	3	1	5	4	6	7	2	2	1	4	3	5	7	6
MaF4	3	1	6	7	5	4	2	3	2	6	4	7	5	1
MaF5	4	1	3	7	5	2	6	3	2	6	7	5	4	1
MaF6	6	2	7	1	5	4	3	3	1	6	2	5	7	4
MaF7	7	3	5	4	1	6	2	3	1	4	7	5	6	2
MaF8	7	6	3	5	4	2	1	3	5	2	6	4	1	7
MaF9	1	2	5	6	3	4	7	4	3	6	7	2	1	5
MaF10	5	2	6	7	4	3	1	5	1	3	4	6	7	2
MaF11	3	4	6	7	2	5	1	1	2	4,5	4,5	3	6	7
MaF12	6	5	3	4	2	7	1	2	1	4	3	5	6	7
MaF13	2	1	6	7	5	3	4	4	1	7	3	2	5	6
MaF14	4	2	6	1	3	7	5	7	3	6	2	1	4	5
MaF15	4	3	6	7	1	5	2	3	2	6	1	4	5	7
Rank	62	42	76	76	54	68	45	45	45	76,5	57,5	66	71	63

Fazendo uma análise dos resultados para a métrica Hipervolume nas funções de *benchmark* com três objetivos, a metaheurística MOLAILS se comportou de forma melhor que os outros algoritmos nas funções (MaF3, MaF4, MaF5 e MaF13). No ranking geral o MOLAILS obteve a primeira colocação se destacando entre os demais algoritmos, pois ele nunca foi o último colocado nas 15 funções.

Classificado em segundo lugar está a metaheurística NSGA III que se comportou de forma melhor que os outros algoritmos nas funções (MaF8, MaF10, MaF11 e MaF12). O NSGA possui várias colocações intermediárias, porém tem colocação em último lugar.

Em terceiro lugar no rank está a metaheurística MOEA/D comportando-se de forma melhor que os outros algoritmos na função (MaF1, MaF7 e MaF15). O MOEA/D, porém, tem colocações intermediárias nas outras funções e também colocação em último lugar, o que leva ele a esta colocação.

A metaheurística MOLA se classificou em quarto lugar tendo se comportado de forma melhor que os outros algoritmos na função (MaF2 e MaF9). Já a metaheurística MSSA ficou em quinto lugar, porém sem nenhuma vitória, porém com colocações intermediárias nas diversas funções. Empatados em último lugar no rank o algoritmo MOGWO que se comportou de forma melhor que os outros algoritmos nas funções (MaF6 e MaF14), porém tem várias colocações em último lugar, e o MOPSO que não obteve nenhuma vitória, porém com colocações intermediárias que compensou as vitórias do MOGWO.

Agora, fazendo uma análise dos resultados para as funções de *benchmark* com cinco objetivos, a metaheurística MOLAILS se comportou de forma melhor que os outros algoritmos nas funções (MaF1, MaF3, MaF6, MaF7, MaF10, MaF12 e MaF13). No ranking geral o MOLAILS obteve a primeira colocação se destacando entre os demais algoritmos, pois ele está sempre os três primeiros colocados, exceção somente na MaF8 que ficou em quinto lugar.

Classificado em segundo lugar está a metaheurística MOLA que se comportou de forma melhor que os outros algoritmos na função (MaF11). O MOLA nas outras funções possui colocações intermediárias, porém tem uma colocação em último lugar.

Em terceiro lugar no rank está a metaheurística MOGWO que se comportou de forma melhor que os outros algoritmos nas funções (MaF2, e MaF15). O

MOGWO possui colocações em segundo e terceiro lugar, porém tem colocações em último lugar, o que o leva a esta colocação.

A metaheurística NSGA III se classificou em quarto lugar tendo se comportar de forma melhor que os outros algoritmos na função (MaF4 e MaF5), porém possui colocações piores nas demais funções sendo que está na última colocação em várias funções. Em quinto lugar a metaheurística MSSA se comportou de forma melhor que os outros algoritmos nas funções (MaF8, e MaF9). Em sexto lugar ficou a metaheurística MOEA/D que se comportou de forma melhor que os outros algoritmos na função (MaF14). Em último lugar ficou a metaheurística MOPSO que não obteve nenhuma vitória.

Para uma análise da semelhança entre as metaheurísticas, foram utilizados os testes estatísticos Kruskal – Wallis (p_{valor}) e comparações múltiplas (z_{teste}) e a Tabela 13 apresenta os resultados da distância geracional invertida. Os com índice menor que 0,03 (ou 3%) apresentam discrepâncias e as metaheurísticas com índice superior a 0,03 apresentam amostras semelhantes ou não há diferença entre as metaheurísticas. Os valores obtidos são destacados em negritos de acordo com o valor aceitável de p_{valor} .

Ao analisar a similaridade dos resultados para as funções de *benchmark* com três objetivos utilizando a métrica HV (Tabela 11), o MOLAILS apresenta similaridade com MOLA nas funções (MaF4, MaF5, MaF6, MaF7, MaF8, MaF9, MaF11, MaF12, MaF13, MaF14 e MaF15). O MOPSO apresenta similaridade com MOLA nas funções (MaF1, MaF3, MaF5, MaF7, MaF11 e MaF12). O MOGWO apresenta similaridade com MOLA nas funções (MaF1, MaF2, MaF4, MaF5, MaF6, MaF7, MaF8, MaF9, MaF10, MaF11, MaF2 e MaF13). O MOEA/D apresenta similaridade com MOLA nas funções (MaF2, MaF6, MaF7 e MaF14). O MSSA apresenta similaridade com MOLA nas funções (MaF5, MaF7, MaF9, MaF11, MaF12, MaF14 e MaF15). O NSGA III apresenta similaridade com MOLA nas funções (MaF2, MaF7 e MaF8). Portanto o NSGA III é o algoritmo que apresenta a menor similaridade com o MOLA e o MOGWO é o algoritmo que representa a maior similaridade.

Agora ao analisar a similaridade dos resultados para as funções de *benchmark* com cinco objetivos utilizando a métrica HV (Tabela 11), o MOLAILS apresenta similaridade com MOLA nas funções (MaF1, MaF2, MaF3, MaF5, MaF6, MaF9 e MaF15). O MOPSO apresenta similaridade com MOLA nas funções (MaF2, MaF3, MaF4, MaF11, MaF12, MaF13 e MaF14).

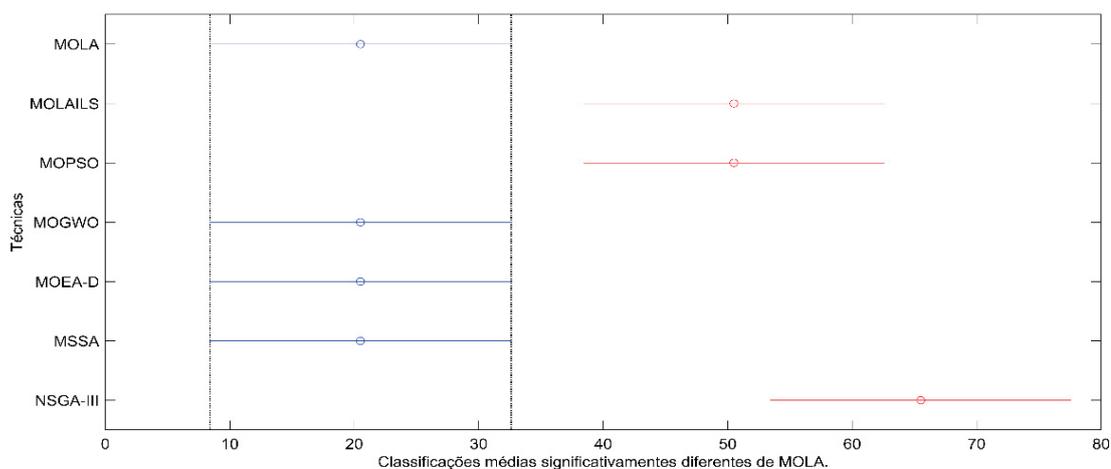
Tabela 13 – Teste estatístico da hipótese nula usando os resultados do HV.

P_{valor}	3 objetivos					5 objetivos					NSGA - III	MSSA	NSGA - III
	MOLAIS	MOPSO	MOGWO	MOEA-D	MSSA	NSGA - III	MOLAIS	MOPSO	MOGWO	MOEA-D			
MaF1	0,2913	0,0033	0	0,9999	0,0457	0,7826	0,0012	1	0	0,1362	0,0040	0,0864	
MaF2	0,0642	0,8307	0	0,0155	0,5844	0,0004	0,0040	0	0,1263	0,0014	0,0012	0,0001	
MaF3	0,1362	0	0,0650	0,1362	0,1362	0,8598	0,0022	0	0,6622	0	0	0,0022	
MaF4	0,0023	0,1362	0,0023	0,1362	0,1362	0,1362	0,1362	0	0,1362	0,1362	0,1362	0,8598	
MaF5	0,0002	0,0001	0	0,2269	0	0,5292	0,0001	0,2152	0	0,0835	0,0036	0	
MaF6	0,0012	0,2219	0	0,0022	0,2079	0,9015	0,0020	0,3281	0	0,0020	0,0805	0,8983	
MaF7	0,0048	0,0048	0	0	0	0,0048	0,0514	0,6578	0	0,4907	0,4907	0,0010	
MaF8	0	0,4627	0	0,6362	0,1494	0,0019	0,6366	0,0792	0	0,1056	0,0019	0,5612	
MaF9	0	0,9985	0	0,5031	0,0063	0,9949	0,0093	1	0	0,9800	0,0008	0,0008	
MaF10	0,9104	0,9989	0	0,9824	0,9989	0,9824	0,9067	0,9815	0	0,9943	0,9943	0,9989	
MaF11	0,0015	0	0	0,0798	0	0,6378	0,6378	0	0	0	0,0007	0,0689	
MaF12	0,0001	0	0	0,6398	0	0,0809	0,6378	0	0	0,0025	0,0119	0,0002	
MaF13	0,0009	0,9836	0	0,0721	0,4999	0,1138	0,1892	0,0019	0	0,2328	0,0001	0,7597	
MaF14	0,0048	0,9989	0,0650	0	0	0,5292	0,0514	0	0	0	0	0	
MaF15	0,9815	0,2219	0,0650	0,2269	0,0063	0,9989	0,0020	0,9989	0	0,9989	0,9989	0,0604	

O MOGWO apresenta similaridade com MOLA nas funções (MaF1, MaF5, MaF6, MaF7, MaF8, MaF9, MaF10, MaF11, MaF12, MaF13, MaF14 e MaF15). O MOEA/D apresenta similaridade com MOLA nas funções (MaF2, MaF3, MaF6, MaF11, MaF12 e MaF14). O MSSA apresenta similaridade com MOLA nas funções (MaF1, MaF2, MaF3, MaF5, MaF8, MaF9, MaF11, MaF12, MaF13 e MaF14). O NSGA III apresenta similaridade com MOLA nas funções (MaF2, MaF3, MaF5, MaF7, MaF9, MaF 12 e MaF14). Portanto o MOEA/D é o algoritmo que apresenta a menor similaridade com o MOLA e o MOGWO é o algoritmo que representa a maior similaridade.

A Figura 33 ilustra as classificações médias significativamente diferentes entre os algoritmos para a função de teste MaF7 com 3 objetivos com base nos dados apresentados na Tabela 13.

Figura 33 – Classificações médias usando a métrica HV.



Fonte: Autora, 2024.

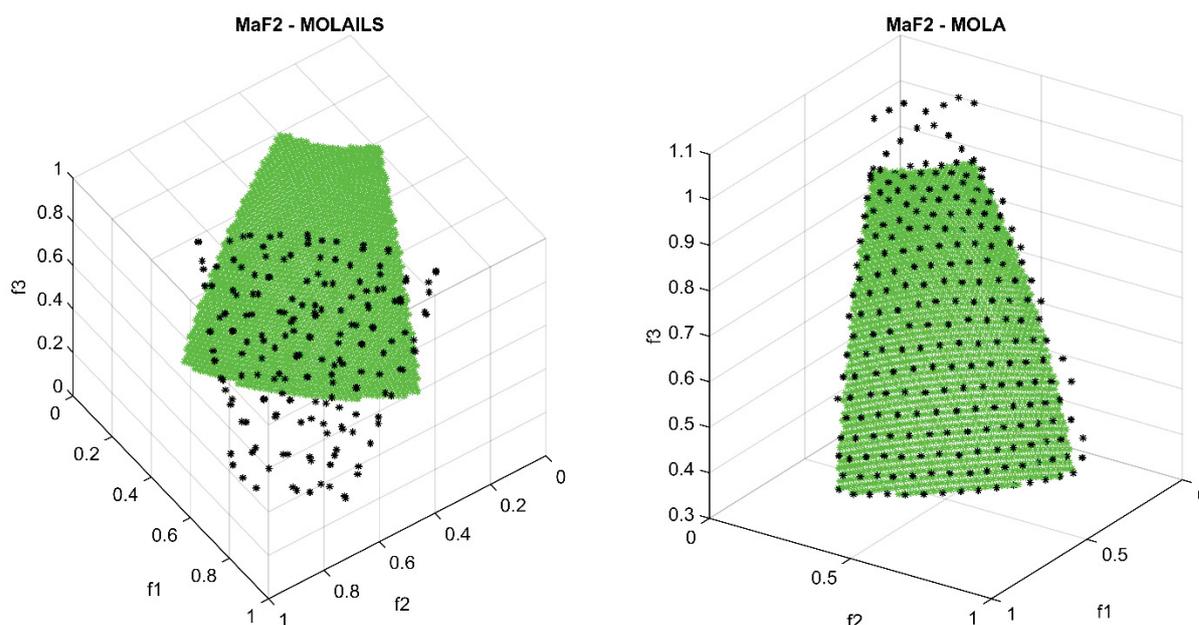
O algoritmo MOLA possui classificações médias significativamente diferentes em relação aos algoritmos MOLAILS, MOPSO e NSGA III (Figura 20), por exemplo ao utilizar a MaF7 com 5 objetivos. Neste sentido, se os algoritmos não contiver classificações médias aproximadamente de 32 conforme o intervalo (linha pontilhada – Figura 20), pode-se dizer que as médias correspondentes serão significativamente diferentes. Entretanto as técnicas MOLA, MOGWO, MOEA/D e MSSA apresentaram classificações médias diferentes da proposta MOLAILS, como exemplo, para a MaF7 com 5 objetivos (Figura 33).

5.1.2.3. Análise gráfica das respostas do MOLA e MOLAILS

Nesta subseção serão apresentados alguns resultados dos algoritmos MOLA e MOLAILS, cujas respostas são plotadas junto com as respectivas fronteiras de Pareto ótima.

Ao analisar a Figura 34 observa que a metaheurística MOLA tem a maioria das suas soluções na fronteira de Pareto ótima, enquanto a metaheurística MOLAILS apresenta menor volume de soluções e mais dispersas dentro da fronteira de Pareto ótima. Isso corrobora os resultados obtidos pelas métricas IGD e HV, sendo os melhores resultados imputados ao MOLA ao lidar com fronteiras fronteira com propriedade côncava (MaF 2).

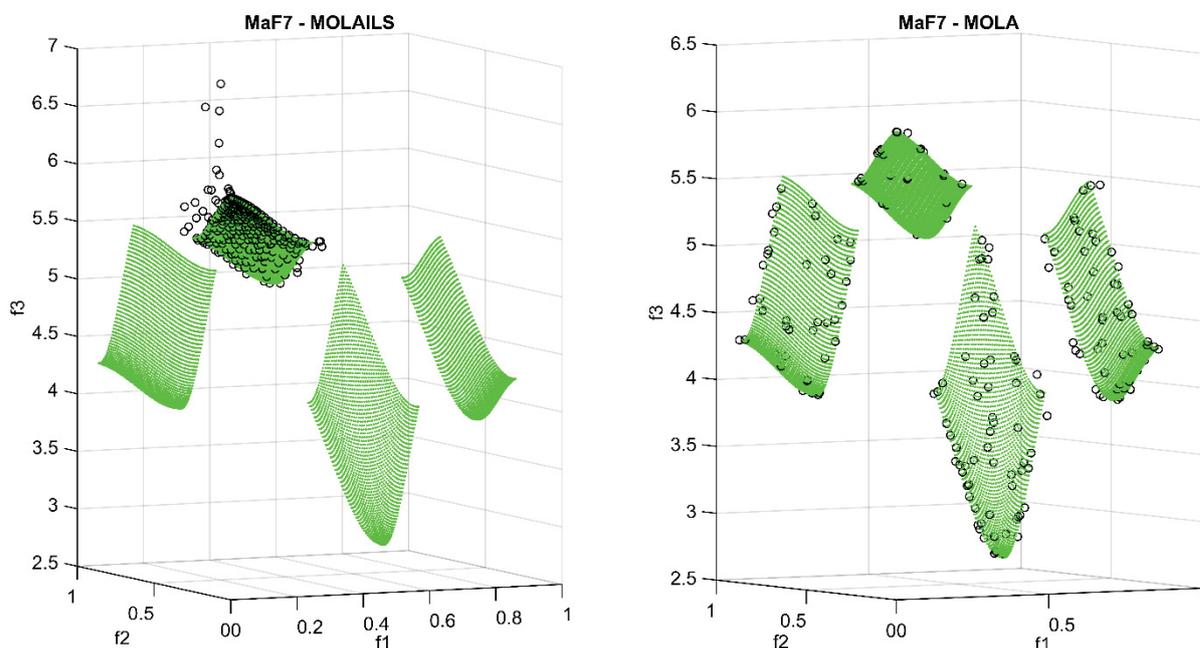
Figura 34 – Soluções obtidas por MOLA e MOLAILS para MaF2.



Fonte: Autora, 2024.

A Figura 35 apresentam a fronteira de Pareto ótima da função MaF7 com 3 objetivos trata-se de uma fronteira com propriedade mista, desconectado e multimodal ao mesmo tempo e as respectivas soluções encontradas pelas metaheurísticas MOLA e MOLAILS.

Figura 35 – Soluções obtidas por MOLA para MaF7.



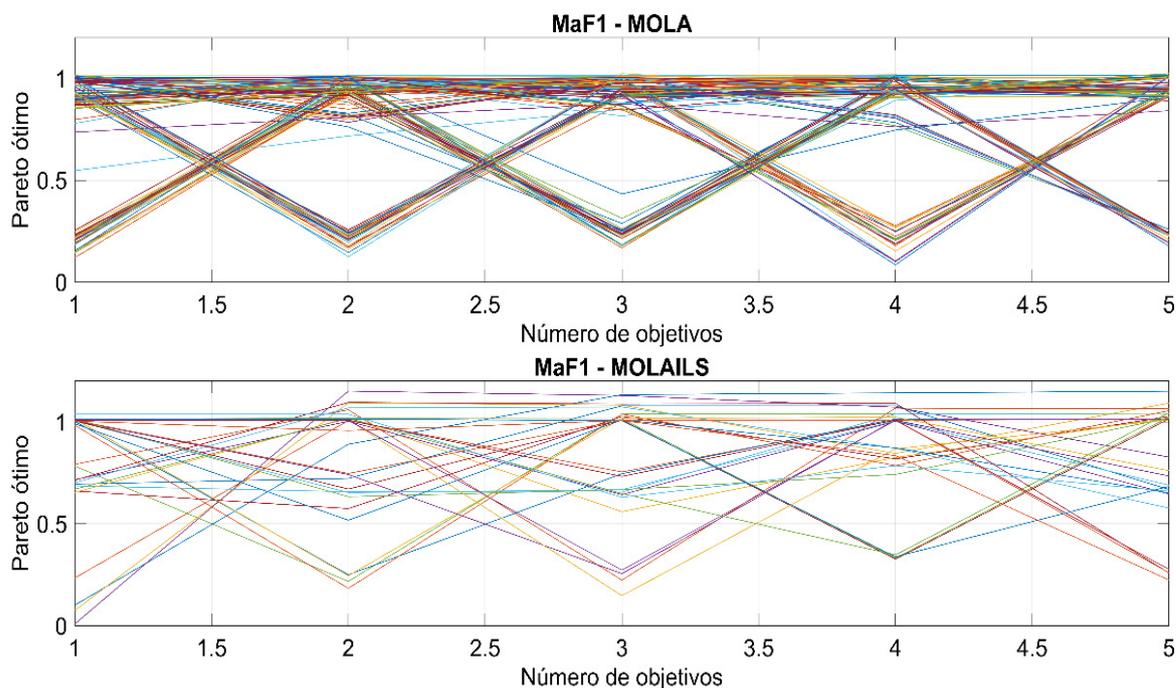
Fonte: Autora, 2024.

Ao analisar as Figura 35 observa que as metaheurísticas MOLA e MOLAILS tem a maioria das suas soluções na fronteira de Pareto ótima, a diferença da quantidade de soluções entre os algoritmos é por se estar utilizando os hiper parâmetros do RVEA na metaheurística MOLAILS. Esses resultados corrobora os resultados obtidos pelas métricas IGD e HV, sendo o melhor resultado da métrica IGD obtido pelo MOLA, enquanto na métrica HV o melhor resultado é obtido pelo MOLAILS.

A Figura 36 apresentam a fronteira de Pareto ótima da função MaF1, trata-se de uma fronteira com propriedade lineares, para 5 objetivos e as respectivas soluções encontradas pelas metaheurísticas MOLA e MOLAILS.

Ao analisar a Figura 36 observa que a metaheurística MOLA possui uma quantidade maior de soluções na fronteira de Pareto ótima compara a metaheurística MOLAILS. Isso ocorre provavelmente pelo algoritmo MOLAILS estar utilizando os hiper parâmetros do algoritmo RVEA. Esses resultados corrobora os resultados obtidos pelas métricas IGD e HV, sendo o melhor resultado da métrica IGD obtido pelos dois algoritmos MOLA, enquanto na métrica HV o melhor resultado é obtido pelo MOLAILS.

Figura 36 – Soluções obtidas por MOLA e MOLAILS para MaF1.



Fonte: Autora, 2024.

5.1.3. Ambiente experimental e tempo de computação

Todos os experimentos deste estudo foram conduzidos utilizando o software MATLAB® em um notebook Samsung com Windows 10, 64 bits, 11ª Geração Intel(R) Core (TM) i5 2.40GHz e memória RAM de 8 GB. Os tempos computacionais médios dos algoritmos são dados na Tabela 14.

Da Tabela 14, o tempo computacional foi destacado em vermelho para as quinze MaFs com 3 objetivos e em azul para as MaFs com 5 objetivos e pode-se observar que o algoritmo MOLA é o mais ágil em termos de tempo computacional comparado com MOLAILS, MOPSO, MOGWO, MOEDA/D, NSGA-III e MSSA.

Tabela 14 – Tempo computacional dos algoritmos multiobjetivo.

Tempo Médio Computacional (segundos)								
Função	M	MOLA	MOLAILS	MOPSO	MOGWO	MOEA/D	MSALP	NSGA III
MaF1	3	3.211	7.322	33.14	201.9	42.55	117.0	4.95
	5	4.580	6,231	76.12	404.8	41.39	80.82	6.81
MaF2	3	2.987	5,371	27.04	342.1	42.08	88.92	6.55

	5	3.122	9,035	104.1	288.6	40.90	95.86	3.58
MaF3	3	10.24	6,197	36.684	300.79	42.494	108.1	26.4
	5	4.599	8,301	124.72	410.42	41.459	508.7	8.96
MaF4	3	15.293	18,233	70.545	377.78	43.220	302.8	55.17
	5	10.30	11,215	277.60	744.29	41.541	753.7	17.43
MaF5	3	4.106	9,062	13.952	287.67	42.610	66.80	8.072
	5	4.908	10,088	26.334	340.95	41.497	81.79	2.903
MaF6	3	3.13	10,531	9.1722	515.87	43.459	23.70	3.553
	5	3.908	20,358	14.698	599.19	46.301	43.04	3.107
MaF7	3	2,848	25,261	18,517	298,53	44,912	32.54	3.476
	5	3,009	19,124	43,144	400,12	42,597	40.31	2.918
MaF8	3	1,692	12,150	19,344	$5,07e^3$	42,806	105.9	2.440
	5	1,922	19,367	41,210	$1,83e^3$	43,490	128.7	2.055
MaF9	3	2,312	8,296	5,909	$1,22e^3$	42,824	49.28	1.928
	5	3,120	6,747	7,134	27,617	44,007	79.34	3.540
MaF10	3	2,108	4,234	18,69	10,991	42,889	83.61	2.922
	5	5,565	5,401	30,05	9,324	43,179	51.02	5.824
MaF11	3	3,002	6,725	40,114	26,624	43,916	141.6	4.761
	5	4,324	5,362	42,307	14,338	42,747	203.1	4.769
MaF12	3	6,413	8,348	20,735	28,850	42,572	324.4	8.244
	5	5,719	5,307	22,001	25,699	43,110	105.1	6.124
MaF13	3	2,552	6,237	13,049	18,227	43,665	97.08	4.549
	5	1,262	5,184	13,855	11,163	42,240	93.98	4.238
MaF14	3	3,334	7,949	7,3871	30,271	42,717	121.3	5.687
	5	4,087	15,348	53,694	89,249	44,362	136.7	4.485
MaF15	3	3.984	11,234	7,5501	25,135	43,042	105.1	5.362
	5	4,556	15,231	47,431	46,276	44,083	240.5	6.872

Em relação ao tempo computacional, NSGA III e MOLAILS apresentaram grande agilidade para as quinze funções de teste. O MOGWO foi o algoritmo que apresentou o pior custo computacional para as quinze funções de *benchmark*.

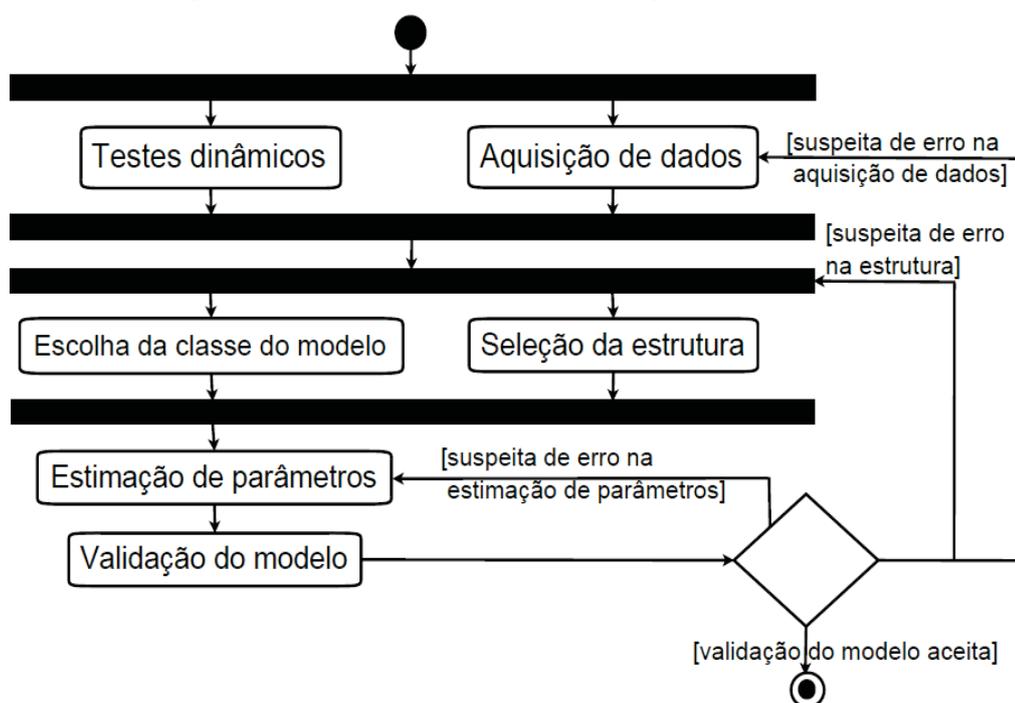
5.2. MODELOS EM ENGENHARIA

A área responsável para a obtenção de modelos matemáticos de sistemas dinâmicos é a identificação de Sistemas. Problemas que envolvem identificação de sistemas podem ser formulados como um problema de otimização, no qual definimos o problema com um objetivo ou com vários objetivos que podem ser minimizadas considerando os parâmetros do problema (Aguirre, 2015).

A classificação da Identificação de Sistemas está associada a que tipo de modelagem se deseja do sistema real, as quais são definidas como: modelagem caixa branca, modelagem caixa preta e modelagem caixa cinza.

Nos modelos caixa branca, tem-se o conhecimento pleno do sistema, ou seja, suas variáveis e equações são conhecidas. Nos modelos caixa preta, somente as variáveis de entrada e saída são conhecidas e os parâmetros do modelo são considerados constantes matemáticas e não possuem interpretações físicas. Nos modelos caixa cinza há alguma informação a priori (vindo de testes empíricos, por exemplo) que são acrescentados na modelagem do sistema identificado para melhorar sua representatividade (Martins; Nepomuceno, 2016).

Figura 37 – Processo de Identificação de Sistemas.



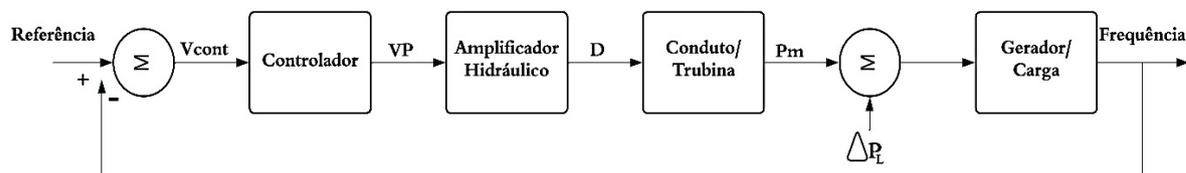
Fonte: Adaptada de Souza (2014).

A identificação de sistemas (Figura 37) é dividida em quatro etapas primordiais, que são: escolha dos dados a serem utilizadas, a escolha da estrutura, estimação de parâmetros e a validação do modelo. Os aspectos da identificação de sistemas são ditos de etapas fundamentais para se obter um “bom” modelo.

5.2.1. Estudo de caso 1: Modelagem da malha hidráulica e turbina hidráulica de uma usina elétrica.

Nesta modelagem, parte-se do princípio de que algumas propriedades/etapas são conhecidas, por exemplo, a estrutura do modelo, a ordem do modelo, etc. Porém, ainda há necessidade de se determinar propriedades para a completa representação do sistema (Aguirre, 2015). O modelo do sistema é do tipo caixa cinza, pois é um sistema real onde se tem o conhecimento das variáveis de entrada e saída, e parâmetros deverão ser estimados. Na Figura 38 pode-se visualizar o funcionamento do sistema sob estudo.

Figura 38 – Sistema Não Linear sob Estudo.



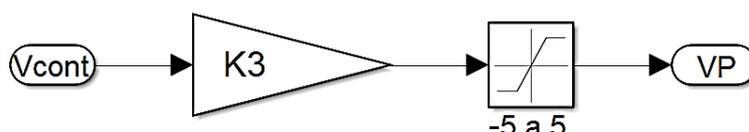
Fonte: Autora, 2024.

O governador consiste no controlador de velocidade e no amplificador hidráulico. Esses dois sistemas (controlador de velocidade e amplificador hidráulico) controlam a frequência da tensão terminal gerada pelo gerador síncrono. O controlador de velocidade regula a velocidade da mudança na posição do distribuidor da turbina hidráulica e, com a ajuda do controlador de carga de frequência (do termo em inglês, *Load Frequency Controller* – LFC), pode manter o sistema estabilizado (Donaisky et al. 2016). O amplificador hidráulico é um servo atuador de válvula que opera por meio do piloto linear. Devido a uma grande força que atua no sistema, são necessários mecanismos de controle de potência para operar no acionamento destas válvulas. Dentro do amplificador hidráulico atuam três

válvulas fundamentais (ou subsistemas) para seu desempenho, que são: válvula piloto, válvula distribuidora e distribuidor.

Posto isto, a Figura 39 mostra o modelo do primeiro subsistema que faz parte da malha hidráulica.

Figura 39 – Modelagem da Válvula Piloto/Proporcional.



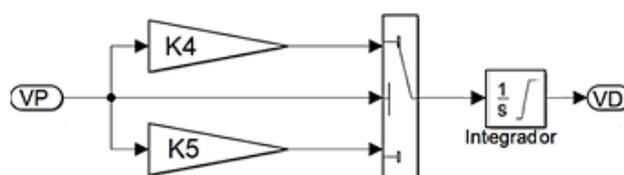
Fonte: Autora, 2024.

A válvula proporcional ou válvula piloto (VP) recebe o sinal de entrada do controlador (V_{cont}) modifica os sinais de controle elétrico em um deslocamento mecânico-hidráulico proporcional, atuando diretamente na válvula distribuidora (VD).

Esta válvula é composta pelo ganho (K_3) que representa a dinâmica da válvula, e o bloco na sequência representa uma saturação cujas condições são: limite superior ($U_1 = 5$); e limite inferior ($L_1 = -5$). A entrada da válvula é o sinal de controle (V_{cont}) e a saída (VP) resulta no sinal da válvula proporcional. A Figura 40 ilustra o segundo subsistema da malha hidráulica.

A válvula distribuidora (VD) regula as diretrizes de posição através da abertura ou fechamento dos bicos dos servomotores (distribuidor). Além disso, ela depende dos dois parâmetros: K_4 e K_5 , o integrador varia em limite superior ($U_2 = 8$) e limite inferior ($L_2 = -8$) da válvula distribuidora.

Figura 40 – Modelagem da Válvula Distribuidora.

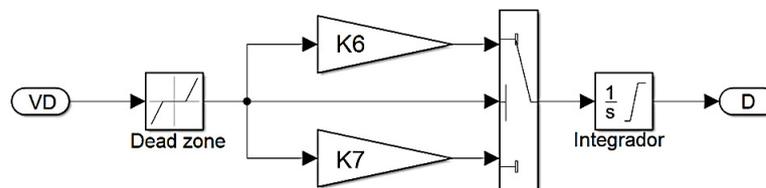


Fonte: Autora, 2024.

A entrada deste sistema é a saída da válvula piloto. O sinal da válvula piloto define a abertura ou fechamento da válvula distribuidora, com isso tem-se dois ganhos, sendo cada um correspondente ao sentido do movimento de deslocamento

da válvula. O último subsistema da malha hidráulica é o Distribuidor e está representado na Figura 41.

Figura 41 – Modelagem do Distribuidor.

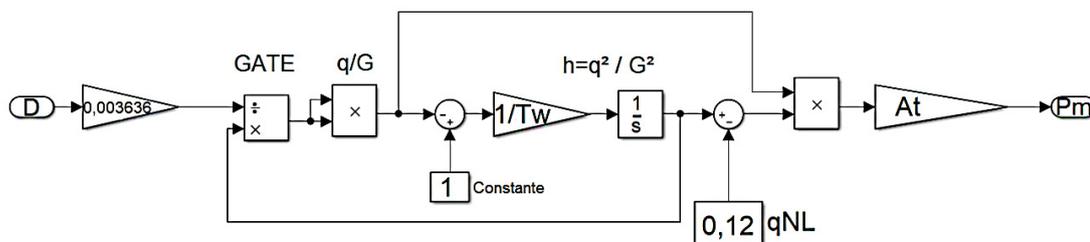


Fonte: Autora, 2024.

O distribuidor (D) é um elemento fixo que serve para controlar o fluxo de água através do rotor, controlando assim a energia fornecida para a turbina. Além disso, transforma parcialmente a pressão da água em energia cinética na entrada do rotor. Além disso, o distribuidor depende dos parâmetros $K6$ e $K7$, a condição inicial do integrador é gerada entre os limites superior ($U_3 = 275$) e inferior ($L_3 = 0$). Uma não linearidade do tipo zona morta, representando o recobrimento da válvula, é inclusa no modelo MENDELEY CITATION PLACEHOLDER 0. A entrada deste sistema é a saída da válvula distribuidora. O sinal da válvula distribuidora define a abertura ou fechamento do servomotor do distribuidor, com isso tem-se dois ganhos, sendo cada um correspondente ao sentido do movimento de deslocamento do servomotor e a saída deste sistema é o deslocamento do servomotor.

Por fim, a Figura 42 ilustra a modelagem da turbina hidráulica utilizada neste trabalho.

Figura 42 – Modelagem da Turbina Hidráulica.



Fonte: Adaptada de IEEE (1992).

A turbina hidráulica utilizada neste trabalho é o modelo IEEE e pode ser encontrada na literatura (IEEE, 1992). A turbina depende da malha hidráulica (válvulas e distribuidor), o qual é a chave principal do sistema. Além disso, é considerada uma máquina que transforma energia potencial em energia mecânica e sua potência é calculada pelo modelo mostrado na Figura 42, onde h é a altura nominal em pu, q é o fluxo de água em pu, G é a posição da porta do distribuidor em pu, q_{NL} é a vazão nominal da água, T_w é constante de tempo de inércia da água em segundos, A_t é a relação para mudança de base da turbina para o gerador, finalmente P_m é a potência mecânica.

5.2.1.1. Estimativa de parâmetros do HTRS

O problema de estimativa de parâmetros do HTRS pode ser considerado um problema de otimização, e algoritmos meta-heurísticos têm sido usados como métodos importantes para obter as soluções ótimas para a otimização prática de projetos de engenharia, uma vez que os problemas são amplamente aplicados nesta tarefa. Para estudos é proposto um método metaheurística multiobjetiva MOLA para resolver problemas de estimação de parâmetros.

Para estimação dos parâmetros utiliza-se o modelo mostrado na Figura 38, onde o controlador é um PID e o gerador/carga é modelado como sistema de primeira ordem.

A minimização da função objetivo é dada pela Equação (5.11).

$$\begin{aligned}
 f_1(x) &= VP - x_1 \widehat{V}_{cont} \\
 f_2(x) &= VD - \begin{cases} \int x_2 g(f_1(x)) & \text{se } f_1(x) \geq 0 \\ \int x_3 g(f_1(x)) & \text{se } f_1(x) < 0 \end{cases} \\
 f_3(x) &= D - \begin{cases} \int x_4 g(f_2(x)) & \text{se } f_2(x) \geq 0 \\ \int x_5 g(f_2(x)) & \text{se } f_2(x) < 0 \end{cases} \\
 f_4(x) &= P_m - ((\int \left[1 - \frac{q}{\frac{0,0036 f_3(x)}{x_6}} \right] - 0,12) h x_7
 \end{aligned} \tag{5.11}$$

sendo $x = [K_3 K_4 K_5 K_6 K_7 T_w A_t]$, \hat{V}_c é o sinal obtido depois do ganho (oito valores) $g()$ são funções de saturação e zona morta.

q é fluxo.

h é o fluxo ao quadrado dividido pela abertura da porta ao quadrado.

A função minimiza o erro entre a saída da válvula piloto e a saída estimada após limitação por $L3$ e $L4$. A função minimiza o erro entre a saída da válvula distribuidora e a saída estimada após limitação por $L5$ e $L6$. A função minimiza o erro entre a saída do distribuidor e a saída estimada após limitação por $L7$ e $L8$. E a função minimiza o erro entre a potência mecânica e a potência mecânica estimada.

- Experimentos e análise de resultados de HTGS

Para o problema de HTGS, sete parâmetros são escolhidos para serem estimados na simulação que são K_3 K_4 K_5 K_6 K_7 Tw e At . Os dados obtidos são selecionados sob duas condições operacionais. Uma etapa de potência cujo valor inicial foi de 0,7512 pu (99,9107 MW) a 0,5467 pu (72,7127 MW), outra etapa de potência cujo valor inicial foi de 0,5303 pu (70,5269 MW) a 0,7468 (99,3277 MW) que será utilizada na estimativa e validação respectivamente. Os dados obtidos da UHE foram amostrados a uma taxa de 1 ms, que foi então reamostrada (decimação) para 10 ms.

A Tabela 15 apresenta a faixa de valores em que os parâmetros são encontrados, para que a metaheurística encontre o melhor valor para cada parâmetro, os valores definidos para essas faixas foram baseados em testes de bancada, manuais dos fabricantes e dados fornecidos pela UHE.

Tabela 15 – Parâmetro do modelo estudado.

Parâmetro	Valor
Ganho da válvula piloto K_3	[0,65 0,75]
Ganho da válvula distribuidora K_4	[4 5]
Ganho do servo motor do distribuidor K_5	[3 4]
Ganho do servo motor do distribuidor K_6	[7 8]
Ganho do servo motor do distribuidor K_7	[13 15]
Relação com a mudança de base At	[1,0 1,3]
Hora de início da água Tw	[2,5 2.9]

Além dos parâmetros de configuração algoritmos MOLA, MOLAILS e MOGWO fornecidos na Tabela 7, outros parâmetros de otimização foram utilizados para o estudo de caso HTGS, como: a dimensionalidade do problema (D) é sete; número máximo da população (N) é igual a setenta, o número de objetivos (M) é igual a quatro (Equação (5.1)), o número máximo de gerações é igual duzentos, o critério de parada é igual a 0,003 e o total de execuções é cinquenta.

A Tabela 16 apresenta os valores obtidos da estimação do parâmetro para MOLA executado 50 vezes.

Tabela 16 – Estimação de parâmetros utilizando MOLA.

MOLA				
	Mínimo	Média	Máximo	Std
K3	0.6667	0.6936	0.7253	0.0274
K4	4.0495	4.3667	4.7212	0.3264
K5	3.1854	3.4366	3.8860	0.3096
K6	7.6805	7.7767	7.9203	0.1037
K7	13.657	13.689	14.7982	0.2544
Tw	2.6379	2.7394	2.8741	0.1192
At	1.1175	1.1561	1.1912	0.0366

O MOLAILS é executado 50 vezes. A Tabela 17 apresenta os valores obtidos da estimativa dos parâmetros obtidos pelo algoritmo MOLAILS.

Tabela 17 – Estimativa de parâmetros usando MOLAILS.

MOLAILS				
	Mínimo	Média	Máximo	Std
K3	0.6612	0.6942	0.7219	0.0265
K4	4.0481	4.3659	4.6997	0.3001
K5	3.1850	3.4360	3.8839	0.3057
K6	7.6687	8.0004	8.0105	0.1041
K7	13.0014	13.677	14.1223	0.2827
Tw	2.6400	2.7232	2.8794	0.1200

At	1.1155	1.1580	1.1859	0.0369
----	--------	---------------	--------	--------

O MOGWO é executado 50 vezes. A Tabela 18 apresenta os valores obtidos da estimativa dos parâmetros obtidos pelo algoritmo MOGWO.

Tabela 18 – Estimativa de parâmetros usando MOGWO.

MOGWO				
	Mínimo	Média	Máximo	Std
K3	0,6509	0,6899	0,7068	0,0222
K4	4,0429	4,3604	4,6527	0,2971
K5	3,1655	3,4360	3,8711	0,3097
K6	7,8987	8,0012	8,2314	0,1030
K7	12,9980	13,7709	14,7785	0,3009
Tw	2,6061	2,6487	2,8804	0,2125
At	1,1098	1,1601	1,1822	0,0806

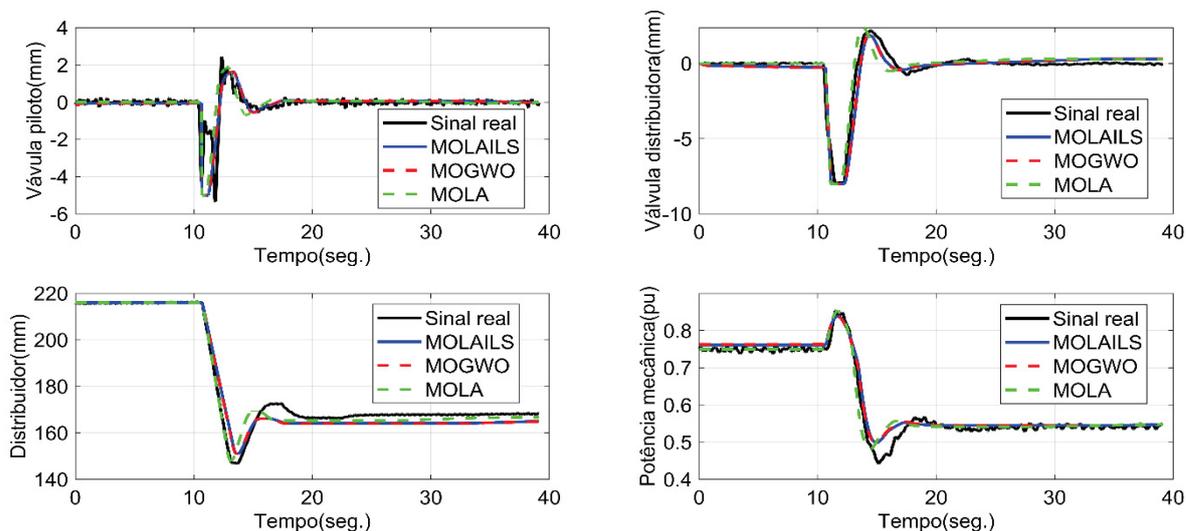
As soluções que melhor se aproximam das quatro funções objetivo foram escolhidas como aquelas em que as quatro funções objetivo foram unificadas pela soma dos respectivos erros quadráticos ponderados, obtendo-se o menor valor. Os pesos foram determinados de acordo com os valores de aptidão das respectivas funções. Por se tratar de um problema de minimização, o menor fitness foi ponderado com valores maiores e a soma dos pesos é sempre igual a 1 (um) e pré-definidos pelo codificador. A Equação (5.12) mostra os pesos para a função objetivos.

$$fw = \sum_{i=1}^k w_i f_i(x), w_1 f_1(x), w_2 f_2(x), w_3 f_3(x), w_4 f_4(x) \quad (5.12)$$

onde, $w_1 = 0.4$; $w_2 = 0.2$; $w_3 = 0.1$, $w_4 = 0.3$ e para as funções objetivos $f_1(x)$, $f_2(x)$, $f_3(x)$ e $f_4(x)$ seus respectivos pesos são baseados nos valores das funções objetivos, ou seja, as funções que apresentaram menor valor de fitness terão pesos (w) maiores por se tratar de um problema de minimização. No caso do presente trabalho as funções $f_1(x)$ e $f_4(x)$ terão pesos maiores do que as funções $f_2(x)$ e $f_3(x)$.

A Figura 43 mostra os resultados estimados obtidos usando os valores médios das Tabelas 16 – 18.

Figura 43 – Desempenho do sistema regulador de turbina hidráulica.



Fonte: Autora, 2024.

O objetivo é observar se o modelo reproduz ao longo do tempo a saída medida do sistema quando submetido à mesma entrada medida. Nota-se boa proximidade entre o sinal real da UHE e os obtidos pelos modelos. Isto indica a eficiência dos métodos para resolver o problema de estimação de parâmetros destes sistemas complexos.

Os resultados de desempenho utilizando RMSE, RMSLE, MAE e ITAE obtidos de MOLA, MOLAIS e MOGWO são mostrados na Tabela 19.

Tabela 19 – Resultados de desempenho usando MOLA e MOLAIS.

MOLA			MOLAIS			MOGWO		
Métricas	Média	Std	Métricas	Média	Std	Métricas	Média	Std
ITAE	316,2	443,1	ITAE	355,6	420,7	ITAE	408,2	5120,0
MAE	1,1878	0,0408	MAE	1,1668	0,0566	MAE	1,1021	0,0913
RMSE	1,7892	0,1197	RMSE	1,7466	0,0199	RMSE	1,7009	0,2011
RMSLE	0,7378	0,0222	RMSLE	0,7445	0,0205	RMSLE	0,7870	0,0402

Os resultados apresentados na Tabela 19 confirmam que MOLA, MOLAILS e MOGWO são métodos viáveis para lidar com problemas de minimização para tratar da estimação de parâmetros. Além disso, pode-se perceber que as duas propostas, MOLA e MOLAILS apresentaram valores médios semelhantes ao do algoritmo MOGWO. Os critérios de desempenho RMSE, RMSLE e ITAE apresentaram menor variância na busca do conjunto de Pareto para a proposta MOLAILS. Pode-se observar que o algoritmo MOLAILS fica em segundo lugar em relação aos algoritmos MOLA e MOGWO para as métricas: ITAE, MAE, RMSE e RMSLE, enquanto o algoritmo MOLA teve melhor desempenho ao utilizar as métricas RMSLE e ITAE. Por fim, o algoritmo MOGWO apresentou bom desempenho para as métricas MAE e RMSE.

O tempo computacional deste estudo é maior em relação ao teste de quinze funções. O tempo de minimização utilizando o algoritmo MOLA foi $3,5680e+3$ com RMSE, $1,7386e+4$ com RMSLE, $3,9935e+3$ com MAE e $1,4551e+3$ com ITAE. No MOLAILS RMSE o tempo computacional médio foi $4,0371e+3$, RMSLE foi $3,0124e+4$, MAE foi $2,2199e+3$ e, ITAE foi $1,3739e+3$. Para o algoritmo MOGWO o tempo computacional foi de RMSE $6,2124e+5$, RMSLE de $4,3337e+5$, MAE de $5,0998e+5$ e ITAE de $5,0058e+6$. Os três algoritmos apresentaram tempo computacional elevado, mas o tempo computacional torna-se relativo porque depende da complexidade dos sistemas não lineares, ou seja, das características individuais dos sistemas não lineares (como exemplo o número de parâmetros a serem estimados). A validação do modelo é uma das etapas importantes no processo de identificação do sistema. Os testes de validação aplicados a um modelo indicam sua capacidade de representar com sucesso o sistema original para todos os sinais de entrada, mesmo aqueles não utilizados durante a estimativa dos parâmetros.

As saídas do sistema foram obtidas usando entradas medidas de outra condição operacional que não foi usada para estimativa de parâmetros. Os valores de desempenho são apresentados na Tabela 20. Os parâmetros dos modelos foram os valores médios semeados na Tabela 16 para o MOLA, na Tabela 17 para o MOLAILS e Tabela 18 para MOGWO. Todos os erros foram pequenos e a dispersão de dados em relação ao desvio padrão, as métricas RMSE, RMSLE e MAE apresentam uma boa distribuição das soluções.

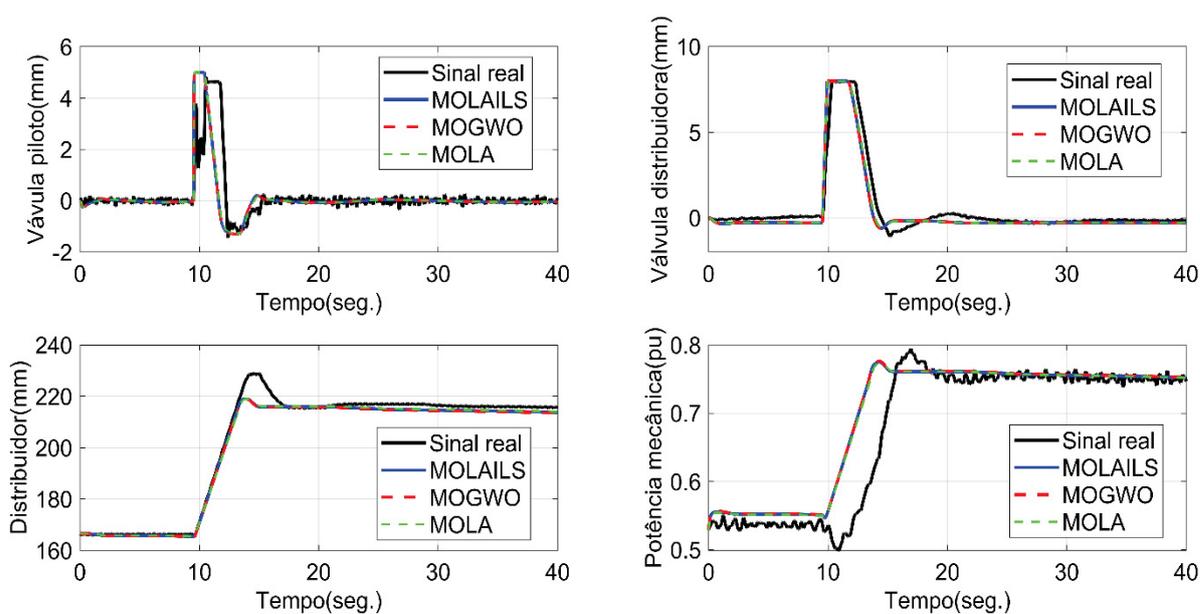
Tabela 20 – Resultados médios de desempenho usando MOLA e MOLAILS.

MOLA		MOLAILS		MOGWO	
Métricas	Valor	Métricas	Valor	Métricas	Valor
ITAE	320.1025	ITAE	335.2827	ITAE	409,1214
MAE	1.1902	MAE	1.1915	MAE	1,1106
RMSE	1.9904	RMSE	1.9793	RMSE	1,8825
RMSLE	0.8107	RMSLE	0.7989	RMSLE	0,8001

A validação dos dados pode ser considerada uma das etapas mais importantes, pois analisa a relação entre as variáveis de entrada e saída não representadas por equações lineares. Com isto, essa etapa visa validar o comportamento dos dados que não foram utilizadas na simulação (Figura 43).

Ao analisarmos a Figura 44, pode-se notar boa concordância entre a saída medida do sistema e as saídas dos modelos mesmo utilizando condições diferentes para ambos os algoritmos propostos (MOLA e MOLAILS) e algoritmo da literatura MOGWO.

Figura 44 – Dados de validação do sistema regulador de turbina hidráulica.



Fonte: Autora,2024.

5.2.2. Estudo de Caso 2: Interação Fluido-Estrutura com um Grau de Liberdade

O problema da interação fluido-estrutura pode ser considerado um acoplamento não estacionário entre o fluxo de fluido e a estrutura mecânica flexível, não havendo garantia de que as condições de equilíbrio serão satisfeitas: as vibrações e a dinâmica da estrutura podem, por exemplo, ter um impacto. Além do mais, este tipo de problema está presente em diversas áreas da engenharia, como exemplo, na civil (a interação entre as vigas de sustentação de uma ponte com o fluxo da água) na aeronáutica (a interação das asas do avião com a pressão do ar), na biomedicina (o fluxo sanguíneo e a parede das veias) são práticas e desafiadoras.

Por ser um problema de alta complexidade computacional, acaba sendo desafiador de se equacionar o fluxo de fluidos devido aos possíveis efeitos consecutivos e de turbulência. Sendo necessário ter cautela ao se acoplar a estrutura e solucionadores de fluido.

A figura 45 exemplifica o problema da interação fluido-estrutura de um sistema de massa-mola-amortecimento para um grau de liberdade (GDL) utilizado no presente trabalho reproduzido por Silva (2008).

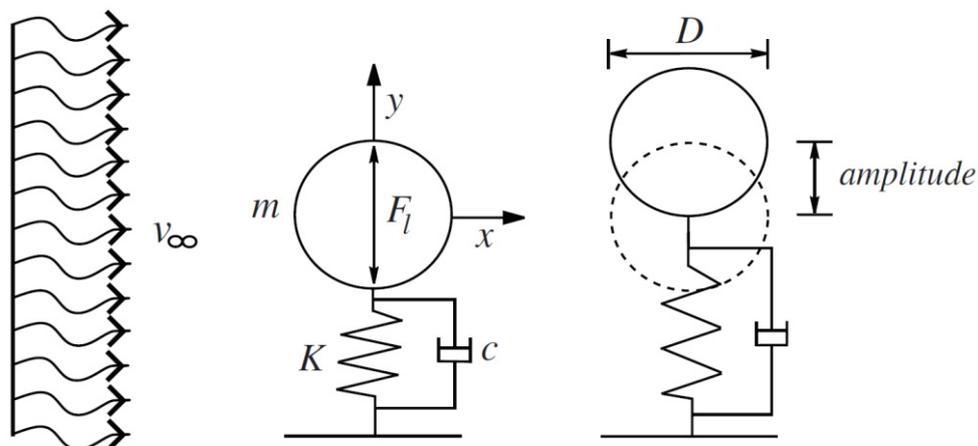
Para o sistema massa-mola-amortecimento para 1 GDL se tem: o cilindro bidimensional de massa m com diâmetro D , acoplado em uma mola linear de rigidez K de fator de amortecimento c , imerso em um fluido com velocidade v_{∞} . E, ao considerar uma força única atuando sobre a força de sustentação (F_l), proveniente do fluido, a força restauradora da mola com o amortecimento do tipo linear, a Equação (5.2) sintetiza o movimento

$$m \frac{d^2y}{dt^2} + c \frac{dy}{dt} + Ky = F_l \quad (5.2)$$

sendo o deslocamento transversal do cilindro (y).

Uma informação importante a ser observada é sobre se a presença de fluido é desconsiderada ou não no sistema. Caso a presença de fluido seja desconsiderada no sistema (Figura 45), o cilindro ficará livre para responder à força que atuar sobre o mesmo.

Figura 45 – Representação do sistema massa-mola-amortecimento para 1 GDL.



Fonte: Silva (2008).

Mas, se o fluido for considerado no sistema o cilindro terá uma limitação definido pela mola em direção perpendicular ao escoamento incidente e com as restrições de descolamento em outra direção. Portanto, o cilindro poderá atuar como um corpo rígido, locomovendo-se por meio do fluido e restrito pelo suporte elástico.

Em Lobato (2008), o problema direto de interação fluido-estrutura foi tratado através do acoplamento entre o método da Fronteira Imersa e o modelo Físico virtual, onde as interações entre o fluido e a estrutura (no caso será o cilindro) são representadas por equações de Navier-Stokes e do sistema massa-mola-amortecimento (Figura 30). Além disso, uma discretização foi realizada na equação de Navier-Stokes usando as diferenças finitas de segunda ordem no espaço e a discretização temporais de Admas-Blashforth e Runge-Kutta de segunda ordem. (Silva, 2008).

A modelagem física do problema direto da interação fluido-estrutura está disponibilizada em Lobato (2008) e Silva (2008). O modelo apresenta como respostas: o coeficiente de arrasto (C_d), o coeficiente de sustentação (C_l), o deslocamento (y) e a amplitude (A) máxima do cilindro, dadas pelas Equações (5.3) e (5.4).

$$C_d = \frac{F_d}{0,5\rho v_\infty^2 D} \quad (5.3)$$

$$C_l = \frac{F_l}{0,5\rho v_\infty^2 D} \quad (5.4)$$

onde F_d é a força de arrasto, ρ é a densidade e v_∞^2 é a velocidade.

Para o planejamento de experimentos e simulações numéricas, o autor Lobato (2008) indica as variáveis independentes para o problema, que são sintetizadas nas Equações (5.5) a (5.7).

$$Re = \frac{Dv_\infty\rho}{\mu} \quad (5.5)$$

$$m' = \frac{m}{\rho\pi R^2 L} \quad (5.6)$$

$$Vr = \frac{v_\infty}{f_n D} \quad (5.7)$$

sendo R o raio do cilindro com diâmetro D , o comprimento do cilindro é L , a viscosidade cinemática é μ , a frequência natural do cilindro é f_n e as coordenadas do cilindro são $16,5D$ na direção x e $15D$ na direção y .

Estas variáveis independentes foram geradas entre o limite inferior e limite superior: $8000 \leq Re \leq 80000$, $1 \leq m' \leq 30$ e $1 \leq Vr \leq 16$. Como entrada do modelo se tem as seguintes variáveis x_i ($i = 1,2,3$) que são dadas pelas Equações (5.8) – (5.10).

$$x_1 = \frac{Re-44000}{28000} \quad (5.8)$$

$$x_2 = \frac{Vr-8,5}{6} \quad (5.9)$$

$$x_3 = \frac{m'-15}{11} \quad (5.10)$$

5.2.2.1. Estimativa do Problema Direto da Interação Fluido-Estrutura

Para realizar a estimativa de parâmetros do problema Direto da Interação Fluido-Estrutura para 1 GDL foi utilizado como dados de referência apresentados nas Tabelas 20 e para se obter as respostas simuladas $Y_{sr} = [C_{dsr}, C_{lsr}, y_{sr} \text{ e } A_{sr}]$ é dada pela Equação (5.13)

$$Y_{sr} = \{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1 x_2 + \beta_5 x_1 x_3 + \beta_6 x_2 x_3 + \beta_7 x_1^2 + \beta_8 x_2^2 + \beta_9 x_3^2 + \beta_{10} x_1^3 + \beta_{11} x_2^3 + \beta_{11} x_3^3\} \quad (5.13)$$

para β_k serão os coeficientes da aproximação polinomial estimados pelos algoritmos MOLA e MOLAILS, $k = (1, 2, \dots, n_{ensaios})$, x_i ($i = 1, 2$ e 3) são as variáveis de projeto codificadas e Y_{sr} é a resposta obtida pelos algoritmos. Portanto o problema a ser tratado pode ser considerado como um problema para muitos objetivos por ter quatro funções objetivas $Y_{sr} (Y_{sr} = ([C_{dsr}, C_{lsr}, y_{sr} \text{ e } A_{sr}])$.

5.2.3. Experimentos e análise de resultados da Interação Fluido-Estrutura

Os dados de referência obtidos por Lobato (2008) para o modelo do problema Direto da Interação Fluido-Estrutura são apresentados na Tabela 21.

Tabela 21 – Dados do planejamento de entrada para o problema interação fluido-estrutura.

Ensaio	x_1	x_2	x_3	Re	Vr	m'
1	-1	-1	-1	16000	2,5	4
2	1	-1	-1	72000	2,5	4
3	-1	1	-1	16000	14,5	4
4	4	1	-1	72000	14,5	4
5	-1	-1	1	16000	2,5	26
6	1	-1	1	72000	2,5	26
7	-1	1	1	1600	14,5	26
8	1	1	1	72000	14,5	26
9	0	0	0	44000	8,5	15
10	$-\alpha$	0	0	9968,8	8,5	15
11	α	0	0	78031,2	8,5	15
12	0	$-\alpha$	0	44000	1,207	15
13	0	α	0	44000	15,792	15
14	0	0	$-\alpha$	44000	8,5	1,630
15	0	0	α	44000	8,5	28,369

O valor de $\alpha = 1,2154$ foi determinado pela matriz de variância e covariância diagonal, ou seja, pelo planejamento ortogonal que eliminará as correlações entre as variáveis utilizadas no modelo (Lobato, 2008).

A partir dos dados de entrada da Tabela 21, o autor Lobato (2008) alcançou as saídas que podem ser analisadas na Tabela 22. Para realizar a estimativa de parâmetros de β_k , foram utilizados os mesmos dados de entradas de x_i apresentados na Tabela 21.

Tabela 22 – Saídas obtidas pelas simulações via planejamento de experimentos elaborado por Lobato (2008).

Ensaio	C_d	C_l	y	A
1	1,3724	1,0506	0,0476	0,0813
2	1,3887	1,0650	0,0487	0,0801
3	1,0770	0,3197	0,0616	0,3383
4	1,0510	0,3254	0,0632	0,2511
5	1,1949	0,5281	0,0042	0,0086
6	1,1904	0,5213	0,0040	0,0080
7	1,1358	0,4694	0,0168	0,1420
8	1,1310	0,4699	0,0412	0,1877
9	1,0936	0,4237	0,0632	0,1566
10	1,1144	0,4504	0,0550	0,1273
11	1,0885	0,4289	0,0690	0,1557
12	1,2032	0,5155	0,0013	0,0017
13	1,0891	0,4496	0,0398	0,1888
14	1,0050	0,2107	0,2871	0,5360
15	1,1079	0,4715	0,0537	0,0836

As configurações iniciais para a estimativa para o estudo de caso são: população de tamanho é 200, as gerações máximas igual à 1000 e a dimensionalidade do problema é igual ao número de β_k vezes o número de objetivos ($Dim = \beta_k * M \therefore Dim = (13 * 4 = 52)$). Os resultados de simulação são apresentados na Tabela 23.

Tabela 23 – Coeficientes obtidos através da estimação de parâmetros utilizando os algoritmos MOLA e MOLAILS.

MOLA	$C_{d\ est}$	$C_{l\ est}$	y_{est}	A_{est}	MOLAILS	$C_{d\ est}$	$C_{l\ est}$	y_{est}	A_{est}
β_0	-1,8245	-1,2731	-1,6478	-1,6698	β_0	-0,3962	-1,6875	1,2633	-1,5014

β_1	0,0471	1,1051	1,8591	-0,1392	β_1	1,4550	-1,9932	-0,8030	1,1047
β_2	1,1104	1,6153	-1,7650	-0,4617	β_2	-0,4360	-1,2841	-1,2577	0,6637
β_3	-0,1588	0,6351	-0,6822	1,7659	β_3	-0,8882	0,4936	0,4811	-0,5391
β_4	1,2561	-1,2430	0,4577	1,2383	β_4	-1,9906	-1,7696	-1,2313	1,2336
β_5	0,1995	1,8529	1,9278	0,4136	β_5	-1,4264	-1,8765	-0,5131	-1,9709
β_6	-1,5603	-1,8351	0,2582	1,7336	β_6	-0,3328	-1,1332	0,4692	-1,1366
β_7	1,7132	0,2239	1,3882	-1,0251	β_7	-1,4695	-1,8624	-0,3729	-0,7630
β_8	-0,1758	0,6888	-1,2079	-0,3074	β_8	-1,2766	0,4367	0,1733	-0,4247
β_9	0,1402	-1,9720	-1,7119	-1,1790	β_9	-0,6687	-1,2739	1,7223	1,2379
β_{10}	1,8438	0,3750	1,7250	0,1980	β_{10}	-0,5489	-0,8316	1,4237	-1,9981
β_{11}	-1,6875	-0,7117	-0,6568	1,0098	β_{11}	0,5226	-0,8481	0,2107	-0,1385
β_{12}	-0,5399	0,9260	-0,0552	0,7276	β_{12}	1,0879	1,0513	-1,3337	1,3032

Para as variáveis estimadas $\beta_0, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9, \beta_{10}, \beta_{11}$ e β_{12} utilizou-se o seguinte intervalo $[-2, 2]$ resultando nos valores médios apresentados na Tabela 16 para as saídas estimadas utilizando os algoritmos MOLA e MOLAILS. A respeito do desempenho, utilizou-se as métricas MAE, RMSE, RMSLE e ITAE para dos valores médios e desvio padrão disponibilizados na Tabela 24.

Tabela 24 – Saídas obtidas utilizando os algoritmos MOLA e MOLAILS.

MOLA								
Métrica	C_{d_est}		C_{l_est}		y_{est}		A_{est}	
	Med.	Std	Med.	Std	Med.	Std	Med.	Std
MAE	0,6291	0,1998	0,3147	0,1560	0,4658	0,2205	0,4423	0,2142
RMSE	0,6057	0,0307	0,4465	0,2038	0,5820	0,0312	0,7582	0,2128
RMSLE	7,23e-4	0,0003	0,0015	0,0024	0,0117	0,0313	0,0033	0,0060
ITAE	117,49	0,8993	108,84	0,8420	207,24	1,4351	179,11	0,9807
MOLAILS								
Métrica	C_{d_est}		C_{l_est}		y_{est}		A_{est}	
	Med.	Std	Med.	Std	Med.	Std	Med.	Std
MAE	0,6512	0,2141	0,3219	0,1199	0,4111	0,1983	0,4697	0,2109
RMSE	0,0942	0,1447	0,0504	0,1911	0,0616	0,2134	0,0871	0,2287
RMSLE	9,61e-4	0,0022	9,13e-4	0,0017	0,0179	0,0415	0,0018	0,0249
ITAE	50,73	1,6121	20,49	1,2051	35,95	2,5313	26,77	2,8066

Ao analisar os valores médios apresentados na Tabela 24, a métrica MAE na saída C_{l_est} apresenta o menor desvio padrão em relação as demais métricas utilizando a metaheurística MOLAILS. Para a métrica RMSLE na saída C_{d_est} apresenta o menor desvio padrão e menor média para os dois algoritmos propostos. A

métrica ITAE apresenta o maior desvio padrão das soluções de Pareto para as saídas C_{d_est} , γ_{est} e A_{est} .

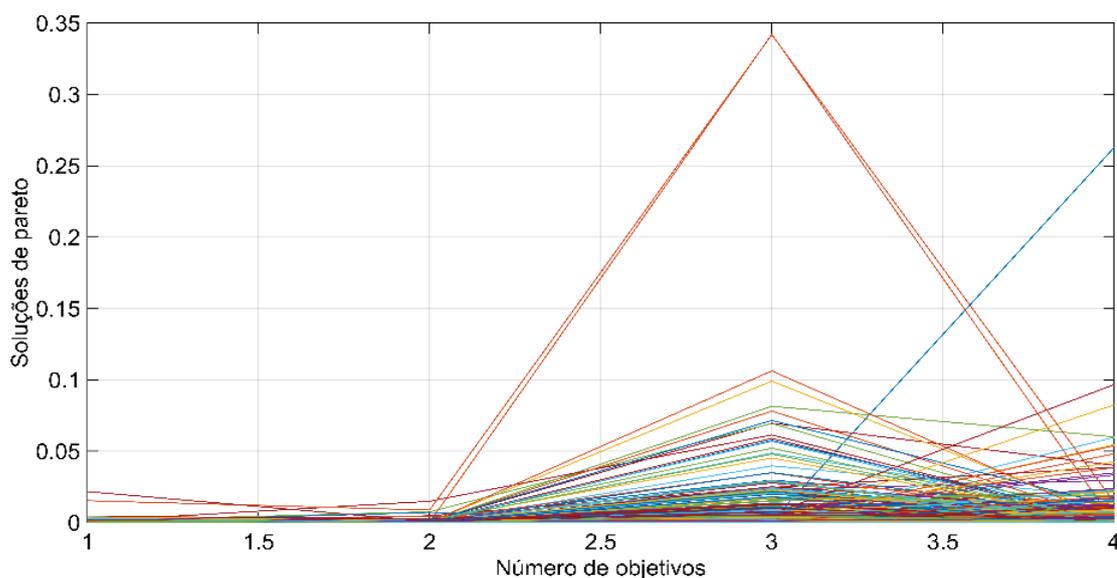
Em relação ao comportamento das soluções de Pareto das quatro funções objetivas, as Figura 46 e 47 demonstram as soluções alcançadas por MOLA e MOLAILS.

A Figura 46 utiliza a métrica RMSLE para a obtenção das soluções de Pareto com MOLA e as coordenadas paralelas com 4 objetivos utilizando MOLAILS são apresentadas na Figura 47.

As distribuições dos conjuntos de soluções foram semelhantes entre MOLA e MOLAILS, comprovando a boa representação das soluções em problemas com muitos objetivos a partir dos dados observados na Tabela 23.

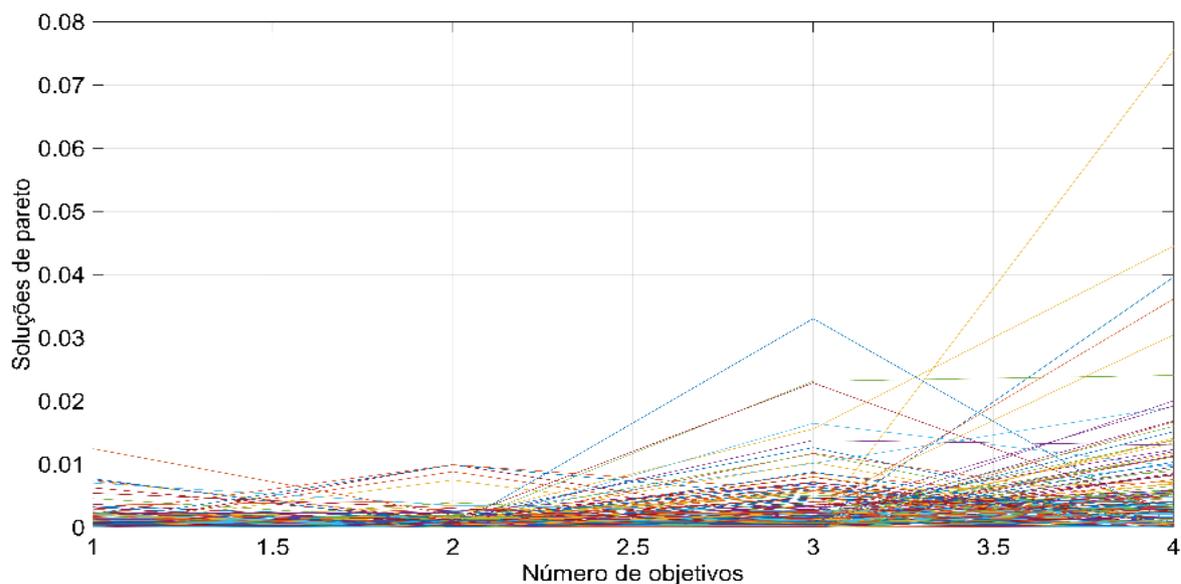
Em questão do tempo computacional para obtenção dos dados estimados como algoritmo MOLAILS, com a métrica MAE o tempo médio é $4,865e+3$ segundos, a métrica RMSE igual à $3,708e+3$ segundos, para a métrica RMSLE é de $5,376e+3$ e para ITAE é $6,164e+4$. Para o algoritmo MOLA, com a métrica MAE o tempo médio é $4,028 e+3$ segundos, a métrica RMSE igual à $3,172e+3$ segundos, para a métrica RMSLE é de $4,556e+3$ e para ITAE é $5,998e+4$.

Figura 46 – Coordenadas paralelas de conjuntos de soluções de Pareto obtidas com o algoritmo MOLA.



Fonte: Autora,2024.

Figura 47 – Coordenadas paralelas de conjuntos de soluções de Pareto obtidas com o algoritmo MOLAILS.



Fonte: Autora,2024.

O tempo computacional se torna elevado devido a dois fatores: o primeiro é o número da dimensão e o número máximo de gerações. Para sistemas reais o tempo computacional se torna uma desvantagem, mas as metaheurísticas MOLA e MOLAILS apresentaram bom desempenho (Tabela 19) ao tratar do problema de estimação de parâmetros para o problema Interação Fluido-Estrutura com Grau de Liberdade.

6. CONCLUSÃO

Este trabalho propôs o algoritmo de leão multiobjetivo que encontra as soluções de Pareto utilizando o critério de menor não-dominância do algoritmo NSGA III e a não-dominância usando a direção das referências normalizadas para se obter os pontos de referências do conjunto de Pareto ótimo. Na fase de fertilização das leas, o MOLA melhora a aptidão das leas através de atualizações e verificação da não dominância entre leas velhas e leas recém-formadas. Isso tende a melhorar as soluções, acelerando a convergência.

Como segunda contribuição, foi proposto o algoritmo cooperativo MOLAILS que melhora a exploração das soluções de Pareto utilizando os algoritmos de otimização de leão e busca local iterativa. Além da troca de informações entre os algoritmos MOLA e ILS foi aplicado a dominância do algoritmo RVEA, o vetor de referência para escolher as soluções de Pareto. Com a troca de informações entre os algoritmos MOLA e ILS, observa-se que foi possível melhorar ou manter as soluções de Pareto quando comparadas com as soluções de Pareto obtidas pelos algoritmos da literatura e do algoritmo proposto MOLA.

No processo de otimização das quinze funções de *benchmark*, o MOLA e MOILS tiveram desempenho semelhantes às técnicas estabelecidas na literatura. O algoritmo MOLA se destaca ao utilizar a métrica IGD com 3 objetivos, onde os valores alcançados foram os menores para a maioria das funções de teste e resultando em 66,67%, ficando em primeiro posto em relação aos demais algoritmos. Para as MaFs com 5 objetivos, o algoritmo MOGWO apresentou 33,33%, mas acaba ficando em segundo posto de melhor algoritmo em relação aos algoritmos por apresentar soluções maiores em pelo menos em 5 MaFs (MaF1, MaF4, MaF6, MaF7 e MaF9). O algoritmo MOLAILS se destaca ao utilizar a métrica HV ficando aproximadamente com 33,33% em primeiro posto, onde os valores altos representam que houve um elevado espalhamento do conjunto de Pareto encontrado por MOLAILS para as quinze MaFs com 3 objetivos e com 5 objetivos.

Ao analisar as Tabelas (6 e 9) pode-se observar que os algoritmos propostos possuem resultados semelhantes aos algoritmos que foram os primeiros colocados nas funções de teste. Isto demonstra que as metaheurísticas propostas neste trabalho se mostraram competitivas em relação as demais metaheurísticas. Além

disso, a metaheurística MOLA se torna atrativa por ter apresentado o menor tempo de processamento em relação as demais metaheurísticas utilizadas no presente trabalho. A respeito da metaheurística proposta MOLAILS, o tempo de processamento fica entre os algoritmos MOLA e NSGA – III, sendo uma metaheurística atrativa quando se trata de agilidade sem prejudicar a busca por soluções de Pareto ótimo.

Os resultados obtidos nas simulações para o problema HTGS pode se observar as saídas simuladas obtêm erros dentro de uma faixa aceitável e quando comparadas as curvas do sinal real (dados em campos) e do sinal estimado (obtidas pelos algoritmos), apresentaram resultados promissores ao sobrepor um sinal no outro. Neste contexto, o algoritmo MOLA apresenta o primeiro posto para as métricas ITAE e RMSLE, resultando valores mínimos quando comparados com os algoritmos MOLAILS e MOGWO. O algoritmo MOGWO ficou em primeiro posto para as métricas de desempenho MAE e RMSE. Por fim, o algoritmo MOLAILS ficou em segundo posto para todas as métricas de desempenho.

Porém uma vantagem da proposta MOLAILS é que, ao analisar o desvio padrão das soluções, apresenta a melhora na dispersão de suas soluções para as métricas RMSE, ITAE e RMSLE quando comparadas com as técnicas MOLA e MOGWO. No teste de validação foi possível observar a robustez dos algoritmos propostos, o MOLA e o MOLAILS, por conseguirem reproduzir a quase sobreposição das saídas reais do modelo com as saídas estimadas. Por fim, MOGWO também obteve uma quase sobreposição das saídas para os dados de validação.

Por fim, os valores obtidos no estudo de caso Interação Fluido-Estrutura com 1 Grau de Liberdade são menores do que os resultados apresentados por Lobato (2008), porém apresentaram comportamentos similares. Sobre uma possível análise das soluções de Pareto encontradas pode-se dizer que o estudo Fluido-Estrutura com 1 grau de liberdade apresenta fronteira de Pareto com propriedades similares de multimodal e convexa (Figuras 46 e 47).

Para trabalhos futuros, os algoritmos propostos MOLA e MOILS poderão ser utilizados como um conjunto de métricas (por exemplo, espaçamento, spread, épsilon e entre outras) para se analisar a exploração e convergência das soluções de Pareto. Além disso, outra proposta futura é de se utilizar os algoritmos propostos como um conjunto de operadores de mutação e cruzamentos para melhorar a busca do conjunto de Pareto a cada geração sem que haja convergência prematura.

REFERÊNCIAS

- ABDELSATTAR, M.; MESALAM, A.; FAWZI, A.; HAMDAN, I. Optimizing Grid-Dependent Hybrid Renewable Energy System with the African Vultures Optimization Algorithm. **SVU-International Journal of Engineering Sciences and Applications**, v. 5, n. 1, p. 89–98, 2024. Egypt's Presidential Specialized Council for Education and Scientific Research.
- ABRAHAM, A.; JAIN, L.; GOLDBERG, R. **Evolutionary Multiobjective Optimization**. Springer, 2005.
- AGUIRRE, L. A. **Introdução à Identificação de Sistemas – Técnicas Lineares e Não-Lineares Aplicadas as Sistemas Reais**. 3ª ed. Belo Horizonte: UFMG, 2015.
- AKBARI, R.; HEDAYATZADEH, R.; ZIARATI, K.; HASSANIZADEH, B. A multi-objective artificial bee colony algorithm. **Swarm and Evolutionary Computation**, v. 2, p. 39–52, 2012. Elsevier B.V.
- ALHIJAWI, B.; AWAJAN, A. Genetic algorithms: theory, genetic operators, solutions, and applications. **Evolutionary Intelligence**, 2023. Springer Science and Business Media Deutschland GmbH.
- AMBEKAR, R. K.; KOLEKAR, U. D. AFL-TOHIP: Adaptive fractional lion optimization to topology-hiding multi-path routing in mobile ad hoc network. **Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017**, p. 727–732, 2017. Institute of Electrical and Electronics Engineers Inc. Acesso em: 22/6/2024.
- BABALIK, A.; OZKIS, A.; UYMAZ, S. A.; KIRAN, M. S. A multi-objective artificial algae algorithm. **Applied Soft Computing Journal**, v. 68, p. 377–395, 2018. Elsevier B.V.
- BABU, B. V.; JEHAN, M. M. L. Differential evolution for multi-objective optimization. **2003 Congress on Evolutionary Computation, CEC 2003 - Proceedings**, v. 4, p. 2696–2703, 2003.
- BANERJEE, A.; PRADHAN, S.; MISRA, B.; CHAKRABORTY, S. A Guide to Meta-Heuristic Algorithms for Multi-objective Optimization: Concepts and Approaches. In: Springer (Org.); **Applied Multi-objective Optimization**. p.1–19, 2024. Singapore.
- BEAUSOLEIL, R. P. “MOSS” multiobjective scatter search applied to non-linear multiple criteria optimization. **European Journal of Operational Research**, v. 169, n. 2, p. 426–449, 2006.

- BERGAMINI, M. G.; MORAES, C. E. O.; OLIVEIRA, A. H. R.; et al. COB-2019-0532 Parameter Estimation of the Hydraulic Amplifier plus Hydraulic Turbine using Metaheuristics. 25th ABCM International Congress of Mechanical Engineering. **Anais...** . p.10, 2019. Uberlândia, MG.
- BERGAMINI, M. G.; OLIVEIRA, G. H. C.; RIBEIRO, E. P.; LEANDRO, G. V. Parameter estimation of the hydraulic turbine regulator system using multi-objective lion algorithm. **Journal of Intelligent and Fuzzy Systems**, v. 45, n. 4, p. 6391–6412, 2023. IOS Press BV.
- BERGAMINI, M. G.; OLIVEIRA, J. L.; OLIVEIRA, G. H. C.; LEANDRO, G. V. Estimação de Parâmetros de Sistemas Não Lineares Utilizando Metaheurísticas Híbridas Paralelizadas. **XIII Brazilian Congress on Computational Intelligence**, p. 11, 2017.
- BHARDWAJ, R.; KUMAR, D. MOFPL: Multi-objective fractional particle lion algorithm for the energy aware routing in the WSN. **Pervasive and Mobile Computing**, v. 58, p. 101029, 2019. Elsevier. Acesso em: 22/6/2024.
- BILAL; PANT, M.; ZAHEER, H.; GARCIA-HERNANDEZ, L.; ABRAHAM, A. Differential Evolution: A review of more than two decades of research. **Engineering Applications of Artificial Intelligence**, v. 90, p. 1–24, 2020. Elsevier Ltd. Disponível em: <<https://doi.org/10.1016/j.engappai.2020.103479>>. .
- BLUM, C.; ROLI, A.; SAMPELS, M. **Hybrid Metaheuristics**. 2013.
- BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information Sciences**, p. 82–117, 2013.
- CHAITRA, T.; AGRAWAL, S.; JIJO, J.; ARYA, A. Multi-Objective Optimization for Dynamic Resource Provisioning in a Multi-Cloud Environment using Lion Optimization Algorithm. **20th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2020 - Proceedings**, p. 83–90, 2020. Institute of Electrical and Electronics Engineers Inc. Acesso em: 22/6/2024.
- CHANDER, S.; VIJAYA, P.; DHYANI, P. Fgv MO-ADDOFL: Multi-objective-based adaptive dynamic directive operative fractional lion algorithm for data clustering. **2017 International Conference on Infocom Technologies and Unmanned Systems: Trends and Future Directions, ICTUS 2017**, v. 2018-January, p. 533–539, 2017. Institute of Electrical and Electronics Engineers Inc. Acesso em: 22/6/2024.
- CHANDER, S.; VIJAYA, P.; DHYANI, P. Fgv MO-ADDOFL: Multi-objective-based adaptive dynamic directive operative fractional lion algorithm for data clustering. **2017**

International Conference on Infocom Technologies and Unmanned Systems: Trends and Future Directions, ICTUS 2017, v. 2018-Janua, p. 533–539, 2018.

CHENG, R.; JIN, Y.; OLHOFFER, M.; SENDHOFF, B. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. **IEEE Transactions on Evolutionary Computation**, v. 20, n. 5, p. 773–791, 2016a. Institute of Electrical and Electronics Engineers Inc.

CHENG, R.; JIN, Y.; OLHOFFER, M.; SENDHOFF, B. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. **IEEE Transactions on Evolutionary Computation**, v. 20, n. 5, p. 773–791, 2016b. Institute of Electrical and Electronics Engineers Inc.

CHENG, R.; LI, M.; TIAN, Y.; et al. Benchmark Functions for CEC'2018 Competition on Many-Objective Optimization. **IEEE Congress on Evolutionary Computation**, p. 1–22, 2018.

CHENG, S.; SHI, Y.; QIN, Q. On the performance metrics of multiobjective optimization. **International Conference in Swarm Intelligence**, p. 504–512, 2012.

CHINTALAPALLI, R. M.; ANANTHULA, V. R. M-LionWhale: Multi-objective optimisation model for secure routing in mobile ad-hoc network. **IET Communications**, v. 12, n. 12, p. 1406–1415, 2018. Institution of Engineering and Technology. Acesso em: 22/6/2024.

COELLO, C. A. C. Evolutionary multi-objective optimization: Some current research trends and topics that remain to be explored. **Frontiers of Computer Science in China**, v. 3 (1), p. 18–30, 2009.

COELLO COELLO, C. A.; LECHUGA, M. S. MOPSO: A proposal for multiple objective particle swarm optimization. **Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002**, v. 2, p. 1051–1056, 2002a.

COELLO COELLO, C. A.; LECHUGA, M. S. MOPSO: A proposal for multiple objective particle swarm optimization. **Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002**, v. 2, p. 1051–1056, 2002b.

COELLO COELLO, C. A.; LECHUGA, M. S. MOPSO: A proposal for multiple objective particle swarm optimization. **Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002**, v. 2, p. 1051–1056, 2002c.

COLLETTE, Y.; SIARRY, P. **Multiobjective optimization: principles and case studies**. Berlin, German: Springer, 2003.

CUERVAS, E.; ZALDIVAR, D.; PEREZ, M. **Advances in Metaheuristics Algorithms_ Methods and Applications**. 2018.

DAS, I.; DENNIS, J. E. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. **SIAM Journal on Optimization**, v. 8, n. 3, p. 631–657, 1998. Society for Industrial and Applied Mathematics Publications.

DEB, K.; AGRAWAL, RAM BHUSAN; AGRAWAL, RAM BHUSHAN. **SIMULATED BINARY CROSSOVER FOR CONTINUOUS SEARCH SPACE**. 1994.

DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. **IEEE Transactions on Evolutionary Computation**, v. 18, n. 4, p. 577–601, 2014. Institute of Electrical and Electronics Engineers Inc.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 2, p. 182–197, 2002.

DEVAGNANAM, J.; ELANGO, N. M. Chronological and exponential-based Lion optimisation for optimal resource allocation in cloud. **International Journal of Grid and Utility Computing**, p. 602–614, 2020. Disponível em: <<https://www.inderscienceonline.com/doi/abs/10.1504/IJGUC.2020.110052>>. Acesso em: 22/6/2024.

DONAIISKY, E.; OLIVEIRA, G. H. C.; SANTOS, E. A. P.; et al. Semi-physical piecewise affine representation for governors in hydropower system generation. **Electric Power Systems Research**, v. 136, p. 181–188, 2016. Elsevier B.V.

FALCÓN-CARDONA, J. G.; COELLO, C. A. C. Indicator-based Multi-objective Evolutionary Algorithms: A Comprehensive Survey. **ACM Computing Surveys**, 1. jun. 2020. Association for Computing Machinery.

FATHY, A.; ABDELAZIZ, A. Y. Single and multi-objective operation management of micro-grid using krill herd optimization and ant lion optimizer algorithms. **International Journal of Energy and Environmental Engineering**, v. 9, n. 3, p. 257–271, 2018. Springer Berlin Heidelberg.

FENG, Z. KAI; NIU, W. JING; LIU, S. Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. **Applied Soft Computing**, v. 98, 2021. Elsevier Ltd.

- FONSECA, C. M.; FLEMING, P. J. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. Proceedings of the Fifth International Conference. **Anais...** , 1993. San Mateo, CA.
- GEIGER, M. J. Randomised Variable Neighbourhood Search for Multi Objective Optimisation. , p. 1–8, 2004.
- GENDREAU, M.; POTVIN, J.-Y. **Handbook of Metaheuristics**. Second ed. Montreal, Canada: Springer, 2010.
- GRID, M.; BELAICHE, L.; KAHLOUL, L.; BENHARZALLAH, S. Parallel dynamic multi-objective optimization evolutionary algorithm. 2021 22nd International Arab Conference on Information Technology, ACIT 2021. **Anais...** , 2021. Institute of Electrical and Electronics Engineers Inc.
- GUPTA, A.; SONI, K. M.; SINGHAL, S. A hybrid metaheuristic and machine learning algorithm for optimal task scheduling in cloud computing. 2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021. **Anais...** , 2021. Institute of Electrical and Electronics Engineers Inc.
- HANSEN, M. P. Tabu Search for Multiobjective Optimization: MOTS. **Proceedings of the Thirteenth International Conference on Multi-Criteria Decision Making (MCDM)**, v. 97, p. 6–10, 1997.
- HASSANZADEH, H. R.; ROUHANI, M. A multi-objective gravitational search algorithm. **2nd International Conference on Computational Intelligence, Communication Systems and Networks**, p. 7–12, 2010.
- HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. E. Niche Pareto genetic algorithm for multiobjective optimization. **IEEE Conference on Evolutionary Computation - Proceedings**, p. 82–87, 1993.
- IEEE. Hydraulic Turbine and Turbine Control Models For System Dynamic Studies. **Transactions on Power Systems**, p. 167–179, 1992. IEEE Working Group on Prime Mover and Energy Supply Models for System Dynamic Performance Studies.
- ISHIBUCHI, H.; MASUDA, H.; TANIGAKI, Y.; NOJIMA, Y. Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems. **IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - MCDM 2014: 2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, Proceedings**, p. 170–177, 2015.

ISHIBUCHI, H.; MURATA, T. A multi-objective genetic local search algorithm and its application to flowshop scheduling. **IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews**, v. 28, n. 3, p. 392–403, 1998.

KATHURIA, M.; GAMBHIR, S. Reliable packet transmission in WBAN with dynamic and optimized QoS using multi-objective lion cooperative hunt optimizer. **Multimedia Tools and Applications**, v. 80, n. 7, p. 10533–10576, 2021. Springer. Disponível em: <<https://link.springer.com/article/10.1007/s11042-020-10144-9>>. Acesso em: 22/6/2024.

KAUR, M.; KADAM, S. A novel multi-objective bacteria foraging optimization algorithm (MOBFOA) for multi-objective scheduling. **Applied Soft Computing Journal**, v. 66, p. 183–195, 2018. Elsevier B.V.

KAVEH, A.; MAHJOUBI, S. Lion pride optimization algorithm: A meta-heuristic method for global optimization problems. **Scientia Iranica**, v. 25, n. Issue 6: Special Issue Dedicated to Professor Goodarz Ahmadi, p. 3113–3132, 2018. Sharif University of Technology. Disponível em: <https://scientiairanica.sharif.edu/article_20833.html>. Acesso em: 22/6/2024.

KAYRI, M.; IPEK, C.; IZCI, D.; EKER, E. SCANM: A Novel Hybrid Metaheuristic Algorithm and its Comparative Performance Assessment. **Electrica**, v. 22, n. 2, p. 143–159, 2022. Istanbul University.

KISENGEU, S. M.; NYAKOE, G. N.; MURIITHI, C. M. **Under Voltage Load Shedding using Hybrid Metaheuristic Algorithms for Voltage Stability Enhancement: A Review**. Nairobi, Kenya: IEEE PES/IAS Power Africa, 2020.

KNOWLES, J.; CORNE, D. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. **Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999**, p. 98–105, 1999.

KUMAR SINGLA, M.; NIJHAWAN, P.; SINGH OBEROI, A. Parameter estimation of proton exchange membrane fuel cell using a novel meta-heuristic algorithm. **Environmental Science and Pollution Research**, v. 28, p. 34511–34526, 2021. Disponível em: <<https://doi.org/10.1007/s11356-021-13097-0>>.

KUMAWAT, I. R.; NANDA, S. J.; MADDILA, R. K. Multi-objective whale optimization. **IEEE Region 10 Annual International Conference, Proceedings/TENCON**, v. 2017-Decem, p. 2747–2752, 2017.

- LEE, S. (SAM); WANG, H. P. (BEN). Modified simulated annealing for multiple-objective engineering design optimization. **Journal of Intelligent Manufacturing**, v. 3, n. 2, p. 101–108, 1992.
- LI, L.; CHAI, Z.; LI, Y.; CHENG, Y.; NIE, Y. Many-objective multi-tasking optimization using adaptive differential evolutionary and reference-point based nondominated sorting. **Expert Systems with Applications**, v. 248, 2024. Elsevier Ltd.
- LIU, H.; ABRAHAM, A.; GROSAN, C. A novel variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems. **2007 2nd International Conference on Digital Information Management, ICDIM**, v. 1, p. 138–145, 2007.
- LOBATO, F. S. **Otimização Multi-Objetivo para o Projeto de Sistemas de Engenharia**, 2008. Uberlândia: Universidade federal de Uberlândia. Disponível em: <<https://repositorio.ufu.br/handle/123456789/14677>>. Acesso em: 6/5/2024.
- VON LÜCKEN, C.; BARÁN, B.; BRIZUELA, C. A survey on multi-objective evolutionary algorithms for many-objective problems. **Computational Optimization and Applications**, v. 58, n. 3, p. 707–756, 2014a. Kluwer Academic Publishers.
- VON LÜCKEN, C.; BARÁN, B.; BRIZUELA, C. A survey on multi-objective evolutionary algorithms for many-objective problems. **Computational Optimization and Applications**, v. 58, n. 3, p. 707–756, 2014b.
- MARROUCHE, W.; CHLEBIKOVA, J. **Unlocking the Potential of Metaheuristics for NP-Hard Problems**. 2024.
- MIRJALILI, S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. **Neural Computing and Applications**, v. 27, n. 4, p. 1053–1073, 2016. Springer London.
- MIRJALILI, S.; GANDOMI, A. H.; MIRJALILI, S. Z.; et al. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. **Advances in Engineering Software**, v. 114, p. 163–191, 2017. Elsevier Ltd.
- MIRJALILI, S.; JANGIR, P.; SAREMI, S. Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. **Applied Intelligence**, v. 46, n. 1, p. 79–95, 2017. Applied Intelligence.
- MIRJALILI, S.; SAREMI, S.; MIRJALILI, S. M.; COELHO, L. D. S. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. **Expert Systems with Applications**, v. 47, p. 106–119, 2016. Elsevier Ltd.

- MORIYAMA, Y.; IIMURA, I.; NAKAYAMA, S. Consideration on Isolation Strategy for Multi-Objective Quantum-Inspired Evolutionary Algorithm. 2021 IEEE 12th International Workshop on Computational Intelligence and Applications, IWCI 2021 - Proceedings. **Anais...**, 2021. Institute of Electrical and Electronics Engineers Inc.
- MOSSER, A. A. **Group territoriality of the African lion: behavioral adaptation in a heterogeneous landscape**. Minnesota, EUA, 2008.
- NAGHAVIPOUR, H.; SOON, T. K.; IDRIS, M. Y. I. BIN; et al. Hybrid Metaheuristics for QoS-Aware Service Composition: A Systematic Mapping Study. **IEEE Access**, v. 10, p. 12678–12701, 2022. Institute of Electrical and Electronics Engineers Inc.
- NASSEF, A. M.; ABDELKAREEM, M. A.; MAGHRABIE, H. M.; BAROUTAJI, A. Review of Metaheuristic Optimization Algorithms for Power Systems Problems. **Sustainability (Switzerland)**, 1. jun. 2023. Multidisciplinary Digital Publishing Institute (MDPI).
- O'HAGAN, S.; KNOWLES, J.; KELL, D. B.; et al. DEMO: Differential Evolution for Multiobjective Optimization. **American Association for Artificial Intelligence**, v. 38, n. March, p. 9–40, 2000.
- OSTERTAGOVÁ, E.; OSTERTAG, O.; KOVÁČ, J. Methodology and application of the Kruskal-Wallis test. **Applied Mechanics and Materials**, v. 611, p. 115–120, 2014. Trans Tech Publications Ltd.
- ÖZKIŞ, A.; BABALIK, A. A novel metaheuristic for multi-objective optimization problems: The multi-objective vortex search algorithm. **Information Sciences**, v. 402, p. 124–148, 2017.
- PALAKONDA, V.; MALLIPEDDI, R. Pareto Dominance-Based Algorithms with Ranking Methods for Many-Objective Optimization. **IEEE Access**, v. 5, n. October, p. 11043–11053, 2017.
- PETCHROMPO, S.; COIT, D. W.; BRINTRUP, A.; WANNAKRAIROT, A.; PARLIKAD, A. K. A review of Pareto pruning methods for multi-objective optimization. **Computers and Industrial Engineering**, v. 167, 2022. Elsevier Ltd.
- PIEREZAN, J.; COELHO, S.; MARIANI, V. C.; LEBENSZTAJN, L. Multiobjective Coyote Algorithm Applied to Electromagnetic Optimization. **2019 22nd International Conference on the Computation of Electromagnetic Fields (COMPUMAG)**, p. 1–4, 2019. IEEE.

- POTTER, M. A.; DE JONG, K. A. Cooperative coevolution: an architecture for evolving coadapted subcomponents. **Evolutionary computation**, v. 8, n. 1, p. 1–29, 2000.
- PRADHAN, P. M.; PANDA, G. Solving multiobjective problems using cat swarm optimization. **Expert Systems with Applications**, v. 39, n. 3, p. 2956–2964, 2012. Elsevier Ltd.
- QASEM, S. N.; SHAMSUDDIN, S. M.; ZAIN, A. M. Multi-objective hybrid evolutionary algorithms for radial basis function neural network design. **Knowledge-Based Systems**, v. 27, p. 475–497, 2012. Elsevier B.V.
- QIN, H.; LI, K.; LIU, Y.; LU, C. Multi-objective Optimal Configuration of Electric-Hydrogen Hybrid Energy Storage System. 2022 IEEE 7th International Conference on Power and Renewable Energy, ICPRE 2022. **Anais...** . p.1056–1062, 2022. Institute of Electrical and Electronics Engineers Inc.
- RAJAKUMAR, B. R. The Lion’s Algorithm: A New Nature-Inspired Search Algorithm. **Procedia Technology**, v. 6, p. 126–135, 2012.
- RAJAKUMAR, B. R. **Chapter 5: Lion Algorithm and Its Applications**. Singapore, 2020.
- SADOUKI, S. C.; TARI, A. Multi-objective and discrete Elephants Herding Optimization algorithm for QoS aware web service composition. **RAIRO - Operations Research**, v. 53, n. 2, p. 445–459, 2019.
- SCHAFFER, J. D. Multiple objective optimization with vector evaluated genetic algorithms. **The 1st international Conference on Genetic Algorithms**, , n. JANUARY 1985, p. 93–100, 1985.
- SCHULZ, S.; NEUFELD, J. S.; BUSCHER, U. A multi-objective iterated local search algorithm for comprehensive energy-aware hybrid flow shop scheduling. **Journal of Cleaner Production**, v. 224, p. 421–434, 2019. Elsevier Ltd.
- SHI, X.; KONG, D. A multi-objective ant colony optimization algorithm based on elitist selection strategy. **Metallurgical and Mining Industry**, v. 7, n. 6, p. 333–338, 2015.
- SILVA, A. R. DA. **ALICE ROSA DA SILVA MODELAGEM MATEMÁTICA DE INTERAÇÃO FLUIDO-ESTRUTURA UTILIZANDO O MÉTODO DA FRONTEIRA IMERSA UNIVERSIDADE FEDERAL DE UBERLÂNDIA FACULDADE DE ENGENHARIA MECÂNICA**, 2008. Uberlândia: Universidade Federal de Uberlândia. Disponível em: <<https://repositorio.ufu.br/handle/123456789/14678?mode=full>>. Acesso em: 5/5/2024.

SINGH RAGHUWANSHI, G.; SINHA, P.; NARAYAN THAKRE, J.; PROFESSOR, A. **Solving Environmental Economic Load Dispatch with Line Losses in Power System using Lion Pride Optimization Algorithm**. 2017.

SIVASUBRAMANI, S.; SWARUP, K. S. Multi-objective harmony search algorithm for optimal power flow problem. **International Journal of Electrical Power and Energy Systems**, v. 33, n. 3, p. 745–752, 2011. Elsevier Ltd.

SOUZA, D. K. **Estimação de Parâmetros por Busca de Cuco via Voos de Lévy**, 2014. Universidade Federal do Paraná.

SUN, X.; GUO, S.; GUO, J.; DU, B. A Hybrid Multi-Objective Evolutionary Algorithm with Heuristic Adjustment Strategies and Variable Neighbor-Hood Search for Flexible Job-Shop Scheduling Problem Considering Flexible Rest Time. **IEEE Access**, v. 7, p. 157003–157018, 2019. IEEE.

TALBI, E. G. A taxonomy of hybrid metaheuristics. **Journal of Heuristics**, v. 8, n. 5, p. 541–564, 2002.

TALBI, E. G. Hybrid metaheuristics for multi-objective optimization. **Journal of Algorithms and Computational Technology**, v. 9, n. 1, p. 41–63, 2015.

TEZEL, B. T.; MERT, A. A cooperative system for metaheuristic algorithms. **Expert Systems with Applications**, v. 165, 2021. Elsevier Ltd.

TOMAR, V.; BANSAL, M.; SINGH, P. Metaheuristic Algorithms for Optimization: A Brief Review. RAiSE-2023. **Anais...** . p.238, 2024. Basel Switzerland: MDPI. Disponível em: <<https://www.mdpi.com/2673-4591/59/1/238>>. .

VIANNA, D. S.; ARROYO, J. E. C. A GRASP algorithm for the multi-objective knapsack problem. **Proceedings - International Conference of the Chilean Computer Science Society, SCCC**, p. 69–75, 2004.

WANG, B.; JIN, X. P.; CHENG, B. Lion pride optimizer: An optimization algorithm inspired by lion pride behavior. **Science China Information Sciences**, v. 55, n. 10, p. 2369–2389, 2012.

WANG, J.; DU, P.; NIU, T.; YANG, W. A novel hybrid system based on a new proposed algorithm—Multi-Objective Whale Optimization Algorithm for wind speed forecasting. **Applied Energy**, v. 208, n. October, p. 344–360, 2017. Elsevier.

WOLPERT, D. V.; MACREADY, WILLIAM, G.; No Free Lunch Theorems for Optimization. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 1, p. 259–264, 1997.

WONG, W.; MING, C. I. A Review on Metaheuristic Algorithms: Recent Trends, Benchmarking and Applications. **7th International Conference on Smart Computing and Communications, ICSCC 2019**, p. 1–5, 2019. IEEE.

WU, G.; LUO, Q.; DU, X.; et al. Ensemble of Metaheuristic and Exact Algorithm Based on the Divide-And-Conquer Framework for Multisatellite Observation Scheduling. **IEEE Transactions on Aerospace and Electronic Systems**, v. 58, n. 5, p. 4396–4408, 2022. Institute of Electrical and Electronics Engineers Inc.

XU, Y.; XU, C.; ZHANG, H.; et al. A Multi-Population Multi-Objective Evolutionary Algorithm Based on the Contribution of Decision Variables to Objectives for Large-Scale Multi/Many-Objective Optimization. **IEEE Transactions on Cybernetics**, v. 53, n. 11, p. 6998–7007, 2023. Institute of Electrical and Electronics Engineers Inc.

YADAV, D.; RAMU, P.; DEB, K. Finding Robust Solutions for Many-Objective Optimization Using NSGA-III. 2023 IEEE Congress on Evolutionary Computation, CEC 2023. **Anais...**, 2023. Institute of Electrical and Electronics Engineers Inc.

YANG, L.; ZHANG, Y.; CAO, J.; LI, K.; WANG, D. A many-objective evolutionary algorithm based on reference vector guided selection and two diversity and convergence enhancement strategies. **Applied Soft Computing**, v. 154, p. 111369, 2024. Disponible em:

<<https://linkinghub.elsevier.com/retrieve/pii/S1568494624001431>>. .

YANG, X. S.; DEB, S. Multiobjective cuckoo search for design optimization. **Computers and Operations Research**, v. 40, n. 6, p. 1616–1624, 2013. Elsevier.

YAZDANI, M.; JOLAI, F. Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm. **Journal of Computational Design and Engineering**, v. 3, n. 1, p. 24–36, 2016. Elsevier. Disponible em: <<http://dx.doi.org/10.1016/j.jcde.2015.06.003>>. .

ZENG, H.; ZHU, J.; LI, Y.; YING, W. MOMS-HDEA: A Multi-Objective Multi-State Hybrid Differential Evolution Algorithm for system reliability optimization design problems. **International Conference on Computer and Communications Security, ICCCS 2009**, p. 92–95, 2009.

ZHANG, Q.; LI, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. **IEEE Transactions on Evolutionary Computation**, v. 11, n. 6, p. 712–731, 2007a.

ZHANG, Q.; LI, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. **IEEE Transactions on Evolutionary Computation**, v. 11, n. 6, p. 712–731, 2007b.

ZHANG, Y.; WANG, G. G.; LI, K.; et al. Enhancing MOEA/D with information feedback models for large-scale many-objective optimization. **Information Sciences**, v. 522, p. 1–16, 2020. Elsevier Inc.

ZITZLER, E. **Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications** Institut für Technische Informatik und Kommunikationsnetze Computer Engineering and Networks Laboratory, 1999. Swiss Federal Institute of Technology Zurich.

ZITZLER, E.; KÜNZLI, S. Indicator-based selection in multiobjective search. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 3242, p. 832–842, 2004. Springer Verlag.

ZITZLER, E.; LAUMANN, M.; THIELE, L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Research Collection. **Anais...** p.12–19, 2001. Zurich, Suíça.