



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Evolução Diferencial Bidirecional de Ordenamento por Não-dominância com Predição no Espaço Objetivo e de Decisão

Cicero Samuel Rodrigues Mendes



Recife

2025

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Cicero Samuel Rodrigues Mendes

**Evolução Diferencial Bidirecional de Ordenamento por
Não-dominância com Predição no Espaço Objetivo e de
Decisão**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Aluizio Fausto Ribeiro Araújo

Recife

2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Mendes, Cicero Samuel Rodrigues.

Evolução diferencial bidirecional de ordenamento por não-dominância com predição no espaço objetivo e de decisão / Cicero Samuel Rodrigues Mendes. - Recife, 2025.

95f.: il.

Dissertação (Mestrado) - Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-Graduação em Ciência da Computação, 2024.

Orientação: Aluizio Fausto Ribeiro Araújo.

Inclui referências.

1. Problemas multiobjetivo com restrições; 2. Algoritmos evolucionários; 3. Evolução diferencial; 4. Predição no espaço de busca; 5. Otimização multiobjetivo; 6. Controle de parâmetros. I. Araújo, Aluizio Fausto Ribeiro. II. Título.

UFPE-Biblioteca Central

Cicero Samuel Rodrigues Mendes

**“Evolução Diferencial Bidirecional de Ordenamento por Não-dominância
com Predição no Espaço Objetivo e de Decisão”**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Inteligência Computacional.

Aprovado em: 28/10/2024.

BANCA EXAMINADORA

Prof. Dr. Aluizio Fausto Ribeiro Araújo
Centro de Informática / UFPE
(orientador)

Prof. Dr. André Luiz Vizine Pereira
Instituto de Mar / UNIFESP

Prof. Dr. Fernando Buarque de Lima Neto
Escola Politécnica de Pernambuco / UPE

Dedico este trabalho à minha família, pelo apoio incondicional em todos os momentos. Em especial, dedico à memória da minha mãe, que foi uma das minhas maiores incentivadoras e cuja força continua sendo uma inspiração constante em meu caminho.

Agradecimentos

Primeiramente, expresso a minha profunda gratidão à minha família pelo apoio incondicional ao longo de toda a minha jornada. Destaco especialmente meu pai, José, meu irmão, Sawyo e minhas irmãs, Samara e Maria do Socorro, cuja presença foi essencial em cada etapa. Em particular, agradeço a Maria do Socorro, que, com sua força e dedicação, me encorajou a seguir em frente mesmo nos momentos mais difíceis.

Agradeço também aos meus avós, primos e demais familiares pelo incentivo e apoio de sempre, que me deram forças para continuar em busca dos meus objetivos.

Expresso também a minha gratidão às minhas amigas Claésia, Bruna e Fernanda, pela amizade sincera e pelo companheirismo, que tornaram essa trajetória mais leve e especial.

Um agradecimento especial ao meu ex-professor e amigo, Guilherme Esmeraldo, pela amizade, companheirismo e pelo incentivo em cursar o mestrado. Sua confiança em mim foi um impulso significativo nessa jornada.

Agradeço profundamente ao meu orientador, Aluízio Araújo, pela paciência, orientação e parceria ao longo dessa jornada acadêmica. Sua dedicação, conselhos e disponibilidade foram essenciais para a conclusão deste trabalho.

If I have seen further, it is by standing upon the shoulders of giants.
– Sir Isaac Newton

Resumo

Problemas multiobjetivo com restrições (CMOPs) são amplamente encontrados em diversas aplicações científicas e de engenharia. Algoritmos Evolucionários (EAs) são empregados para resolver estes problemas devido a sua forte habilidade de busca. Durante o processo evolucionário, EAs geram um grande número de soluções candidatas que são pouco exploradas. Técnicas de aprendizado de máquina podem ser utilizadas para analisar e extrair conhecimento útil destas soluções para guiar e melhorar a capacidade de busca, o que implica em um algoritmo inteligente e bem informado. Este trabalho propõe uma nova abordagem para lidar com CMOPs, o P-NSBiDiCo, que emprega previsão para identificar tendências evolucionárias e gerar novas soluções em áreas promissoras preditas no espaço de busca. A abordagem proposta introduz novas estratégias para selecionar os parâmetros F e Cr da evolução diferencial (DE), bem como estratégia de previsão nos espaços objetivo e de decisão. Resultados experimentais em 46 funções de *benchmark* e 23 problemas do mundo real sugerem que o P-NSBiDiCo obteve resultados competitivos na resolução de problemas práticos complexos quando comparados a sete algoritmos bem estabelecidos no estado da arte.

Palavras-chave: Problemas Multiobjetivo com Restrições, Algoritmos Evolucionários, Evolução Diferencial, Previsão no Espaço de Busca, Otimização Multiobjetivo, Controle de Parâmetros.

Abstract

Constrained multi-objective problems (CMOPs) are widely found in a number of scientific and engineering applications. Evolutionary algorithms (EAs) are used to solve these problems due to their strong search capabilities. During the evolutionary process, EAs generate a large number of candidate solutions that are poorly explored. Machine learning techniques can be used to analyze and extract useful knowledge from these solutions to guide and improve the search capability, which implies in an intelligent and well-informed algorithm. This study proposes a new approach for handling CMOPs, P-NSBiDiCo, which employs prediction to identify evolutionary trends and generate new solutions in predicted promising areas in the search space. The proposed approach introduces new strategies for the selection of the F and Cr parameters of differential evolution (DE), as well as prediction strategies in the objective and decision spaces. Experimental results on 46 benchmark functions and 23 real-world problems suggest that P-NSBiDiCo achieved competitive results in solving complex practical problems when compared to seven well-established state-of-the-art algorithms.

Keywords: Constrained Multi-objective Problems, Evolutionary Algorithms, Differential Evolution, Search Space Prediction, Multi-objective Optimization, Parameter Control.

Lista de ilustrações

| | |
|---|----|
| Figura 1 – Ilustração dos Tipos de CMOPs. | 22 |
| Figura 2 – Fluxo de Execução do Algoritmo P-NSBiDiCo. | 41 |
| Figura 3 – Fluxograma para a seleção de soluções inviáveis não dominadas. . . | 44 |
| Figura 4 – Curvas de mudança da média μ ao longo das gerações para diferentes valores de λ | 50 |
| Figura 5 – Fluxograma do pré-processamento de dados para o treinamento dos modelos NARX. | 52 |
| Figura 6 – Conjunto de treinamento, previsões e população observada no espaço objetivo para DASCMP4 | 55 |
| Figura 7 – Conjunto de treinamento, previsões e população observada no espaço de decisão para DOC1 | 56 |
| Figura 8 – Fluxograma para seleção de soluções promissoras preditas e a geração de novas soluções tentativas. | 58 |
| Figura 9 – Ilustração dos LIRCMOPs. | 63 |
| Figura 10 – Perfil de convergência (IGD e HV) dos algoritmos para os LIRCMOPs. | 66 |
| Figura 11 – Ilustração dos DASCMPs. | 67 |
| Figura 12 – Gráficos de Convergência dos DASCMPs. | 70 |
| Figura 13 – Ilustração das funções de <i>benchmark</i> da base DOC. | 71 |
| Figura 14 – Gráficos de Convergência dos DOC. | 74 |
| Figura 15 – Ilustração dos MWs. | 75 |
| Figura 16 – Gráficos de Convergência dos MW. | 77 |
| Figura 17 – Convergência média dos modelos para o conjunto de problemas do mundo real - RWMOPs. | 80 |
| Figura 18 – Comparação da Convergência Média entre as Variações dos Modelo – NSBiDiCo-V1 vs NSBiDiCo-V3 para LIRCMOP. | 82 |
| Figura 19 – Comparação da Convergência Média entre as Variações dos Modelo – NSBiDiCo-V1 vs NSBiDiCo-V3 para DASCMP. | 84 |
| Figura 20 – Comparação da Convergência Média entre as Variações dos Modelo – NSBiDiCo-V1 vs NSBiDiCo-V3 para DOC. | 85 |
| Figura 21 – Comparação da Convergência Média entre as Variações dos Modelo – NSBiDiCo-V1 vs NSBiDiCo-V3 para MW. | 86 |

Lista de tabelas

| | |
|--|----|
| Tabela 1 – Média e desvio padrão de IGD para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas LIRCMOP. | 65 |
| Tabela 2 – Média e desvio padrão de HV para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas LIRCMOP. | 65 |
| Tabela 3 – Média e desvio padrão de IGD para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas DASCMP. | 69 |
| Tabela 4 – Média e desvio padrão de HV para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas DASCMP. | 69 |
| Tabela 5 – Média e desvio padrão de IGD para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas DOC. | 73 |
| Tabela 6 – Média e desvio padrão de HV para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas DOC. | 73 |
| Tabela 7 – Média e desvio padrão de IGD para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas MW. | 76 |
| Tabela 8 – Média e desvio padrão de HV para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas MW. | 78 |
| Tabela 9 – Média e desvio padrão de HV para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas do mundo real. | 79 |
| Tabela 10 – Comparação do Desempenho do NSBiDiCo-V1 e NSBiDiCo-V2 em Diversos Problemas de <i>Benchmark</i> Multiobjetivo com Restrições. | 83 |
| Tabela 11 – Comparação do Desempenho do NSBiDiCo-V2 e NSBiDiCo-V3 em Diversos Problemas de <i>Benchmark</i> Multiobjetivo com Restrições. | 87 |

Lista de algoritmos

| | | |
|---|---|----|
| 1 | <i>Framework</i> do P-NSBiDiCo | 43 |
| 2 | Atualização da População do Arquivo | 44 |
| 3 | Conjunto de Cruzamento Restrito | 46 |
| 4 | Seleção Ambiental | 59 |

Lista de abreviaturas e siglas

| | |
|-------|--|
| CDP | Princípio de dominância de restrições (do inglês, <i>Constraint dominance principle</i>). |
| CHT | Técnica de tratamento de restrições (do inglês, <i>Constraint Handling Technique</i>). |
| CMOEA | Algoritmo evolucionário multiobjetivo com restrições (do inglês, <i>Constrained multiobjective Evolutionary Algorithm</i>). |
| CMOP | Problema de otimização multiobjetivo com restrições (do inglês, <i>Constrained multiobjective optimization problems</i>). |
| CPF | Fronteira de Pareto restrita (do inglês, <i>Constrained Pareto front</i>). |
| DE | Evolução Diferencial (do inglês, <i>Differential Evolution</i>). |
| EA | Algoritmo evolucionário (do inglês, <i>Evolutionary Algorithm</i>). |
| EdD | Espaço de decisão. |
| EO | Espaço objetivo. |
| GP | Processo Gaussiano (do inglês, <i>Gaussian Process</i>). |
| HV | Hipervolume. |
| IGD | Distância geracional invertida (do inglês, <i>Inverted generational distance</i>). |
| IoT | Internet das Coisas (do inglês, <i>Internet of Things</i>). |
| ML | Aprendizagem de máquina (do inglês, <i>Machine Learning</i>). |
| MOP | Problema de otimização multiobjetivo (do inglês, <i>multiobjective optimization problems</i>). |
| NARX | Modelo exógeno autoregressivo não linear (do inglês, <i>Nonlinear autoregressive exogenous model</i>). |
| NSDE | Evolução diferencial de ordenação por não-dominância (do inglês, <i>Nondominated sorting differential evolution</i>). |
| SR | Ranking estocástico (do inglês, <i>Stochastic ranking</i>). |
| UPF | Fronteira de Pareto irrestrita (do inglês, <i>Unconstrained Pareto front</i>). |

Sumário

| | | |
|----------|---|-----------|
| 1 | Introdução | 15 |
| 1.1 | Motivação | 16 |
| 1.2 | Objetivos | 17 |
| 1.3 | Contribuições | 17 |
| 1.4 | Estrutura do Documento | 17 |
| 2 | Problemas multiobjetivos com Restrições: Definições e Abordagens | 19 |
| 2.1 | CMOPs: Definição Formal | 19 |
| 2.1.1 | Definições Básicas | 20 |
| 2.1.2 | Tipos de CMOPs | 21 |
| 2.1.3 | Tipos de Dificuldades de Otimização de CMOPs | 21 |
| 2.2 | Algoritmos Evolucionários para CMOPs | 24 |
| 2.2.1 | Métodos Baseados em Função de Penalidade | 24 |
| 2.2.2 | Métodos Baseados na Separação dos Objetivos e Restrições | 25 |
| 2.2.3 | Métodos Baseados na Definição de Problema multiobjetivo | 25 |
| 2.2.4 | Métodos Baseados em Co-evolução e Multi-estágios | 26 |
| 2.2.5 | Métodos Híbridos | 32 |
| 2.3 | Métricas de Desempenho | 33 |
| 2.3.1 | Distância Geracional Invertida (IGD) | 34 |
| 2.3.2 | Hiper-volume (HV) | 34 |
| 2.4 | Sucessos e Limitações das Soluções de CMOPs por EAs | 35 |
| 3 | Evolução Diferencial Bidirecional de Ordenamento por Não-dominância com Predição no Espaço Objetivo e de Decisão | 37 |
| 3.1 | Requisitos de Software | 38 |
| 3.1.1 | Requisitos Funcionais | 38 |
| 3.1.2 | Requisitos Não Funcionais | 40 |
| 3.2 | P-NSBiDiCo | 40 |
| 3.3 | Determinação das Soluções Inviáveis Não-dominadas | 44 |
| 3.4 | Seleção Baseada em Ângulo | 45 |
| 3.5 | Algoritmo de Busca | 47 |
| 3.6 | Amostragem dos Parâmetros F e Cr | 48 |
| 3.6.1 | Amostragem de F | 48 |
| 3.6.2 | Amostragem de Cr | 49 |
| 3.7 | Coleta e Pré-processamento de Dados | 51 |
| 3.8 | Predição com <i>Non-linear AutoRegressive eXogenous</i> (NARX) | 52 |

| | | |
|----------|---|-----------|
| 3.9 | Exemplos de Funcionamento da Predição | 54 |
| 3.10 | Seleção de Soluções Promissoras Preditadas | 55 |
| 3.11 | Seleção Ambiental | 58 |
| 3.12 | Considerações do Capítulo | 59 |
| 4 | Validação e Testes da Abordagem Proposta | 60 |
| 4.1 | Configuração dos Experimentos | 60 |
| 4.2 | Resultados Experimentais para os <i>Benchmarks</i> | 62 |
| 4.2.1 | LIRCMOP | 62 |
| 4.2.2 | DASCMOP | 67 |
| 4.2.3 | DOC | 69 |
| 4.2.4 | MW | 73 |
| 4.3 | Resultados Experimentais em Problemas do Mundo Real | 78 |
| 4.4 | Estudo de <i>Ablation</i> | 79 |
| 4.4.1 | Variações do Modelo | 79 |
| 4.4.2 | Análise das Variações dos Modelos | 81 |
| 4.5 | Considerações do Capítulo | 81 |
| 5 | Conclusão | 88 |
| 5.1 | Considerações | 89 |
| 5.1.1 | Pontos Fortes do P-NSBiDiCo | 89 |
| 5.1.2 | Limitações | 89 |
| 5.1.3 | Trabalhos Futuros | 90 |
| 5.2 | Publicação | 90 |
| 5.2.1 | Conferência Internacional – <i>Qualis A2</i> | 90 |
| | Referências | 91 |

1

Introdução

Problemas de otimização multiobjetivo com restrições (CMOPs, do inglês *Constrained Multiobjective Optimization Problems*) (Liang et al., 2023) são amplamente encontrados em diversas aplicações práticas do mundo real, especialmente nas áreas de ciência e engenharia. Exemplos incluem roteirização de veículos (Jozefowicz; Semet; Talbi, 2008), projeto de sistemas de abastecimento de água (Mala-Jetmarova; Sultanova; Savic, 2017) e otimização de sistemas de armazenamento de energia (Farzin; Fotuhi-Firuzabad; Moeini-Aghaie, 2017). Esses problemas caracterizam-se pela presença de múltiplos objetivos conflitantes, que devem ser otimizados simultaneamente, e por um conjunto de restrições que precisam ser satisfeitas para garantir a viabilidade das soluções (Liang et al., 2023).

A resolução dos CMOPs é especialmente desafiadora devido à complexidade de manter um equilíbrio entre a otimização dos diferentes objetivos e a satisfação das restrições. Algoritmos evolucionários multiobjetivo com restrições (CMOEA, do inglês *Constrained Multiobjective Evolutionary Algorithms*) têm sido amplamente utilizados para tratar esse tipo de problema, devido à sua robustez e capacidade de explorar o espaço de busca (Liang et al., 2023). No entanto, o grande número de soluções inviáveis que surgem durante o processo de otimização é um dos principais obstáculos enfrentados por esses algoritmos. Técnicas especializadas de tratamento de restrições (CHTs, do inglês *Constraint Handling Technique*) são, portanto, de suma importância para guiar a busca e garantir que as soluções geradas atendam às restrições impostas pelo problema (Zhan et al., 2022).

Diversas maneiras de aprimorar os CMOEAs tem sido investigadas pela comunidade científica, onde têm-se proposto diversas estratégias (Liang et al., 2023), como o uso de co-evolução (Ma et al., 2019), a hibridização com técnicas, incluindo o aprendizado de máquina e o desenvolvimento de novos operadores evolucionários. O aprendizado de máquina, em particular, tem se mostrado uma abordagem promissora para melhorar a eficiência e a eficácia dos CMOEAs, visto que pode ser utilizado para

prever o comportamento da população ao longo das gerações e identificar regiões de busca mais promissoras.

Este trabalho apresenta uma nova abordagem, denominada Evolução Diferencial Bidirecional de Ordenamento por Não-dominância com Predição no Espaço Objetivo e de Decisão (P-NSBiDiCo), que busca tratar problemas de otimização multiobjetivo com restrições. A proposta avança o modelo NSBiDiCo (Mendes; Araújo; Farias, 2023), através da integração de um mecanismo de predições nos espaços objetivo e de decisão, além de introduzir novas estratégias para a seleção dos parâmetros da evolução diferencial. Este estudo visa o desenvolvimento de um algoritmo capaz de lidar de forma eficiente com problemas complexos do mundo real, fornecendo soluções de alta qualidade.

1.1 Motivação

Durante o processo de otimização os algoritmos evolucionários (EAs, do inglês *Evolutionary Algorithms*) geram grandes volumes de informações que, em geral, não são exploradas de forma eficaz para realizar uma busca bem informada. Tendo isso em vista, bem como nos recentes avanços em aprendizado de máquina e inteligência artificial, é que surge a necessidade de se explorar o uso de técnicas de aprendizagem de máquina (ML, do inglês *Machine Learning*) com as informações geradas durante a evolução para projetar novas metodologias para resolver problemas multiobjetivo com restrições.

Para Zhan et al. (2022), os algoritmos evolucionários têm sido aplicados com sucesso em diversos problemas de otimização global. No entanto, novas dificuldades surgem a medida que o desenvolvimento de novas tecnologias como Internet das Coisas (IoT, do inglês *Internet of Things*), computação em nuvem e *big data* tornam os problemas de otimização cada vez mais complexos.

Talbi (2021) argumenta que a aprendizagem de máquina pode ser útil para analisar os dados gerados e extrair conhecimentos que podem guiar e melhorar o desempenho de busca do algoritmo para que este possa ser mais inteligente e bem-informado. Ainda para Talbi (2021), a incorporação de ML em algoritmos de meta-heurística tem se provado ser vantajoso em velocidade de convergência, qualidade das soluções e robustez. Dessa forma, a pesquisa em projeto de meta-heurísticas com o emprego de ML terão impacto importante no avanço do estado da arte.

Um dos estudos de revisão mais recentes sobre EAs para CMOPs apresentado por Liang et al. (2023), ressalta, como um potencial tópico de pesquisa futura, a evolução guiada por tipo de problema. Os autores argumentam que as relações entre fronteira de Pareto restrita (CPF, do inglês *Constrained Pareto front*) e irrestrita (UPF, do inglês *Un-*

constrained Pareto front) podem ser aprendidas e então usadas para guiar os operadores de evolução subsequentes e, conseqüentemente, a capacidade de busca seria melhorada de forma eficiente.

1.2 Objetivos

O principal objetivo deste trabalho é propor um novo algoritmo evolucionário para resolver CMOPs, utilizando predição nos espaços objetivo e de decisão para melhorar a busca por soluções viáveis e de alta qualidade. Especificamente, o P-NSBiDiCo busca:

1. Melhorar a eficiência e a eficácia na resolução de CMOPs;
2. Desenvolver uma abordagem de predição para identificar regiões promissoras no espaço de busca;
3. Implementar novas estratégias para a seleção dos parâmetros da evolução diferencial;
4. Validar o desempenho do algoritmo proposto por meio de testes em *benchmarks* e problemas do mundo real.

1.3 Contribuições

As principais contribuições deste trabalho são:

- O desenvolvimento do algoritmo P-NSBiDiCo, que avança o estado da arte em otimização multiobjetivo com restrições, ao integrar predição nos espaços objetivo e de decisão.
- A introdução de novas estratégias para a seleção dos parâmetros de evolução diferencial, visando melhorar a convergência e a diversidade das soluções.
- A validação experimental do algoritmo proposto em 46 funções de *benchmark* e 23 problemas do mundo real, evidenciando seu potencial em superar algoritmos existentes, especialmente em cenários práticos.

1.4 Estrutura do Documento

O restante deste documento está organizado da seguinte forma:

- **Capítulo 2:** Apresenta a definição formal e os conceitos fundamentais dos CMOPs, além de revisar o estado da arte em algoritmos evolucionários aplicados a esses problemas. Também discute as métricas de desempenho utilizadas para avaliar os algoritmos e analisa os sucessos e limitações das abordagens atuais.
- **Capítulo 3:** Descreve em detalhe o fluxo de execução do algoritmo proposto, incluindo todas as equações relevantes e os pseudocódigos que suportam sua implementação.
- **Capítulo 4:** Apresenta a configuração dos experimentos realizados, além de discutir os resultados experimentais e compará-los com modelos do estado da arte, tanto em problemas de *benchmark* quanto em aplicações do mundo real.
- **Capítulo 5:** Fornece as considerações finais, destacando as limitações do modelo proposto e sugerindo direções para trabalhos futuros.

2

Problemas multiobjetivos com Restrições: Definições e Abordagens

Problemas que envolvem a otimização de dois ou mais objetivos conflitantes são chamados de problemas multiobjetivos (MOPs). A otimização simultânea destes objetivos muitas vezes precisam coexistir com um conjunto de restrições do problema, dando origem aos problemas multiobjetivo com restrições (CMOPs) (Yang et al., 2023a). Algoritmos evolucionários multiobjetivos com restrições (CMOEAs) tornaram-se uma alternativa muito popular para abordar esses problemas de otimização. Estes algoritmos são inspirados pelos mecanismos de evolução biológica, como a seleção natural e variação genética. Desse modo, estes algoritmos conseguem encontrar um conjunto de soluções que podem ser aproximadamente ótimas sem a necessidade de informações de domínio específico, como diferenciabilidade, continuidade e unimodalidade do problema (Liu et al., 2023). Os algoritmos evolucionários (EAs) utilizam mecanismos de busca aleatória heurística e possuem forte capacidade de cooperar com incerteza, o que os torna adequados para encontrar soluções ótimas globais (Jin; Branke, 2005).

Este capítulo apresenta, na seção 2.1, a definição formal e os principais conceitos de CMOPs, incluindo as definições básicas, os tipos de CMOPs e tipos de dificuldades de otimização impostas. A seção 2.2 traz as principais abordagens de CMOEAs para tratar este tipo de problema. A seção 2.3 apresenta algumas métricas de desempenho para comparação de CMOEAs e, por fim, na seção 2.4, uma discussão de relevantes capacidades e limitações das abordagens de CMOEAs presentes na literatura.

2.1 CMOPs: Definição Formal

Problemas multiobjetivo com restrições (CMOPs) contêm diversos objetivos conflitantes que devem ser otimizados simultaneamente enquanto satisfazem um conjunto de restrições. Um CMOP é matematicamente formalizado da seguinte forma (Liang et

al., 2023):

$$\begin{aligned} \min \mathbf{F}(\mathbf{x}) &= [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ \dots \ f_m(\mathbf{x})]^T \in \mathbb{R}^m \\ \text{s.t.:} \quad &\begin{cases} g_j(\mathbf{x}) \leq 0, \ j = 1, \dots, p \\ h_j(\mathbf{x}) = 0, \ j = p + 1, \dots, l \\ x_k^{\min} \leq x_k \leq x_k^{\max}, \ k = 1, \dots, n \end{cases} \end{aligned} \quad (2.1)$$

onde cada $\mathbf{x} = [x_1, \dots, x_n]^T \in S$ é um vetor de decisão n -dimensional, x_k é a k -ésima variável de decisão, $S \subset \mathbb{R}^n$ é o espaço de decisão, \mathbf{F} é o vetor objetivo contendo m funções objetivo, $g_j(\mathbf{x})$ é a j -ésima restrição de desigualdade, $h_j(\mathbf{x})$ é a $(j - p)$ -ésima restrição de igualdade e x_k^{\min} e x_k^{\max} são os limites de x_k . O nível da j -ésima violação de restrição para \mathbf{x} é

$$CV_j(\mathbf{x}) = \begin{cases} \max(0, g_j(\mathbf{x})), & \text{se } j \leq p, \\ \max(0, |h_j(\mathbf{x})| - \epsilon), & \text{caso contrário.} \end{cases} \quad (2.2)$$

onde ϵ é um pequeno valor positivo (ex. $\epsilon = 10^{-4}$) para relaxar as restrições de igualdade. Desse modo, o total de violação de restrições é

$$CV(\mathbf{x}) = \sum_{j=1}^l CV_j(\mathbf{x}). \quad (2.3)$$

onde \mathbf{x} é uma solução viável se $CV(\mathbf{x}) = 0$.

A solução final de um CMOP consiste em um conjunto de soluções ótimas de Pareto (\mathcal{PS}) que melhor representa o *tradeoff* entre os objetivos e que formam, evidentemente, um conjunto de soluções viáveis (Liang et al., 2023).

2.1.1 Definições Básicas

Apresenta-se algumas definições básicas necessárias para fundamentar a revisão da literatura.

- *Dominância de Pareto*: Considerando as m funções objetivo de um CMOP na Equação (2.1), e quaisquer duas soluções \mathbf{x}_a e \mathbf{x}_b do conjunto de candidatos composto pela população e suas proles, com $a = 1, \dots, 2N, b = 1, \dots, 2N, a \neq b$, considere-se que \mathbf{x}_b é dominado por \mathbf{x}_a , ou seja, $\mathbf{x}_a \prec \mathbf{x}_b$, se qualquer uma das seguintes condições é verdadeira:

1. Ambas \mathbf{x}_a e \mathbf{x}_b são viáveis e \mathbf{x}_a domina \mathbf{x}_b , isto é, $\forall i \in \{1, 2, \dots, m\}, f_i(\mathbf{x}_a) \leq f_i(\mathbf{x}_b)$ e $\exists i \in \{1, 2, \dots, m\}, f_i(\mathbf{x}_a) < f_i(\mathbf{x}_b)$.
2. \mathbf{x}_a é viável, enquanto \mathbf{x}_b não.

3. Ambas \mathbf{x}_a e \mathbf{x}_b são inviáveis e \mathbf{x}_a tem menor violação de restrições, ou seja, $CV(\mathbf{x}_a) < CV(\mathbf{x}_b)$.
- *Região Viável*: O espaço de soluções viáveis do CMOP na Equação (2.1) é definido como $\mathbb{F} = \{\forall \mathbf{x} \in \mathbb{R}^n \mid CV(\mathbf{x}) = 0\}$.
 - *Soluções Ótimas de Pareto*: Uma solução $\mathbf{x}_a \in \mathbb{F}$ é chamada de solução ótima de Pareto de um CMOP se, e somente se, $\neg \exists \mathbf{x}_b \in \mathbb{F} \mid \mathbf{x}_b \prec \mathbf{x}_a$.
 - *Conjunto Ótimo de Pareto*: O conjunto de soluções ótimas de Pareto de um CMOP é definido como $\mathcal{PS} = \{\mathbf{x}_a \in \mathbb{F} \mid \neg \exists \mathbf{x}_b \in \mathbb{F}, \mathbf{x}_b \prec \mathbf{x}_a\}$.
 - *Frente de Pareto*: A frente de Pareto de um CMOP é definida como $\mathcal{PF} = \{\mathbf{F}(\mathbf{x}_a) \mid \mathbf{x}_a \in \mathcal{PS}\}$ e consiste na imagem do conjunto ótimo de Pareto no espaço objetivo, $f : \mathcal{PS} \rightarrow \mathcal{PF}$.

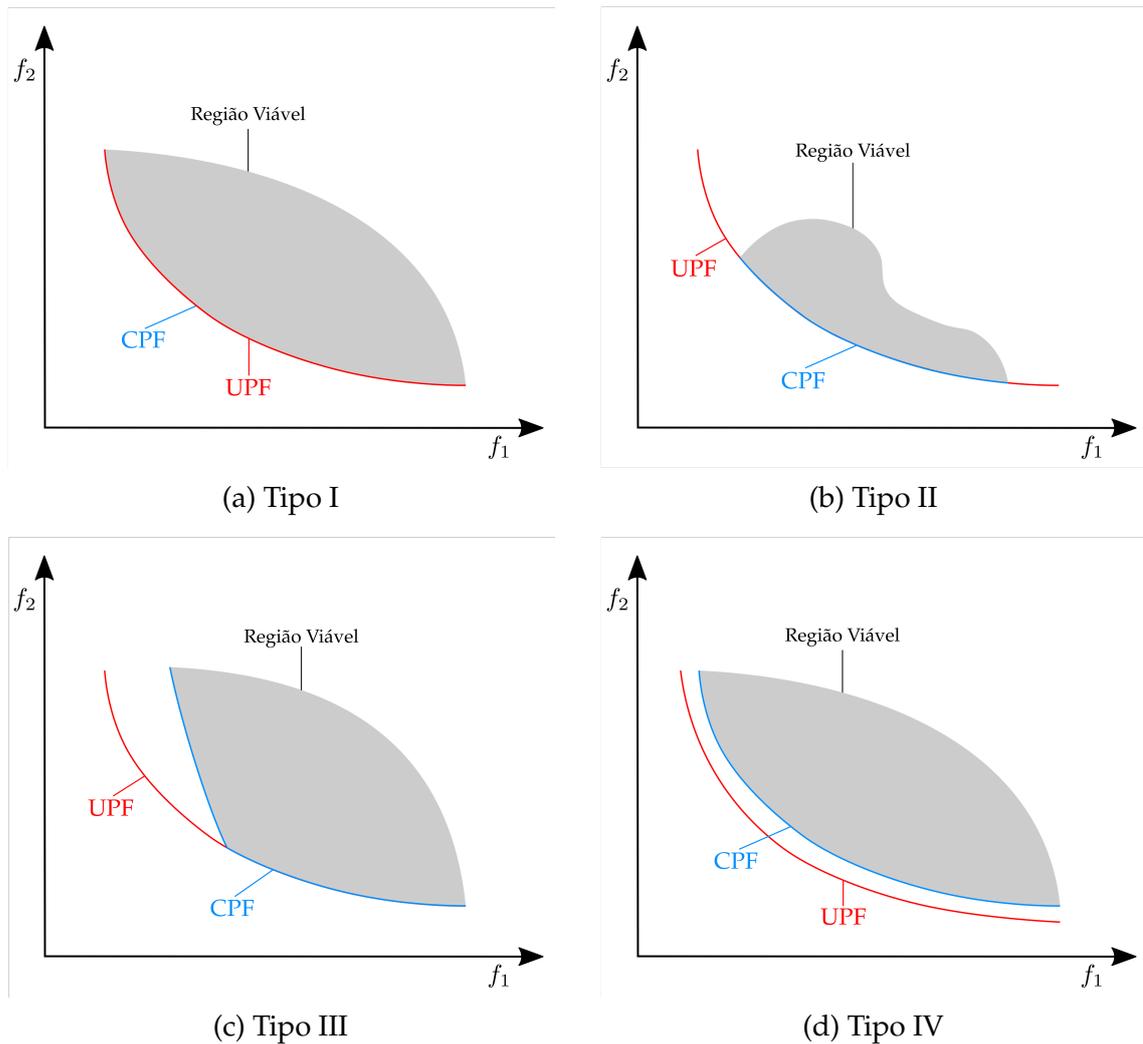
2.1.2 Tipos de CMOPs

De acordo Ma and Wang (2019), os CMOPs podem ser categorizados em quatro categorias diferentes com base em suas características da fronteira de Pareto restrita (CPF) e da fronteira de Pareto irrestrita (UPF), tal como ilustrado na Figura 9. No tipo I, apresentado na Figura 9(a), toda a sua UPF situa-se na mesma posição da CPF, ou seja, a CPF e UPF são idênticas. O tipo II, mostrado na Figura 9(b), há coincidência parcial entre a CPF e UPF, uma vez que as restrições fazem com que parte da UPF se torne inviável. Este tipo II ocorre frequentemente em problemas do mundo real. O tipo III, apresentado na Figura 9(c), contém parte da sua CPF coincidindo com a UPF e o restante da CPF se constitui no limite de viabilidade para a solução do problema. Aqui também, a CPF coincide de forma parcial com a UPF. No tipo IV, ilustrado na Figura 9(d), toda a UPF está localizada fora da região de viabilidade e, portanto, não há coincidências entre a CPF e a UPF. Nesse caso, todas as soluções ótimas viáveis são interiores aos limites estabelecidos pela UPF.

2.1.3 Tipos de Dificuldades de Otimização de CMOPs

Lidar com problemas multiobjetivo com restrições, quando comparados aos problemas de único objetivo, consiste em uma tarefa desafiadora devido à dificuldade aumentada para explorar o espaço de busca, pois se faz necessário que todas as restrições do problema sejam satisfeitas. A implicação direta da existência de restrições é que apenas parte do espaço de busca contenha soluções viáveis, usualmente, ocasionando acréscimo do esforço computacional de processamento (Tan; Feng; Jiang, 2021; Zhan et al., 2022).

Figura 1 – Ilustração dos Tipos de CMOPs.



Fonte: Adaptado de Ma and Wang (2019).

Os CMOPs possuem características intrínsecas que apresentam algumas dificuldades para o algoritmo que está sendo empregado para otimização. Entre eles, dificuldade de encontrar soluções ótimas viáveis, de convergência do algoritmo e de manter a diversidade ao longo do processo de busca são os principais desafios primários que precisam ser superados durante o processo evolucionário (Fan et al., 2020).

- *Dificuldade de viabilidade*: Consiste na dificuldade de encontrar soluções que atendam a todas as restrições impostas pelo problema. Nesta situação, o espaço de soluções viáveis pode ser muito pequeno quando comparado com o espaço total de soluções possíveis. Dessa forma, pode se tornar muito difícil encontrar soluções viáveis devido à presença de: 1) restrições muito complexas, não lineares e/ou altamente interdependentes; 2) alta dimensionalidade, decorrente da presença de um grande número de variáveis e restrições; e, 3) espaço de busca de tamanho

reduzido. De forma geral, a \mathcal{PF} dos CMOPs com dificuldade de diversidade apresentam segmentos ou partes difíceis de serem alcançados do que outras partes devido a presença de largas regiões de inviabilidade, resultando na dificuldade dos CMOEAs conseguirem a \mathcal{PF} completa (Fan et al., 2020).

- *Dificuldade de convergência:* Esta dificuldade está relacionada a capacidade do algoritmo encontrar soluções próximas a \mathcal{PF} , devido a fatores como: 1) objetivos fortemente conflitantes, pois a melhora em um objetivo pode causar degradação significativa em outro; 2) complexidade do problema, isto é, problemas multimodais ou altamente não-lineares tornam a convergência para a \mathcal{PF} desafiadora; 3) baixa eficácia do algoritmo que pode apresentar dificuldades de explorar e explorar o espaço de busca de forma eficiente, causando convergência lenta, além de poder ficar preso em ótimos locais. Em CMOPs com dificuldade de convergência, a proporção de regiões viáveis com respeito a todo o espaço de busca normalmente é muito baixa, atrapalhando um CMOEA em sua busca por soluções viáveis (Fan et al., 2020).
- *Dificuldade de diversidade:* Neste caso, o CMOEA tem dificuldade de manter uma distribuição da população de soluções com diversidade ao longo da frente de Pareto. A diversidade das soluções é um aspecto crítico dos CMOEAs, pois assim é possível gerar um conjunto de soluções ótimas espalhadas e espaçadas para a tomada de decisão. A perda de diversidade está relacionada a: 1) tendência a agrupamento, pois CMOEAs podem se concentrar em uma região específica da \mathcal{PF} e ignorando outras; 2) pressão de seleção, durante o processo evolucionário decorrente do favorecimento da busca em determinadas regiões da \mathcal{PF} , reduzindo a variedade das soluções; e 3) equilíbrio entre exploração e exploração, pois encontrar o equilíbrio correto entre a exploração de novas áreas do espaço de busca e explorar áreas já conhecidas é um grande desafio, especialmente em problemas de alta-dimensionalidade ou com \mathcal{PF} muito complexas. CMOPs com dificuldade de convergência impedem a convergência do CMOEA em direção a \mathcal{PF} e, por isso, CMOEAs encontram dificuldade em alcançar toda a \mathcal{PF} devido as regiões de inviabilidade precisam serem atravessadas à medida que convergem para a \mathcal{PF} (Fan et al., 2020).

Os CMOPs são problemas de otimização complexos, para os quais têm-se empregado Algoritmos Evolucionários (EAs) devido a sua poderosa capacidade de busca (Liang et al., 2023; Zhan et al., 2022). No entanto, é difícil determinar quais soluções são as melhores devido a presença de soluções com bons valores de aptidão em regiões de inviabilidade no espaço de busca (Zhan et al., 2022). Portanto, acoplado ao EA, é necessário utilizar um mecanismo para lidar com as restrições (CHT), que tem como

objetivo guiar o algoritmo na forma como as soluções são selecionadas para gerações futuras, ao passo que o EA atua como motor de busca (Liang et al., 2023; Zhan et al., 2022).

A seção a seguir apresenta as principais abordagens de algoritmos evolucionários para lidar com problemas multiobjetivo com restrições (CMOPs) presentes na literatura recente.

2.2 Algoritmos Evolucionários para CMOPs

Diversas abordagens para tratar problemas multiobjetivo com restrições têm sido propostas e, em geral, elas podem ser divididas em cinco categorias: métodos baseados em função de penalidade, métodos baseados na separação de objetivos e restrições, métodos baseados na definição de problema multiobjetivo, métodos baseados em estratégia co-evolucionária e métodos híbridos.

2.2.1 Métodos Baseados em Função de Penalidade

Um método baseado em função de penalidade adiciona fatores de penalidade às funções objetivo para lidar com as restrições em um CMOP. Os métodos desta categoria consideram o grau de violação das restrições para definir o termo de penalização. Assim, um CMOP é transformado em um problema sem restrições por adicionar-se diferentes penalidades para diferentes graus de violação de restrições (Feng et al., 2023; Liang et al., 2023).

Yang et al. (2019) propuseram o MODE-SaE (*multi-objective differential evolutionary algorithm*), um algoritmo que emprega evolução diferencial (DE) multiobjetivo que integra uma versão melhorada da CHT ϵ , uma estratégia de arquivo para manutenção de soluções elitistas viáveis, e um mecanismo de busca de parâmetros de auto-alteração. O mecanismo melhorado da CHT ϵ auto-adaptativamente ajusta o nível ϵ de acordo com os valores de violação de restrições dos indivíduos, prevenindo que o nível ϵ não seja razoável. As soluções viáveis são armazenadas em um arquivo externo com um esquema de manutenção.

Fan et al. (2019a) apresentaram uma técnica para lidar com restrições baseada numa versão adaptativa da ϵ -CHT com um algoritmo evolucionário baseado em decomposição (MOEA/D), chamado MOEA/D-IEpsilon. Este método ajusta o nível ϵ dinamicamente de acordo com o percentual de soluções viáveis em relação ao total na população atual.

Zhu, Zhang and Lin (2020) propuseram um *framework* com a estratégia "detectar-e-escapar", em que detecta se a população está ou não presa em um ótimo local e escapa-

se dela. Para isso, a taxa de viabilidade e a taxa de mudança de todas as violações de restrições são utilizadas para detectar estagnação e ajustar o peso da violação de restrições para guiar a busca com vistas escapar de estados de estagnação. Este mecanismo foi acoplado ao EA chamado MOEA/D-DAE para tratar CMOPs.

2.2.2 Métodos Baseados na Separação dos Objetivos e Restrições

Um método baseado na separação de objetivos e restrições considera os objetivos e restrições separadamente durante o processo de evolução. Nesta estratégia o algoritmo deve considerar primeiramente se as soluções na população satisfazem o conjunto de restrições. As principais abordagens da literatura nesta família de métodos são princípio de dominância de restrições (CDP), *ranking* estocástico (SR) e o método de restrição ϵ (Feng et al., 2023; Liang et al., 2023).

No estudo apresentado por (Wang et al., 2019), os autores desenvolveram uma extensão do algoritmo CoDE (*composite differential evolution*) (Wang; Cai; Zhang, 2011), nomeado C²DE. O algoritmo utiliza três estratégias complementares para gerar vetores tentativa visando preservar diversidade e atingir convergência. Além disso, os autores propuseram uma CHT baseada na regra de viabilidade e no mecanismo ϵ . No estudo apresentado por Wang, Liang and Zhang (2019), os autores propuseram o CCMODE (*cooperative differential evolution framework*) que trata um problema de M objetivos dividindo-o em M subpopulações de objetivo único com restrições e uma população de arquivo. Estas $(M + 1)$ -populações cooperam para otimizar todos os objetivos e restrições de um CMOP.

2.2.3 Métodos Baseados na Definição de Problema multiobjetivo

Um método baseado na definição de problema multiobjetivo considera que as restrições de um CMOP são tratadas como um objetivo adicional. Assim, o CMOP se transforma em um MOP permitindo que MOEAs sejam empregados como soluções possíveis (Feng et al., 2023; Liang et al., 2023). Esta estratégia assume que a função objetivo e a função de restrições possuam a mesma prioridade. A função objetivo após a transformação é dada pela Equação (2.4).

$$\min \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ \dots \ f_m(\mathbf{x}), CV(\mathbf{x})]^T \in \mathbb{R}^m \quad (2.4)$$

Um novo CMOEA que integra violação de restrições como uma dimensão adicional a otimização dos objetivos, chamado dp-ACS, foi proposto por Yang et al. (2023a). Esta abordagem apresenta uma função de intensidade de restrição adaptável que ajusta dinamicamente a região de intensidade de restrição, permitindo que a população principal explore perto dos limites das regiões viáveis e melhore a distribuição da população.

A população auxiliar conduz a otimização sem considerar as restrições concorrentemente, fornecendo informações valiosas para além da sua região viável. A colaboração entre as duas populações se dá por meio da combinação de populações de descendentes para a seleção ambiental. Yang et al. (2023b) também propuseram um CMOEA assistido pelo grau de violação de restrições como um objetivo adicional. Na abordagem proposta, a população principal é utilizada para buscar todo o espaço, otimizando o objetivo transformado $m + 1$ MOP, enquanto o arquivo externo evolui para explorar regiões viáveis.

2.2.4 Métodos Baseados em Co-evolução e Multi-estágios

Os métodos baseados em co-evolução e multi-estágios adotam uma abordagem co-evolucionária como a técnica para lidar com restrições e adota estratégias de multi-populações e multi-estágios evolucionários para transformar um CMOP em um problema de otimização co-evolucionário (Feng et al., 2023).

Liu and Wang (2019) apresentaram um *framework* de duas fases, chamado ToP (*two-phase framework*). Nesta abordagem, a primeira fase tem como objetivo encontrar áreas viáveis promissoras por transformar um CMOP em um problema de otimização com um único objetivo, ignorando o conflito entre funções objetivo. A segunda fase implementa um CMOEA específico para guiar a população em direção à verdadeira fronteira do Pareto.

No estudo apresentado em (Li et al., 2019) os autores propuseram um mecanismo com duplo arquivo e livre de parâmetros para lidar com restrições chamado de C-TAEA (*two-archive evolutionary algorithm for constrained multiobjective optimization*). Esta abordagem mantém dois arquivos de soluções complementares e colaborativos. Enquanto um dos arquivos é denominado como arquivo orientado a convergência (CA) e foca principalmente em guiar a população em direção a fronteira do Pareto, o outro é denotado como arquivo orientado a diversidade (DA) em que tende a explorar áreas sub-exploradas pelo CA mantendo, desse modo, a diversidade da população.

Em outro estudo, Fan et al. (2019b) propuseram um CMOEA com duas fases, um *framework* denominado *Push and Pull Search (PPS)*. Esta abordagem divide o processo de busca em dois estágios distintos, *pull* e *push*. No estágio de busca *push*, as restrições são ignoradas, ajudando *PPS* a atravessar as regiões inviáveis para determinar a fronteira do Pareto sem restrições. Na etapa de busca *pull*, considera-se tanto os valores de restrições quanto os de função objetivo com a aplicação da CHT ϵ (Fan et al., 2019a) para determinar entre os indivíduos viáveis encontrados na fase *push* aqueles que não violam restrições.

No estudo apresentado em (Yu et al., 2021), os autores propuseram um DE multiobjetivo bifásico com propósito direcionado chamado PDTP-MDE (*purpose directed two-phase multiobjective differential evolution*). No primeiro estágio do processo evolucionário, o PDTP-MDE visa atingir equilíbrio entre convergência e diversidade, enquanto utiliza-se a taxa de viabilidade da população como um critério auxiliar. O segundo estágio mantém a viabilidade e a diversidade preservando as soluções viáveis e soluções inviáveis promissoras de acordo com o *status* de evolução da população. As soluções viáveis obtidas no processo são armazenadas e atualizadas em um arquivo que se constituirá no conjunto de soluções ótimas do Pareto.

Yang et al. (2021) projetaram um algoritmo de DE multiobjetivo baseado em dominância e um mecanismo de alternância de CHT chamado MODE-CHS (*multi-objective differential evolution algorithm based on domination and a mechanism of constraint-handling switching*). No mecanismo de alternância, se não houver nenhuma solução viável na população o processo evolucionário é executado com uma CHT; em caso contrário, executa-se sem CHT. O MODE-CHS armazena soluções viáveis em um arquivo que evolui juntamente com a população. Além disso, as soluções tentativas do arquivo externo participam da atualização dos indivíduos de forma a melhorar a distribuição da população.

Tian et al. (2021) propuseram um *framework* co-evolucionário para CMOPs chamado CCMO (*coevolutionary constrained multiobjective optimization framework*), no qual evolui duas populações que compartilham informações úteis entre elas. Uma população soluciona o CMOP original, ao passo que a outra evolui um problema auxiliar derivado do original. Outro *framework* cooperativo foi proposto em (Peng; Liu; Goodman, 2021). Nesta abordagem utiliza-se uma versão melhorada de vetores de peso diretos para tratar CMOPs e consiste de duas fases alternáveis. Na primeira fase utiliza-se duas subpopulações, em que uma explora regiões viáveis e, a outra, explora todo o espaço de busca. A segunda fase tem como principal objetivo encontrar soluções ótimas de Pareto. A alternância entre as duas fases se dá com base na informação dos valores objetivos encontrados durante o processo evolucionário.

Liu, Wang and Tang (2022) projetou um novo CMOEAs para resolver CMOPs via coevolução bidirecional, chamado BiCo (*bidirectional coevolution*). Na abordagem proposta duas populações são utilizadas, sendo a população principal e a população do arquivo. A população principal armazena soluções viáveis e inviáveis, ao passo que a população do arquivo mantém as soluções inviáveis não-dominadas. A interação entre as duas populações se dá por meio de um mecanismo de seleção de soluções para recombinação restrito que visa produzir descendentes promissores. Desse modo, a população é guiada em direção a fronteira do Pareto por ambos os lados viável e inviável do espaço de busca. Baseado no BiCo, Mendes, Araújo and Farias (2023)

propuseram o NSBiDiCo, uma abordagem que utiliza os operadores de cruzamento e mutação diferencial da evolução diferencial, bem como um esquema de seleção baseado em não-dominância.

Bao et al. (2022) projetou um EA de dois estágios baseado em arquivo nomeado AT-CMOEA (*archive-based two-stage evolutionary algorithm*). Nesta abordagem, o processo de busca é dividido em dois estágios, exploração e exploração. No estágio de exploração o espaço de busca é explorado a fim de encontrar regiões promissoras em que, para isso, duas populações com diferentes prioridades para restrições e objetivos são mantidas. O estágio de exploração tem como objetivo obter soluções ótimas bem distribuídas na fronteira de Pareto pela utilização de informações úteis encontradas e arquivadas durante o estágio de exploração. Desse modo, as duas populações co-evoluem colaborativamente em direção a fronteira de Pareto restrita.

O estudo apresentado em (Tian et al., 2022) também propõe um EA de dois estágios, nomeado CMOEA-MS (*two-stage framework for CMOEA*). Esta abordagem ajusta as estratégias de avaliação de aptidão durante o processo evolucionário para equilibrar de forma adaptativa a otimização dos objetivos e a satisfação das restrições. De acordo com a taxa de viabilidade da população, o algoritmo pode alternar entre os dois estágios, habilitando a população a atravessar a região inviável do espaço de busca e chegar à regiões viáveis em um estágio e espalhar-se ao longo dos limites viáveis no outro estágio.

Huang et al. (2023) propuseram um mecanismo de busca de população auxiliar melhorada (EAPS) baseado em otimização co-evolucionária para abordar o desafio da manutenção da distribuição das soluções em regiões de viabilidade discretas e pequenas enquanto equilibra convergência e viabilidade em CMOPs complexos. A estratégia EAPS emprega uma população auxiliar que utiliza um novo método de agregação dentro do *framework* de decomposição para evoluir independentemente sem considerar as restrições, transmitindo informações sobre diversidade unidirecionalmente para a população principal.

As principais características da EAPS incluem o uso unilateral de uma população auxiliar não-restrita, Pop2, que permite a transmissão de informações de diversidade para a população principal, Pop1, sem ser influenciada por soluções ótimas locais; A busca da Pop2 com o *framework* de decomposição utilizando o novo método de agregação; e a substituição de soluções da Pop2 em Pop1 para evitar que Pop2 convirja para a UPF e para manter a eficiência na condução contínua da Pop1 por meio de várias explorações. Esta abordagem equilibra os requisitos de viabilidade e convergência enquanto promove diversidade e melhora a distribuição da população principal ao resolver CMOPs complexos.

O DPACMO (*dual population approximation CPF algorithm*) é um CMOEA com duas populações, que foi introduzido por Zhou, Zhang and Suganthan (2023), e foi projetado para se aproximar da CPF dos lados viável e inviável dos limites de restrições. A população1 foca em aproximar a CPF a partir das regiões viáveis, ao passo que a população2 visa alcançar a CPF a partir do lado inviável. Nesta abordagem, uma estratégia de migração para trocar informações úteis entre as populações é utilizada para melhorar o desempenho na aproximação da CPF a partir de diferentes direções. O processo evolucionário envolve dois estágios. Inicialmente, ambas as populações aproximam a CPF e UPF, seguido da população1 continuando a aproximação da UPF do lado das regiões viáveis, enquanto a população2 ajusta sua abordagem a medida que o nível do relaxamento das restrições muda, gradualmente movendo em direção a CPF do lado da região inviável do espaço de busca. A população1 emprega o princípio de dominância de restrições (CDP) para aproximar a CPF do lado viável, enquanto a população2 utiliza o método de restrição *epsilon* melhorado para aproximar de ambos os lados viável e inviável.

Zeng, Cheng and Liu (2023) introduzem o algoritmo CCMOTA (*coevolutionary framework assisted by two archives for CMOPs*), uma abordagem co-evolucionária para otimizar CMOPs. O CCMOTA utiliza dois arquivos auxiliares: um arquivo inverso, que armazena soluções inviáveis próximas à fronteira Pareto e um arquivo de diversidade, que promove a exploração de diferentes regiões do espaço objetivo. Além disso, o algoritmo emprega uma estratégia de otimização de único objetivo para auxiliar na descoberta de soluções viáveis em problemas altamente restritos.

Xia, Chong and Dong (2023) desenvolveu o CMOEA-TSRA (*two-stage resources allocation CMOEA*), um CMOEA que apresenta uma estratégia de alocação de recursos em dois estágios. O estágio inicial adaptativamente aloca poucos indivíduos para exploração das regiões viáveis já descobertas e aumenta a alocação para explorar potenciais regiões viáveis. Esta abordagem visa equilibrar eficientemente exploração e exploração, melhorando a capacidade do algoritmo de descobrir soluções promissoras e diversas. No estágio seguinte, a alocação de recursos troca a prioridade para maior exploração de regiões viáveis já descobertas, enquanto reduz a alocação para a exploração de áreas potenciais. Além disso, esta abordagem inclui um mecanismo de reprodução baseado em vizinhança e um método de atribuição de aptidão baseado na distribuição da população.

O CMOEA-TMC (*three-stage multi-population coevolution CMOEA*), proposto por Shi et al. (2023), emprega uma estratégia de três estágios integrando evolução paralela e co-evolução. Nesta abordagem a população principal é particionada em múltiplas sub-populações que são submetidas a uma evolução paralela granulada para acelerar a convergência e facilitar a busca global. Concorrentemente, uma população ajudante é

incorporada para melhorar a manutenção da diversidade por compartilhar informações de soluções inviáveis com a população principal durante a geração de descendentes e a evolução cruzada.

A estratégia de três estágios do CMOEA-TMC começa com ênfase em convergência rápida e busca global através da divisão em subpopulações. O estágio seguinte prioriza a evolução independente dessas subpopulações para manter a diversidade. O estágio final é dedicado para minerar valores extremos, objetivando identificar soluções ótimas globais.

O algoritmo evolucionário cooperativo chamado SaE-CMO (*dual-population evolutionary algorithm based on self-adaptive epsilon method*) foi proposto por Song et al. (2024) para tratar CMOPs. Esta abordagem tem como objetivo equilibrar convergência e viabilidade através de uma abordagem de duas populações. O SaE-CMO contém uma população principal e uma auxiliar que se complementam entre si pela utilização do método ϵ autoadaptativo com diferentes critérios de comparação. Este método permite que a população 2 preserve soluções bem restritas e com fraca convergência, ajudando assim a população 1 manter a diversidade em CMOPs complexos. Além disso, a população 1 emprega o método de relaxamento de restrições, facilitando a utilização de soluções inviáveis promissoras nos estágios iniciais para navegar através das regiões de inviabilidade.

O algoritmo SaE-CMO introduz um método ϵ autoadaptativo que captura o nível de violação de restrições em toda a população, habilitando a utilização de soluções inviáveis promissoras ao longo do processo de busca em ambas as populações. Esta abordagem facilita a tolerância a restrições com dois critérios de comparação distintos adaptados para cada população. O critério na população 1 guia a população através da região inviável pela utilização de soluções inviáveis promissoras quando soluções viáveis são escassas. O critério da população 2 mantém ambas as soluções inviáveis promissoras e soluções viáveis ótimas, garantindo diversidade contínua e provendo pistas evolucionárias suplementares para a população 1.

Um método de multitarefa evolucionária baseada em subconjuntos de restrições chamado SC-EMT (*constraint subsets-based EMT algorithm*), para tratar problemas de otimização multiobjetivo com restrições foi introduzido por Yu et al. (2024). Esta abordagem transforma um CMOP em um problema de otimização multitarefa por decompor o conjunto de restrições em múltiplos subconjuntos em que cada um é combinado com um problema multiobjetivo derivado para criar um novo CMOP. A principal tarefa consiste em encontrar a frente do Pareto verdadeira (CPF) enquanto tarefas auxiliares objetivam promover diversidade ao processo de busca em diferentes direções.

Qiao et al. (2024) introduziu um novo CMOEA, chamado DPCPRA (*dual-population EA based on dynamic constraint processing and resources allocation*), que incorpora um meca-

nismo de processamento de restrição dinâmico e um esquema de alocação de recursos dinâmico para tratar CMOPs. O algoritmo utiliza uma população principal que foca em otimizar diretamente todas as restrições para identificar soluções viáveis ótimas de Pareto, de modo a melhorar a viabilidade. Concorrentemente, a população auxiliar emprega um mecanismo de processamento de restrições dinâmico que gradualmente aumenta o número de restrições processadas, aproveitando soluções inviáveis para identificar regiões viáveis eficazmente. Um esquema dinâmico de alocação de recursos é projetado para alocar recursos computacionais limitados às duas populações com base no *feedback* de desempenho, melhorando o comportamento de busca do algoritmo.

Um *framework* multitarefa para CMOPs pela integração de conceitos de evolução multitarefa foi desenvolvido por (Ming et al., 2024) modela um CMOP como um problema de três tarefas: uma tarefa primária baseada no princípio de dominância de restrições (CDP), uma tarefa auxiliar sem restrições baseada na dominância de Pareto (PD) e uma tarefa auxiliar com restrições. Utilizando diferentes CHTs com diferentes prioridades para viabilidade e convergência, o *framework* trata CMOPs com diferentes características. Os autores introduziram uma estratégia de expressão e transferência de conhecimentos para facilitar a transferência de conhecimentos entre tarefas, utilizando a viabilidade e convergência das soluções para determinar os fatores de aptidão e transferindo seletivamente soluções úteis para a tarefa-alvo para evitar a transferência negativa e reduzir o consumo de tempo. Baseado neste *framework*, um novo algoritmo chamado CMOEMT foi desenvolvido, compreendendo um estágio de evolução para guiar a população em direção às posições desejadas para cada tarefa e outro de transferência de conhecimento para cooperativamente resolver as tarefas.

Li et al. (2024) propôs o MTOTC (*multi-tasking optimization algorithm*), um algoritmo multi-tarefas para tratar CMOPs. Esta abordagem utiliza uma técnica de tarefa clone para replicar o CMOP alvo com q restrições em $q + 1$ cópias resultando em $q + 2$ tarefas. Cada tarefa clonada é gerenciada por uma população independente empregando uma sequência de CHTs únicas, desacoplando as restrições de múltiplas formas. A abordagem proposta facilita o compartilhamento online de informações entre a tarefa alvo e as tarefas clonadas, maximizando a utilização do histórico de busca.

Um algoritmo co-evolucionário multiobjetivo cooperativo paralelo para melhorar o desempenho de métodos multiobjetivo paralelos já existentes foi proposto em (Harada, 2024). Dois métodos de paralelização foram avaliados, o GP-CCMO e DP-CCMO, sendo um modelo paralelo global para avaliação de soluções em paralelo e um modelo híbrido que distribui populações cooperativas enquanto executa cada avaliação em paralelo. O modelo global fornece paralelização simples, enquanto o modelo híbrido explora as características da estrutura co-evolutiva para otimizar a eficiência do algoritmo.

Um *framework* MOEA/D baseado em duas populações para resolver CMOPs com grandes regiões de inviabilidade foi apresentado em (Wang et al., 2024). Esta abordagem divide o processo evolucionário em dois estágios. O estágio 1 foca em permitir que a população 1 alcance a frente de Pareto não restrita (UPF), servindo de base para a população 2 navegar através das regiões inviáveis no estágio 2. O estágio 2 envolve a população 2 auxiliando a população 1 convergir rapidamente para a CPF do lado inviável do espaço objetivo, promovendo uma busca bi-direcional pela CPF. Os autores embutiram o *framework* proposto em um novo CMOEa chamado BDCR.

2.2.5 Métodos Híbridos

Os métodos híbridos empregam uma ou mais das abordagens anteriores de forma a aproveitar os seus benefícios individuais de forma a gerar soluções mais robustas e eficientes.

Um novo algoritmo de evolução diferencial co-evolucionário auxiliado por aprendizado por reforço profundo para tratar CMOPs, chamado CEDE-DRL (*deep reinforcement learning assisted co-evolutionary differential evolution*), foi proposto em (Hu et al., 2023). O CEDE-DRL integra co-evolução na extração de dados para treinamento, garantindo a diversidade de amostras e melhorando a precisão da rede neural através da troca de informações entre múltiplas populações, bem como emprega múltiplas CHTs para seleção de proles. O aprendizado por reforço profundo (DRL) avalia os estados da população considerando viabilidade, convergência e diversidade, utilizando a melhoria geral como recompensa e seleciona populações de pais e arquivos adequados para mutação para prevenir convergência prematura e ótimos locais. O algoritmo CEDE-DRL utiliza quatro populações que evoluem simultaneamente para garantir a diversidade do conjunto de treinamento do DRL e emprega quatro CHTs com diferentes enviesamentos para lidar com vários CMOPs. A estratégia de DRL guia a seleção da população para melhorar a otimização do algoritmo e a eliminação do arquivo individual na estratégia de mutação evita a convergência prematura para ótimos locais.

Ye et al. (2024) propuseram um algoritmo evolucionário de mapeamento auto-organizado baseado em multitarefa, chamado MTSOM (*multi-task-based self-organizing mapping evolutionary algorithm*), para tratar CMOPs. O MTSOM emprega uma estratégia de tarefas principal e auxiliares em que duas populações inicialmente otimizam objetivos independentemente, seguido por uma exploração paralela da estrutura da população utilizando um mapa auto-organizado (SOM) de restrição-a-restrição. No estágio final a tarefa principal prioriza a viabilidade das soluções, enquanto a tarefa auxiliar foca nas restrições de alta prioridade, facilitando rápida convergência em direção a região viável. O MTSOM é dividido em três estágios: otimização dos objetivos livre

de restrições, exploração de regiões de viabilidade pela técnica de restrição-a-restrição e executa-se a tarefa auxiliar para restrição de maior probabilidade.

Wang et al. (2024) desenvolveu um CMOEA baseado em aprendizado profundo, chamado CMOEA-DRL (*deep reinforcement learning based CMOEA*). Esta abordagem foi introduzida para melhorar a seleção adaptativa de técnicas para lidar com restrições (CHTs) durante o processo evolucionário para CMOPs.

Os autores propuseram uma CHT baseada em aprendizado por reforço profundo e um método de seleção de operadores utilizando *Deep Q-Networks* (DQN) para capturar relações complexas existentes entre estados e ações. Para isto, os autores introduziram um método de cálculo para determinar estados e recompensas de uma população, avaliando efetivamente o desempenho de diferentes ações ao longo de diversos estados evolucionários. Este método integra informações dos objetivos e restrições para mapear com precisão o desempenho das ações.

O algoritmo CMOEA-MSWA, introduzido por Gu et al. (2024), foi projetado para melhorar a busca por regiões inviáveis por meio de uma estratégia coordenada envolvendo arquivos e vetores de pesos. Inicialmente, o arquivo atualiza os vetores de peso por preservar a diversidade da população, a qual, posteriormente, utiliza a informação de viabilidade para buscar regiões inviáveis. Esta interação circular melhora a detecção de soluções inviáveis com objetivos favoráveis, demonstrando desempenho competitivo em espalhamento e uniformidade.

O algoritmo enfatiza diversidade enquanto guia a população em direção a CPF com soluções viáveis bem distribuídas. Esta abordagem inclui um mecanismo de modificação adaptativo para soluções inviáveis com base em vetores de peso, uma estratégia de ajuste dos vetores de peso com base em arquivos e uma estratégia de poda em duas fases centrada na diversidade das soluções. Estas estratégias visam manter o desempenho estável para diferentes tipos de problemas por atualizar a distribuição dos vetores de peso, guiar a geração de soluções bem distribuídas e preservar a qualidade e quantidade do arquivo para garantir a diversidade.

A seção a seguir apresenta as principais métricas utilizadas para a avaliação de desempenho de algoritmos evolucionários.

2.3 Métricas de Desempenho

As principais métricas de desempenho buscam avaliar a qualidade das soluções dos CMOEAs com base na fronteira do Pareto obtida ao final da execução. As métricas avaliam a distribuição e a diversidade e são indicadores importantes da robustez do algoritmo em relação a mudança de parâmetros e das condições do problema,

bem como da sua capacidade de busca e ajuda-nos a entender se o algoritmo explora eficientemente o espaço de soluções. Estas métricas são fundamentais para avaliar e comparar CMOEAs, garantindo que estes gerem soluções de qualidade e robustas para problemas complexos de otimização multiobjetivo. As duas principais métricas utilizadas são a distância geracional invertida (IGD) (Bosman; Thierens, 2003) e hiper-volume (HV) (Zitzler; Thiele, 1999).

2.3.1 Distância Geracional Invertida (IGD)

A partir da fronteira do Pareto restrita do CMOP que está sendo avaliado, o valor do IGD é obtido por meio da distância Euclidiana média de cada solução para seu ponto mais próximo na CPF. Dessa forma, é possível avaliar a convergência e a diversidade simultaneamente. Quanto maior for o número de pontos de amostragem da CPF e quanto mais uniformemente estiverem distribuídos, mais precisos e fidedignos serão os resultados (Bosman; Thierens, 2003). O IGD pode ser computado pela Equação 2.5.

$$IGD(\mathbf{P}, \mathbf{P}^*) = \frac{\sum_{x^* \in \mathbf{P}^*} dist(x^*, \mathbf{P})}{|\mathbf{P}|} \quad (2.5)$$

onde, \mathbf{P} consiste na CPF uniformemente distribuída do CMOP, \mathbf{P}^* denota o conjunto de soluções obtidas pelo algoritmo e $dist$ indica a distância Euclidiana entre x^* e seu ponto mais próximo em \mathbf{P} calculado por

$$d(x^*, \mathbf{P}) = \min_{x \in \mathbf{P}} \left\{ \sqrt{\sum_{i=1}^m (x^*_i - x_i)^2} \right\} \quad (2.6)$$

Os resultados experimentais de algoritmos com menor valor de IGD indicam que estes possuem melhor diversidade e convergência.

2.3.2 Hiper-volume (HV)

O hiper-volume é amplamente utilizado para avaliar CMOPs pois, em grande maioria, a CPF de problemas do mundo-real não são conhecidas. O cálculo do hiper-volume não necessita da CPF, mas somente um ponto de referência é necessário para avaliar a qualidade do conjunto de soluções em termos de convergência e diversidade (Zitzler; Thiele, 1999). O hiper-volume pode ser computado pela Equação (2.7).

$$HV(\mathbf{P}^*) = vol \left(\bigcup_{\mathbf{x} \in \mathbf{P}^*} [f_1(\mathbf{x}, z_1^r)] \times \cdots \times [f_m(\mathbf{x}, z_m^r)] \right) \quad (2.7)$$

onde vol é a medida de Lebesgue, $\mathbf{z}^r = (z_1^r, \dots, z_m^r)$ denota uma série de pontos de referência no espaço objetivo e \mathbf{P}^* denota o conjunto de soluções obtidas pelo algoritmo. Algoritmos com maior valor de HV possuem melhor diversidade e convergência.

Na próxima seção, serão discutidos os sucessos e limitações das abordagens de CMOEAs para resolver CMOPs, bem como melhorias que podem ser adotadas para melhor desempenho dos CMOEAs.

2.4 Sucessos e Limitações das Soluções de CMOPs por EAs

Os avanços recentes na área de otimização multiobjetivo com restrições incluem novas abordagens que empregam novas técnicas para lidar com restrições, como a co-evolução, evolução em estágios e até mesmo propõem melhorias para CHTs existentes ou seus operadores. Além disso, algumas abordagens têm integrado aprendizado profundo e estratégias auto-adaptativas para melhorar a seleção de técnicas de tratamento de restrições e equilibrar convergência e viabilidade da população. No entanto, algumas limitações estão presentes, como a complexidade computacional e o desafio em equilibrar a convergência e diversidade da população, especialmente em problemas com restrições muito complexas. Portanto, enquanto os CMOEAs têm avançado em resolver problemas cada vez mais complexos, ainda há a necessidade de prover melhorias na eficiência computacional, adaptabilidade e equilíbrio entre convergência e diversidade e eficácia das soluções.

Algumas abordagens têm empregado o uso de técnicas de inteligência artificial, como *Deep Q-Networks* e outras abordagens, para melhorar o desempenho do algoritmo. Aprendizado de máquina têm se mostrado promissor e pode ser muito útil para coletar e analisar dados de forma a extrair conhecimento. Este conhecimento pode guiar o processo evolucionário e melhorar o desempenho de busca em meta-heurísticas, tornando-as mais inteligentes e bem-informadas. Embutir mecanismos de aprendizagem de máquina em meta-heurísticas já se mostrou promissor para melhorar a velocidade de convergência, qualidade de soluções e robustez (Talbi, 2021).

No estudo apresentado por Liang et al. (2023) os autores ressaltam, como um potencial tópico de pesquisa futura, a possibilidade da evolução ser guiada por tipo de problema. Eles argumentam que as relações entre fronteira do Pareto restrita (CPF) e irrestrita (UPF) podem ser aprendidas e então usadas para guiar os operadores de evolução subsequentes e, conseqüentemente, a capacidade de busca seria melhorada de forma eficiente.

Tendo em vista que os métodos de aprendizado de máquina e inteligência artificial, como redes neurais profundas, aprendizado por reforço, além de modelos supervisionados e não supervisionados, possuem um inquestionável poder na identificação de padrões ocultos em dados, torna-se essencial explorar o uso dessas técnicas

para extrair conhecimentos das informações geradas durante o processo evolucionário e projetar novos algoritmos capazes de tratar problemas multiobjetivos com restrições.

Além da aprendizagem de máquina, diversos outros ajustes podem ser realizados nos modelos para melhorar a convergência e a qualidade final das soluções. O ajuste dos hiperparâmetros desempenha um papel crucial, e técnicas como o ajuste de parâmetros dos algoritmos evolucionários via distribuições de probabilidade podem ser empregadas para determinar os valores para os hiperparâmetros.

3

Evolução Diferencial Bidirecional de Ordenamento por Não-dominância com Predição no Espaço Objetivo e de Decisão

Os com restrições (CMOEA) enfrentam diversos desafios ao longo do processo de otimização, sendo o equilíbrio entre convergência e diversidade um dos mais significativos. Diversos algoritmos têm tentado tratar estes desafios com o emprego de diferentes abordagens, como proposição de mecanismos para lidar com restrições, emprego de co-evolução ou mesmo inclusão de recursos emprestados de aprendizado de máquina e inteligência artificial (Liu et al., 2023; Talbi, 2021).

Nesse contexto, considerando que o comportamento de um algoritmo evolucionário pode ser entendido como aquele de um sistema estocástico, é possível utilizar um modelo de séries temporais para capturar as tendências de movimento da população ao longo das gerações (Guerrero-Peña; Araújo, 2019). Baseado nesta perspectiva, este trabalho apresenta o algoritmo Evolução Diferencial Bidirecional de Ordenamento por Não-dominância com Predição nos Espaços Objetivo e de Decisão (P-NSBiDiCo) para tratar problemas multiobjetivo com restrições.

Este capítulo detalha o algoritmo P-NSBiDiCo, algoritmo que avança o modelo NSBiDiCo (Mendes; Araújo; Farias, 2023), e que se constitui em uma nova abordagem proposta baseada no tipo de tratamento de restrições apresentado no algoritmo Co-evolução Bidirecional (BiCo) (Liu; Wang; Tang, 2022). O NSBiDiCo utiliza ordenamento por não-dominância e os operadores de evolução diferencial (Angira; Babu, 2005). O P-NSBiDiCo avança essa abordagem ao introduzir novas estratégias para a seleção dos parâmetros F e Cr da evolução diferencial, além de um mecanismo de predição no espaço objetivo e no espaço de decisão para encontrar regiões de busca promissoras.

As seções a seguir descrevem em detalhes os requisitos funcionais e não funcionais do modelo proposto, os procedimentos adotados, incluindo os pseudocódigos e as equações que fundamentam a abordagem proposta.

3.1 Requisitos de Software

Esta seção descreve os requisitos de software para a implementação do algoritmo P-NSBiDiCo, detalhando as funcionalidades e características esperadas do sistema, considerando a compatibilidade com a plataforma PlatEMO (Tian et al., 2017) e a utilização de computação distribuída e paralela. O P-NSBiDiCo é formado por componentes gerais de um algoritmo evolucionário, como inicialização, funções de avaliação, operadores de seleção, operadores de variação e critério de término. Além disso, o modelo proposto também inclui componentes específicos, tais como o tratamento de restrições, estratégia de controle de parâmetro e mecanismo de aprendizagem estatística.

3.1.1 Requisitos Funcionais

Os requisitos funcionais descrevem as funções específicas que o sistema deve executar. Para o P-NSBiDiCo, estas funções são:

1. RF1. Inicialização:

- a) **RF1.1.** O sistema deve permitir a inicialização da população principal (\mathcal{P}_0) com vetores aleatórios amostrados no espaço de busca.
- b) **RF1.2.** O sistema deve permitir a inicialização da população de arquivo (\mathcal{A}_0) como um conjunto vazio.

2. RF2. Avaliação de Soluções:

- a) **RF2.1.** O sistema deve ser capaz de avaliar as funções objetivo (f_1, f_2, \dots, f_m) para cada solução nas populações.
- b) **RF2.2.** O sistema deve ser capaz de avaliar as restrições de igualdade (g_1, g_2, \dots, g_p), e desigualdade (h_1, h_2, \dots, h_q) e calcular a violação de restrições (CV) para cada solução.

3. RF3. Ordenação por Não-Dominância:

- a) **RF3.1.** O sistema deve implementar um algoritmo de ordenação por não dominância para classificar as soluções com base em seus valores de função objetivo e violação de restrições.
- b) **RF3.2.** O sistema deve ser capaz de identificar e separar as soluções não-dominadas viáveis e inviáveis.

4. RF4. Evolução Diferencial (ED):

- a) **RF4.1.** O sistema deve implementar os operadores de mutação diferencial, cruzamento binomial e mutação polinomial, como descrito nas Equações (3.10) a (3.17).

- b) **RF4.2.** O sistema deve permitir a definição e atualização dinâmica dos parâmetros de ED, como o fator de escala (F) e a taxa de cruzamento (Cr), utilizando distribuições gaussianas com médias variáveis.

5. **RF5. Predição:**

- a) **RF5.1.** O sistema deve implementar o modelo *nonlinear autoregressive exogenous* (NARX) com Processo Gaussiano para prever valores futuros no espaço objetivo e de decisão.
- b) **RF5.2.** O sistema deve permitir a coleta e o pré-processamento dos dados de treinamento para o modelo NARX.
- c) **RF5.3.** O sistema deve ser capaz de selecionar soluções promissoras com base nas predições do modelo NARX, utilizando a distância cosseno, conforme a Equação (3.27).
- d) **RF5.4. Processamento Distribuído e Paralelo:** O treinamento do modelo NARX deve ser executado utilizando computação distribuída e paralela para acelerar o processo.

6. **RF6. Seleção e Atualização de Populações:**

- a) **RF6.1.** O sistema deve implementar o princípio de dominância de restrição (CDP) para selecionar as soluções que irão compor a nova população principal (\mathcal{P}_{t+1}).
- b) **RF6.2.** O sistema deve implementar a seleção restrita de cruzamento para a interação entre as populações principal e de arquivo.
- c) **RF6.3.** O sistema deve implementar a seleção baseada em ângulo para controlar o tamanho da população de arquivo (\mathcal{A}_{t+1}).

7. **RF7. Critério de Parada:**

- a) **RF7.1.** O sistema deve permitir a definição de um critério de parada para o algoritmo, como o número máximo de avaliações de função (maxFE) ou um tempo limite.

8. **RF8. Saída:**

- a) **RF8.1.** O sistema deve fornecer como saída o conjunto de soluções não-dominadas encontradas ao final da execução do algoritmo, juntamente com seus valores de função objetivo e violação de restrições.
- b) **RF8.2.** O sistema deve fornecer opções para visualização dos resultados, como gráficos de convergência e gráficos das soluções no espaço objetivo, utilizando as funcionalidades da plataforma PlatEMO.

3.1.2 Requisitos Não Funcionais

Os requisitos não funcionais definem características e restrições gerais do sistema, que não estão relacionadas a funcionalidades específicas. Abaixo estão listados os requisitos não funcionais do P-NSBiDiCo:

1. **RNF1. Escalabilidade:** O sistema deve ser escalável para tratar problemas multi-objetivo com diferentes números de objetivos, restrições e dimensões da variável de decisão.
2. **RNF2. Manutenibilidade:** O código fonte do sistema deve ser bem estruturado, claramente documentado e de fácil compreensão para permitir futuras manutenções e extensões.
3. **RNF3. Compatibilidade:** O sistema deve ser compatível com a plataforma PlatEMO para Matlab, seguindo os padrões da plataforma para integração e execução de algoritmos.

As seções a seguir descrevem o algoritmo P-NSBiDiCo desenvolvido de acordo com os requisitos definidos, detalhando os procedimentos, pseudo-códigos e equações empregadas para sua implementação.

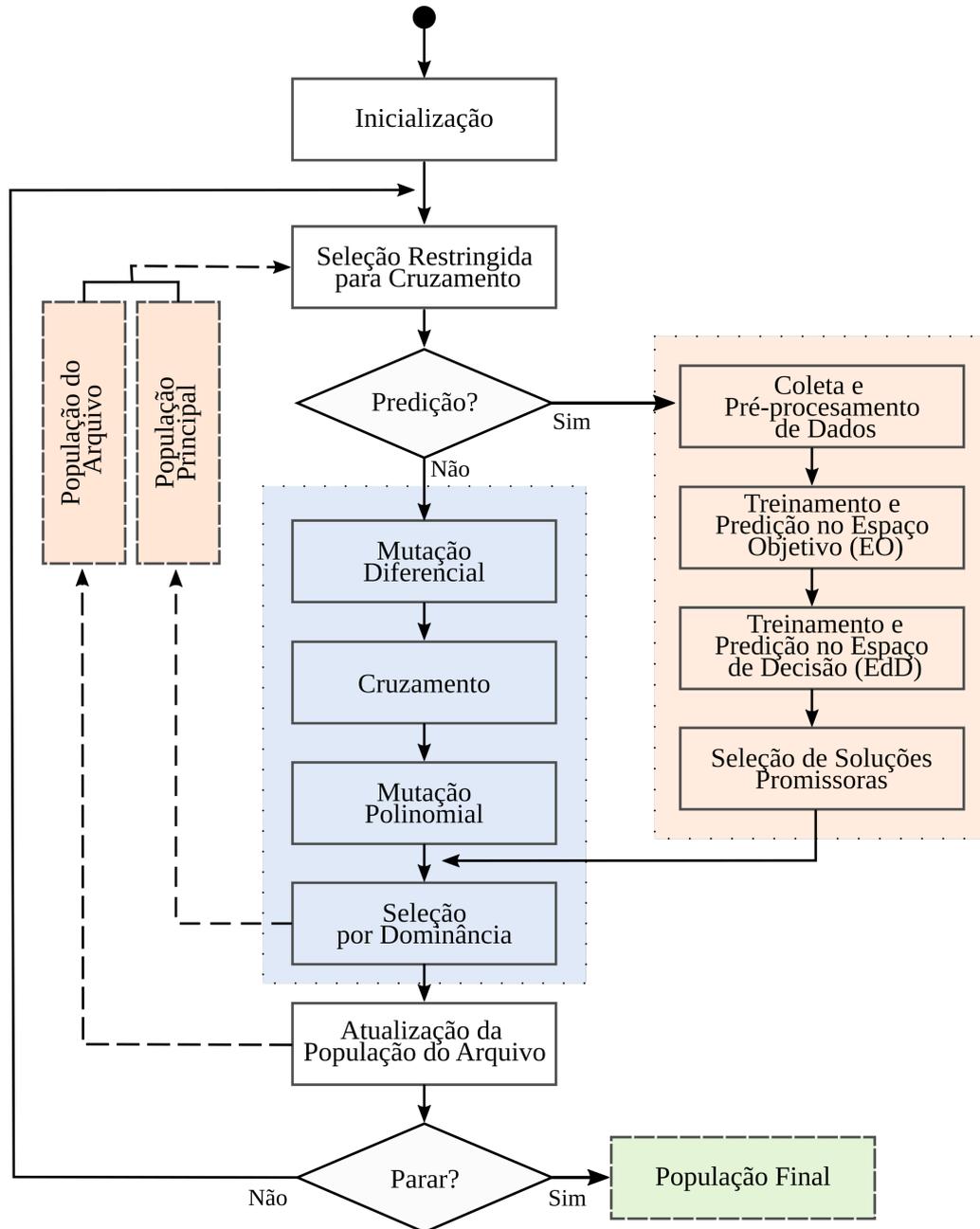
3.2 P-NSBiDiCo

O procedimento do P-NSBiDiCo é detalhado no Algoritmo 1 e tem seu fluxo de execução apresentado na Figura 2. O algoritmo recebe como parâmetros: tamanho da população N , número máximo de avaliações de função $maxFE$; parâmetros da mutação polinomial (probabilidade de mutação p_m e índice de distribuição η_m); número de vizinhos não dominados no espaço objetivo (EO) e espaço de decisão (EdD), n_{ds} e n_{os} , respectivamente; o horizonte de tempo para predições no EO e EdD, h_{ds} e h_{os} , respectivamente; e a janela de gerações para aplicação da predição ω .

O primeiro passo consiste na inicialização da população principal $\mathcal{P}_0 = \{x_1, x_2, \dots, x_N\}$ com vetores aleatórios amostrados no espaço de busca, e uma população de arquivo vazia A_0 . Em seguida, o mecanismo de seleção restringida de cruzamento escolhe soluções candidatas para gerar soluções tentativa. Posteriormente, o algoritmo executa ou a estratégia de predição ou a evolução diferencial de ordenação por não dominância (NSDE).

A estratégia para o acionamento da predição no espaço objetivo e de decisão se dá com base na fase de evolução do algoritmo, onde, na fase de exploração, o algoritmo aplica o método de predição no intervalo de ω gerações. A fase de exploração ocorre do

Figura 2 – Fluxo de Execução do Algoritmo P-NSBiDiCo.



Fonte: Autoria Própria (2024).

início até 50% do total de gerações e é definido como:

$$fase_e = [1, 0.5 \times maxGen] \quad (3.1)$$

onde $maxGen$ é o número máximo de gerações, dado por:

$$maxGen = \frac{maxFE}{N} \quad (3.2)$$

Na fase de exploração condicional o algoritmo adota uma abordagem condicional, onde a estratégia de predição só é executada caso o algoritmo ainda não tenha

convergido. A fase de exploração condicional ocorre entre 50% e 70% do total de gerações e é definida por:

$$fase_{ec} = [0.5 \times maxGen, 0.7 \times maxGen] \quad (3.3)$$

Para decidir se a predição será acionada durante a fase de exploração condicional, o algoritmo calcula o hipervolume médio HV_{μ} da população no intervalo de ω gerações anteriores conforme a Equação (3.4), e o compara com o hipervolume da população na geração anterior \mathcal{P}_{t-1} . O hipervolume médio é dado por:

$$HV_{\mu} = \frac{1}{t-p} \sum_{i=p}^{t-1} HV_{hist}(i), \quad \text{se } p > 0 \quad (3.4)$$

onde HV_{μ} é o hipervolume médio das gerações anteriores em HV_{hist} , desde p até $t-1$; Sendo $p = t - 1 - \omega$. Esse cálculo é realizado se $p > 0$, ou seja, se a janela está dentro dos limites válidos.

Caso o hipervolume de \mathcal{P}_{t-1} seja maior que o percentual $\vartheta = 0.01$ do hipervolume médio HV_{μ} , a estratégia de predição é acionada. Caso contrário, a estratégia de predição é desativada e o algoritmo segue o processo de evolução aplicando o NSDE até esgotar $maxFE$.

A estratégia de predição envolve a coleta e pré-processamento de dados, o treinamento e predição com o modelo *NARX* para prever soluções no espaço objetivo e decisão, seguido da seleção de soluções promissoras preditas para compor o conjunto de soluções tentativa.

Quando, na iteração atual, executa-se o NSDE, novas soluções tentativa são geradas por meio dos operadores de mutação diferencial e cruzamento, seguido pela mutação polinomial para produzir perturbações aleatórias.

Após isso, realiza-se a seleção ambiental, onde as soluções promissoras são atribuídas à nova população \mathcal{P}_{t+1} por meio do princípio de dominância de restrição (CDP) (Deb et al., 2002), que classifica as soluções alvo e tentativa pela relação de dominância entre elas. A atualização da população de arquivo \mathcal{A}_{t+1} ocorre através da combinação da relação de dominância e do esquema de seleção baseado em ângulo, o mesmo processo empregado no BiCo. Todos esses passos são executados em um *loop* até que o critério de parada seja satisfeito.

O algoritmo retorna o conjunto de soluções não-dominadas encontradas ao final da execução.

A seção a seguir detalha o procedimento de seleção de soluções inviáveis não-dominadas para atualização da população de arquivo \mathcal{A}_{t+1} .

Entrada: Um CMOP; tamanho da população N ; probabilidade de mutação p_m ; índice de distribuição η_m ; número máximo de avaliações de função $maxFE$; número de vizinhos não-dominados no EO n_{os} e ED n_{ds} ; o horizonte de predições no EO h_{os} e ED h_{ds} ; janela de gerações para aplicação da predição ω .

Saída : \mathcal{P}_{t+1}

```

1 Inicialize  $t = 0$ ,  $\mathcal{P}_0 \leftarrow \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , e  $\mathcal{A}_0 \leftarrow \emptyset$ ;
2  $HV_{Hist} \leftarrow \emptyset$ ;
3  $maxGen \leftarrow$  definido de acordo com a Equação (3.2);
4  $fase_e \leftarrow [1, 0.5 \times maxGen]$ ;
5  $fase_{ec} \leftarrow [0.5 \times maxGen, 0.7 \times maxGen]$ ;
6  $\vartheta \leftarrow 0.01$ ;
7  $pred \leftarrow \text{true}$ ;
8 while o critério de parada não for atendido do
9   Selecione os pais para cruzamento  $\bar{\mathcal{P}}$  de  $\mathcal{P}_t$  e  $\mathcal{A}_t$  (Alg. 3);
10  if  $p > 0$  then
11     $HV_\mu \leftarrow$  média de  $HV_{Hist}$  de acordo com a Equação (3.4);
12    /* Define se a predição será acionada */
13    if  $pred \ \&\& \ t \leq fase_e$  then
14       $pred \leftarrow \text{true}$ ;
15    else if  $pred \ \&\& \ t \leq fase_{ec} \ \&\& \ HV_{Hist}(t-1) > (\vartheta \times HV_\mu)$  then
16       $pred \leftarrow \text{true}$ ;
17    else
18       $pred \leftarrow \text{false}$ ;
19    if  $pred$  then
20      /* Aciona o mecanismo de predição. */
21      Colete e Pré-processe dados de acordo com (3.7);
22      Execute a predição no espaço objetivo e de decisão de acordo com 3.8;
23      Selecione soluções preditas promissoras de acordo com (3.10) para
24      produzir o conjunto de soluções tentativa  $\mathcal{Q}_t$ ;
25    else
26      Execute os operadores DE para produzir soluções tentativa  $\mathcal{Q}_t$  a partir de
27       $\bar{\mathcal{P}}$ ;
28  Atualize  $\mathcal{P}_{t+1}$  a partir de  $\mathcal{P}_t$  e  $\mathcal{Q}_t$  (Alg. 4);
29  Selecione as soluções inviáveis não dominadas  $\mathcal{V}_t$  de  $\mathcal{P}_t \cup \mathcal{A}_t \cup \mathcal{Q}_t$ ;
30  Execute o esquema de seleção baseado em ângulo para atualizar  $\mathcal{A}_{t+1}$  a
31  partir de  $\mathcal{V}_t$  (Alg. 2);
32   $HV_{Hist,t} \leftarrow HV(\mathcal{P}_t)$ ;
33   $t = t + 1$ ;

```

Algoritmo 1: Framework do P-NSBiDiCo

3.3 Determinação das Soluções Inviáveis Não-dominadas

As soluções inviáveis não-dominadas podem conter informações valiosas contidas nas regiões de inviabilidade que podem subsidiar a busca em direções não exploradas e promissoras. O procedimento para encontrar as soluções inviáveis não-dominadas é apresentado no fluxograma da Figura 3 e detalhado no Algoritmo 2. No geral, envolve os seguintes passos:

1. Incorporar o total de violação de restrições, CV (Equação (2.3)), como uma função objetivo adicional e converter o CMOP original definido pela Equação (2.1) em um problema sem restrições, ou seja, um MOP com $(m + 1)$ -objetivos.

$$\min \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ \dots \ f_m(\mathbf{x}), CV(\mathbf{x})]^T. \quad (3.5)$$

2. Implementar um procedimento para ordenar soluções inviáveis por não-dominância a partir da união da população principal \mathcal{P}_t , população do arquivo \mathcal{A}_t e soluções tentativas \mathcal{Q}_t com base na Equação (3.5).
3. Selecionar as soluções inviáveis não-dominadas para atualizar \mathcal{V}_t .

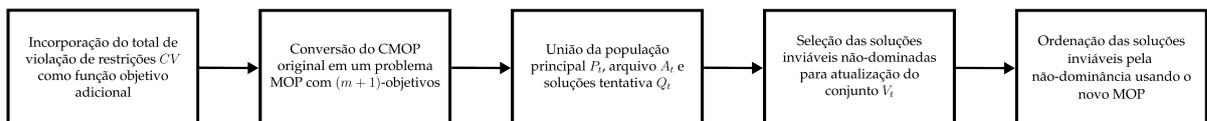
Entrada: $\mathcal{P}_t, \mathcal{A}_t$ e \mathcal{Q}_t

Saída: \mathcal{A}_{t+1}

- 1 $\mathcal{U}_t \leftarrow \mathcal{P}_t \cup \mathcal{A}_t \cup \mathcal{Q}_t$;
- 2 Selecione as soluções inviáveis não-dominadas \mathcal{V}_t em \mathcal{U}_t considerando a Equação (3.5);
- 3 **while** $\|\mathcal{V}_t\| > N$ **do**
- 4 Encontre duas soluções \mathbf{u}_i e \mathbf{u}_j com menor vetor de ângulo $(\Theta_{\mathbf{u}_i, \mathbf{u}_j})$ in \mathcal{V}_t ;
- 5 **if** $CV(\mathbf{u}_j) < CV(\mathbf{u}_i)$ **then**
- 6 $\mathcal{V}_t \leftarrow \mathcal{V}_t \setminus \mathbf{u}_i$; /* Delete \mathbf{u}_i */
- 7 **else**
- 8 $\mathcal{V}_t \leftarrow \mathcal{V}_t \setminus \mathbf{u}_j$; /* Delete \mathbf{u}_j */
- 9 $\mathcal{A}_{t+1} \leftarrow \mathcal{V}_t$;

Algoritmo 2: Atualização da População do Arquivo

Figura 3 – Fluxograma para a seleção de soluções inviáveis não dominadas.



Fonte: Autoria Própria (2024).

3.4 Seleção Baseada em Ângulo

Em casos onde a quantidade de soluções em \mathcal{V}_t é maior que o tamanho do arquivo N , então um esquema de seleção baseado em ângulo para preservação da diversidade é empregado para descartar soluções excedentes.

O esquema de seleção baseado em ângulo foca no ângulo entre duas soluções do espaço objetivo normalizado, sem considerar o conjunto de restrições. O vetor de ângulos é calculado através de ambas as estimações de ponto ideal $\mathbf{Z}^{min} = [z_1^{min}, z_2^{min}, \dots, z_m^{min}]^T$ e ponto nadir $\mathbf{Z}^{max} = [z_1^{max}, z_2^{max}, \dots, z_m^{max}]$, onde z_i^{min} e z_i^{max} representam os valores mínimo e máximo do i -ésimo objetivo para todas as soluções em \mathcal{V}_t . Após isso, o vetor $\mathbf{F}(\mathbf{v}_j)$ é normalizado como $\mathbf{F}'(\mathbf{v}_j) = [f'_1(\mathbf{v}_j), f'_2(\mathbf{v}_j), \dots, f'_m(\mathbf{v}_j)]^T$ de acordo com

$$f'_i(\mathbf{v}_j) = \frac{z_i^{max} - f_i(\mathbf{v}_j)}{z_i^{max} - z_i^{min}}, i = 1, 2, \dots, m. \quad (3.6)$$

Em seguida, o vetor de ângulos θ entre duas soluções \mathbf{v}_j e \mathbf{v}_k é determinado de acordo com a equação:

$$\theta_{\mathbf{v}_j, \mathbf{v}_k} = \arccos \left| \frac{\mathbf{F}'(\mathbf{v}_j) \bullet \mathbf{F}'(\mathbf{v}_k)}{\|\mathbf{F}'(\mathbf{v}_j)\| \cdot \|\mathbf{F}'(\mathbf{v}_k)\|} \right| \quad (3.7)$$

onde $\mathbf{F}'(\mathbf{v}_j) \bullet \mathbf{F}'(\mathbf{v}_k)$ representa o produto interno entre $\mathbf{F}'(\mathbf{v}_j)$ e $\mathbf{F}'(\mathbf{v}_k)$, e $\|\cdot\|$ denota a norma do vetor. Como destacado em Liu, Wang and Tang (2022), o ângulo entre as soluções estima a dissimilaridade entre os vetores, ou seja, mede o grau de diferença entre os vetores.

Além de considerar a diversidade, soluções com pior avaliação são removidas, uma a uma, da população do arquivo \mathcal{A}_t . Isto é, duas soluções em \mathcal{V}_t com o menor valor do ângulo entre elas são identificadas, o ângulo indica que estas possuem direção de busca semelhantes e, devido a isso, o processo para deletar uma delas leva em consideração o total de violação de restrições CV . Portanto, a solução que possuir maior valor de CV é removida de \mathcal{A}_t . Este processo é executado de forma iterativa enquanto o tamanho de \mathcal{A}_t exceder N .

A interação entre a população principal \mathcal{P} e a população do arquivo \mathcal{A} atua como um mecanismo para compartilhamento de informações e é executado através da seleção restringida de cruzamento proposta para o algoritmo BiCo. No procedimento, descrito no Algoritmo 3, as seguintes condições devem ser satisfeitas.

1. Se o tamanho da população do arquivo for menor que N , os pais para cruzamento \mathbf{p}_1 e \mathbf{p}_2 são aleatoriamente selecionados a partir da união de \mathcal{P}_t e \mathcal{A}_t .
2. Caso contrário, \mathbf{p}_1 e \mathbf{p}_2 são selecionados baseados em CV e o valor de diversidade angular (i.e., AD), respectivamente. Durante este passo, para selecionar a primeira

```

Entrada:  $\mathcal{P}_t$  e  $\mathcal{A}_t$ 
Saída : Pais para Cruzamento  $\mathbf{p}_1$  e  $\mathbf{p}_2$ 
1 if  $\|\mathcal{A}_t\| < N$  then
2 | Seleccione aleatoriamente duas soluções  $\mathbf{p}_1$  e  $\mathbf{p}_2$  de  $\mathcal{P}_t \cup \mathcal{A}_t$ ;
3 else
4 | Seleccione aleatoriamente duas soluções  $\mathbf{x}_1$  e  $\mathbf{a}_1$  de  $\mathcal{P}_t \cup \mathcal{A}_t$ , respectivamente;
5 | if  $CV(\mathbf{x}_1) \leq CV(\mathbf{a}_1)$  then
6 | |  $\mathbf{p}_1 \leftarrow \mathbf{x}_1$ ;
7 | else
8 | |  $\mathbf{p}_1 \leftarrow \mathbf{a}_1$ ;
9 | Seleccione aleatoriamente duas soluções  $\mathbf{x}_2$  e  $\mathbf{a}_2$  de  $\mathcal{P}_t \cup \mathcal{A}_t$ , respectivamente;
10 | if  $AD(\mathbf{x}_2) \leq AD(\mathbf{a}_2)$  then
11 | |  $\mathbf{p}_2 \leftarrow \mathbf{x}_2$ ;
12 | else
13 | |  $\mathbf{p}_2 \leftarrow \mathbf{a}_2$ ;

```

Algoritmo 3: Conjunto de Cruzamento Restrito

solução \mathbf{p}_1 , duas soluções \mathbf{x}_1 e \mathbf{a}_1 são sorteadas de \mathcal{P}_t e \mathcal{A}_t , respectivamente. Em seguida, a solução que possuir menor valor de CV é escolhida. Após isso, para selecionar \mathbf{p}_2 , duas soluções \mathbf{x}_2 e \mathbf{a}_2 são sorteadas de \mathcal{P}_t e \mathcal{A}_t , respectivamente. Os valores de AD destas soluções são comparados e, dentre estas, a que possuir maior valor de AD é selecionada.

O cálculo de $AD(\mathbf{x})$ para a j -ésima solução \mathbf{x}_j em \mathcal{P}_t , inclui os seguintes passos.

1. Identificar o ponto ideal $\bar{\mathbf{z}}^{min} = [\bar{z}_1^{min} \ \bar{z}_2^{min} \ \dots \ \bar{z}_m^{min}]^T$ e estimar o ponto nadir $\bar{\mathbf{z}}^{max} = [\bar{z}_1^{max}, \bar{z}_2^{max}, \dots, \bar{z}_m^{max}]^T$, onde \bar{z}_i^{min} e \bar{z}_i^{max} representam os valores máximos e mínimos do i -ésimo objetivo para todas as soluções da união das populações \mathcal{P}_t e \mathcal{A}_t , respectivamente.

2. Normalizar o vetor objetivo $\mathbf{f}(\mathbf{x}_j)$ da j -ésima solução com

$$\mathbf{f}^*(\mathbf{x}_j) = [f_1^*(\mathbf{x}_j) \ f_2^*(\mathbf{x}_j) \ \dots \ f_m^*(\mathbf{x}_j)]^T$$

de acordo com

$$f_i^*(\mathbf{x}_j) = \frac{f_i(\mathbf{x}_j) - \bar{z}_i^{min}}{\bar{z}_i^{max} - \bar{z}_i^{min}}, i = 1, 2, \dots, m. \quad (3.8)$$

3. Calcular os ângulos entre os vetores \mathbf{x}_j e \mathbf{x}_k em \mathcal{P}_t de acordo com

$$\theta'_{\mathbf{x}_j, \mathbf{x}_k} = \arccos \left| \frac{\mathbf{f}^*(\mathbf{x}_j) \bullet \mathbf{f}^*(\mathbf{x}_k)}{\|\mathbf{f}^*(\mathbf{x}_j)\| \cdot \|\mathbf{f}^*(\mathbf{x}_k)\|} \right| \quad (3.9)$$

$$\mathbf{x}_k \in \mathcal{P}_t \cap \mathbf{x}_k \neq \mathbf{x}_j$$

4. Atribuir $AD(\mathbf{x}_j)$ como o k -ésimo valor mínimo no conjunto de $\{\theta'_{\mathbf{x}_j, \mathbf{x}_k} \in \mathcal{P}_t \cap \mathbf{x}_k \neq \mathbf{x}_j\}$, onde k é definido como \sqrt{N} .

É preferível um alto valor de AD para \mathbf{x}_j pois seu cálculo estabelece a diversidade calculada pelo ângulo de $AD(\mathbf{a}_j) \in \mathcal{A}_t$.

A próxima seção discute o algoritmo de busca que emprega evolução diferencial de ordenação por não dominância (NSDE) com um novo mecanismo de seleção dos parâmetros F e Cr .

3.5 Algoritmo de Busca

Após a construção do conjunto de soluções candidatas $\overline{\mathcal{P}}_t$, os operadores de evolução diferencial (ED) de cruzamento e mutação diferencial são aplicados para gerar um novo conjunto de soluções tentativa \mathcal{Q}_t . Para evidenciar a capacidade de exploração e exploração do algoritmo, utilizou-se o $DE/current - to - rand/1$, definido de acordo com a Equação (3.10) a seguir.

$$\mathbf{v}_i = \mathbf{x}_i + F \times (\mathbf{x}_{r_1} - \mathbf{x}_i + \mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (3.10)$$

onde \mathbf{x}_i é o vetor alvo; \mathbf{x}_{r_1} , \mathbf{x}_{r_2} e \mathbf{x}_{r_3} são vetores selecionados aleatoriamente para construir o vetor mutante \mathbf{v}_i , onde $i \neq r_1 \neq r_2 \neq r_3$; $\{r_1, r_2, r_3\} \in [1, N]$; e F é um fator de escala em que $F \in [0, 1]$ que é amostrado como descrito na Subseção 3.6.

Após a realização do procedimento de mutação, a estratégia de cruzamento binomial é então aplicada. O operador de cruzamento cria um conjunto de vetores tentativa ($\mathbf{u}_i \in \mathcal{U}_t$) construído a partir dos vetores alvo \mathbf{x}_i e mutado \mathbf{v}_i :

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{se } (rand < CR), \\ x_{i,j}, & \text{caso contrário.} \end{cases}, j = 1, 2, \dots, D. \quad (3.11)$$

onde $rand$ representa um valor aleatório uniforme amostrado no intervalo $[0, 1]$; CR é um número real entre 0 e 1, amostrado tal como descrito na Subseção 3.6; D é a dimensão do vetor solução; e $u_{i,j}$ é o j -ésimo elemento do i -ésimo novo vetor tentativa. Em seguida, a mutação polinomial é executada (Deb; Goyal, 1996) para perturbar as soluções em \mathcal{U}_t com probabilidade de mutação p_m . Este procedimento gera um vetor tentativa mutado. Para uma dada solução $\mathbf{u}_i \in \mathcal{U}_t$:

$$\mathbf{t}_i = \mathbf{u}_i + (\mathbf{u}_{B_u,i} - \mathbf{u}_{B_l,i}) \times \delta, \quad (3.12)$$

onde B_l e B_u são os limites superior e inferior de cada componente da i -ésima variável de decisão; a distribuição de probabilidade de δ é

$$p(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{\eta_m}$$

onde η_m é o índice da distribuição e determina o formato da distribuição; δ pode ser computado da seguinte maneira:

$$\delta(\rho) = \begin{cases} (2\rho)^{\frac{1}{(\eta_m+1)}} - 1, & \text{se } \rho \leq 0.5 \\ 1 - [2(1 - \rho)^{\frac{1}{(\eta_m+1)}}], & \text{caso contrário.} \end{cases} \quad (3.13)$$

onde ρ é um número aleatório uniforme amostrado no intervalo 0 e 1.

3.6 Amostragem dos Parâmetros F e Cr

O algoritmo evolução diferencial têm como principais parâmetros F e Cr , que determinam a relação entre exploração e exploração durante o processo de busca afetando diretamente o desempenho geral do algoritmo. De modo geral, pode-se privilegiar a exploração durante os estágios inicial e intermediário da busca e ênfase na exploração durante o estágio final. Estas premissas foram caracterizadas por uma abordagem baseada em distribuições de probabilidade com média variável ao longo do processo evolucionário para a seleção destes parâmetros.

3.6.1 Amostragem de F

Para a amostragem do parâmetro F , emprega-se uma distribuição Gaussiana $\mathcal{N}(\mu, \sigma^2)$ com média μ que varia dinamicamente ao longo das gerações. Essa variação foi projetada para explorar o espaço de busca de forma ampla nas gerações iniciais e, gradualmente, refinar as soluções à medida que o processo evolucionário avança. Esta abordagem visa atender à necessidade de transitar entre exploração e exploração de forma controlada, garantindo a eficiência do algoritmo em suas diferentes fases.

A variação de μ segue um padrão de decaimento sigmoide, descrito pela Equação (3.14).

$$\mu(t) = \mu_f + \frac{\mu_0 - \mu_f}{1 + \exp(-\lambda \times (t - \xi_t))} \quad (3.14)$$

onde $\mu(t)$ é o valor da média μ na geração t ; μ_f é o valor médio final desejado; μ_0 é o valor médio inicial; λ é o fator que controla a inclinação da curva da função sigmoide (valor positivo); e ξ_t representa o ponto de inflexão da função sigmoide, geralmente situado em torno de $G/2$, onde G é o número total de gerações.

Essa modelagem sigmoide é crucial para ajustar a intensidade das mutações ao longo das gerações. Valores iniciais mais altos de μ_t incentivam mudanças significativas no espaço de decisão, promovendo maior exploração. Conforme t aumenta, μ_t decresce, promovendo alterações refinadas e, conseqüentemente, maior exploração.

Após determinar $\mu(t)$, os valores de F são amostrados de acordo com a nova distribuição de probabilidade para cada indivíduo i na população $\bar{\mathcal{P}}_t$, de acordo com:

$$F_i \sim \mathcal{N}(\mu(t), \sigma^2), \quad \forall i \in \{1, 2, \dots, N\} \quad (3.15)$$

onde N é o tamanho da população em $\bar{\mathcal{P}}_t$.

A faixa de valores do parâmetro λ , que controla a inclinação da curva sigmoideal, foi definida com base em experimentos práticos, sendo ajustada empiricamente para alcançar um padrão desejado de comportamento ao longo do tempo. Os valores específicos adotados para λ variam gradualmente de 0.95 a 0.15, de acordo com o número máximo de avaliações de função ($\max FE$) disponíveis e são descritos na Equação (3.16).

$$\lambda = \begin{cases} 0.0001, & \text{se } \max FE \leq 50000 \\ 5 \times 10^{-5}, & \text{se } \max FE \leq 100000 \\ 3 \times 10^{-5}, & \text{se } \max FE \leq 200000 \\ 2 \times 10^{-5}, & \text{caso contrário.} \end{cases} \quad (3.16)$$

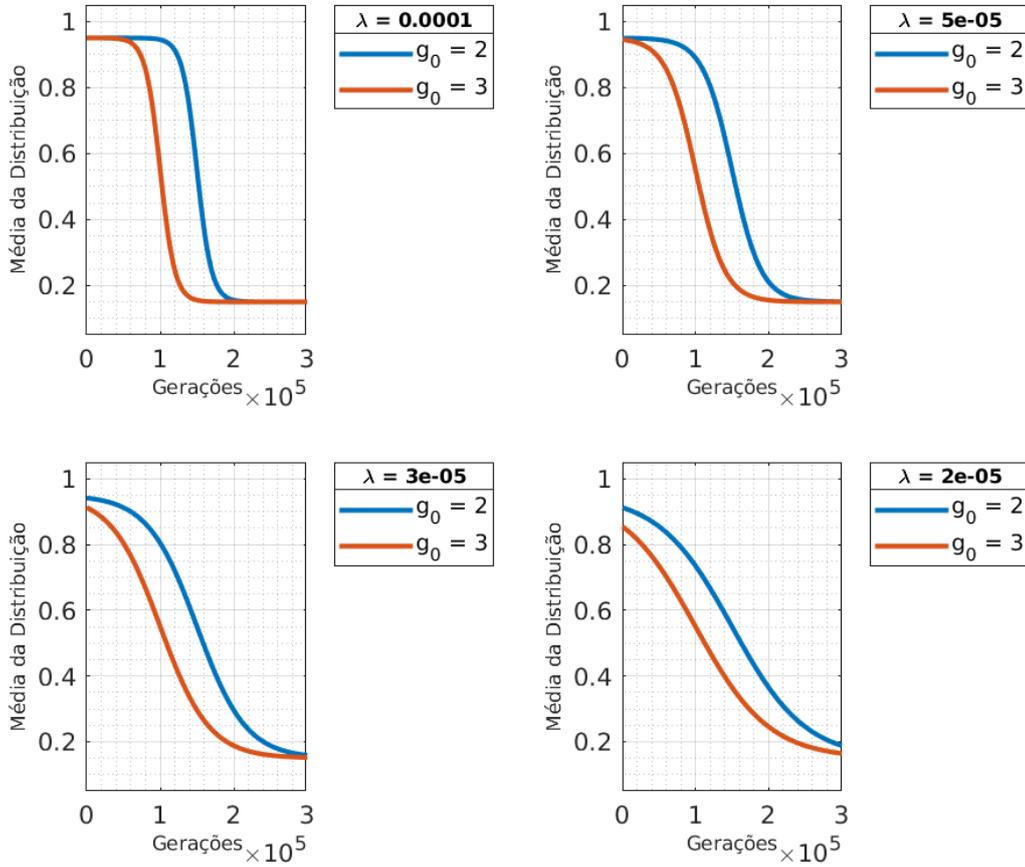
A Figura 4 ilustra como diferentes valores de λ (Equação (3.14)) afetam a transição $\mu(t)$ ao longo das gerações t , destacando sua influência na capacidade do algoritmo de ajustar o equilíbrio entre exploração e exploração. As ilustrações de Figura 4 consideram os diferentes valores de λ ; pontos de inflexão $g_0 = 2$ e $g_0 = 3$; e com média inicial $\mu_f = 0.95$ e a média final $\mu_0 = 0.15$.

3.6.2 Amostragem de Cr

Além de F , outro parâmetro crucial no controle do comportamento do ED é Cr , que regula o processo de cruzamento, ajustando a troca de informações entre as soluções candidatas. O parâmetro Cr determina a probabilidade de que elementos do vetor tentativa u_i sejam copiados do vetor mutado v_i ou do vetor alvo x_i . Dessa forma, com valores altos de Cr há maior probabilidade de u_i obter mais informações do vetor mutado v_i , favorecendo a exploração do espaço de busca e a troca de informações entre as soluções. Quando setado com valores mais baixos, Cr proporciona maiores chances de que u_i mantenha mais características da solução alvo x_i , permitindo que o algoritmo enfatize exploração, ou seja, que refine a busca das soluções dentro de uma região promissora dentro do espaço de busca.

Visando manter o equilíbrio entre exploração e exploração durante o processo de busca do algoritmo, foram utilizadas duas distribuições Gaussianas distintas para amostrar os valores de Cr . Uma distribuição possui média $\mu = 0.9$, promovendo maior exploração ao incorporar mais elementos do vetor mutado. A outra distribuição possui média $\mu = 0.1$, favorecendo exploração ao manter mais características do vetor alvo. A

Figura 4 – Curvas de mudança da média μ ao longo das gerações para diferentes valores de λ .



Fonte: Autoria Própria (2024).

escolha entre essas duas distribuições é determinada por uma probabilidade p a cada iteração i no conjunto $\bar{\mathcal{P}}_t$ definido pela Equação (3.17).

$$Cr_i \sim \begin{cases} \mathcal{N}(0.9, \sigma^2), & \text{se } p > 0.5 \\ \mathcal{N}(0.1, \sigma^2), & \text{se } p \leq 0.5 \end{cases}, \quad \forall i \in \{1, 2, \dots, N\} \quad (3.17)$$

onde $p \in [0, 1]$ é um valor aleatoriamente selecionado de acordo com uma distribuição uniforme; N é o tamanho da população em $\bar{\mathcal{P}}_t$.

Quando o mecanismo de predição é acionado, uma nova estratégia de busca é empregada, onde o algoritmo utiliza o modelo *NARX* para prever soluções promissoras no espaço objetivo e de decisão. A próxima seção detalha o procedimento inicial de coleta e pré-processamento de dados para treinamento e predição com o modelo *NARX*.

3.7 Coleta e Pré-processamento de Dados

O procedimento de coleta e pré-processamento de dados envolve selecionar soluções viáveis e inviáveis não-dominadas para compor o conjunto \mathcal{ND}_t a partir da união da população principal e população do arquivo, ou seja, $\mathcal{ND}_t \leftarrow NonDominated(\mathcal{P}_t \cup \mathcal{A}_t)$. Caso $|\mathcal{ND}_t| - |\mathcal{P}_t| < N$, então \mathcal{ND}_t é preenchido com $N - |\mathcal{ND}_t|$ soluções selecionadas aleatoriamente de \mathcal{P}_t . Em seguida, os valores objetivo de cada solução em \mathcal{ND}_t é normalizado de acordo com a Equação (3.8). Após isso, uma rotina é executada para selecionar recursivamente κ soluções anteriores mais próximas com base na distância Euclidiana para cada solução $\mathbf{x} \in \mathcal{ND}_t$, tal como definido pela Equação (3.18)

$$C[i, t] = \arg \min_{\mathbf{x} \in \mathcal{ND}_{t-g}} \left\{ \sqrt{\sum_{j=1}^m (x_{i,j}^* - x_j)^2} \right\} \quad (3.18)$$

onde $i = 1, 2, \dots, N$; $g = 1, 2, \dots, \kappa$; \mathbf{x}_i^* é a i -ésima solução em \mathcal{ND}_t e \mathcal{ND}_{t-g} é o conjunto de soluções não-dominadas da g -ésima geração passada.

Esta etapa produz o conjunto de dados de treinamento \mathcal{D} , que é composto pelos dados de treinamento do espaço objetivo \mathcal{D}_{obj} e espaço de decisão \mathcal{D}_{dec} .

$$\mathcal{D}_{obj} = \begin{bmatrix} f_1^1(\mathbf{x}) & f_2^1(\mathbf{x}) & \dots & f_m^1(\mathbf{x}) \\ f_1^2(\mathbf{x}) & f_2^2(\mathbf{x}) & \dots & f_m^2(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ f_1^\lambda(\mathbf{x}) & f_2^\lambda(\mathbf{x}) & \dots & f_m^\lambda(\mathbf{x}) \end{bmatrix}$$

$$\mathcal{D}_{dec} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^\lambda & x_2^\lambda & \dots & x_d^\lambda \end{bmatrix}$$

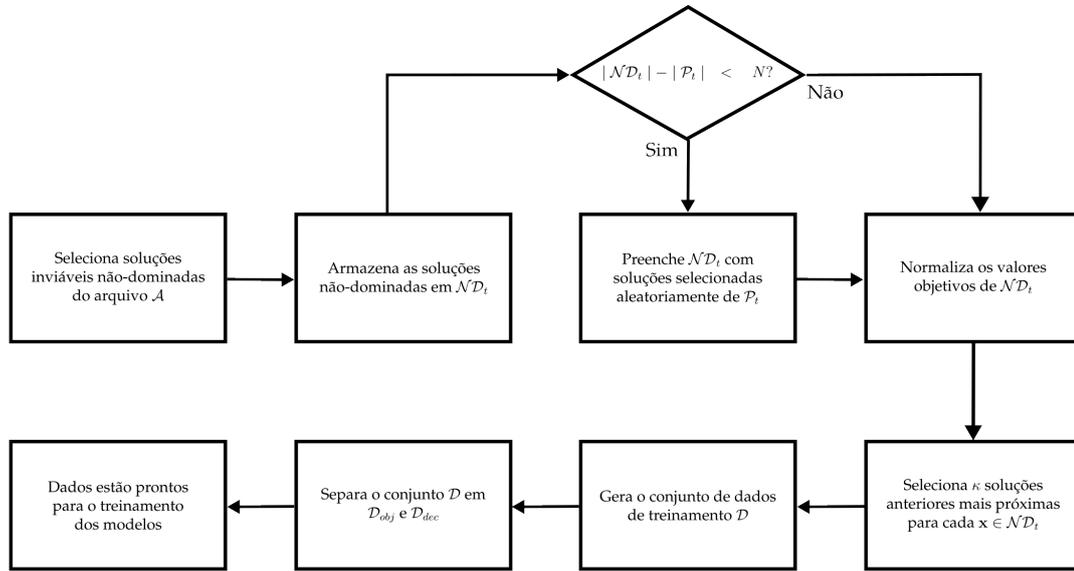
Após a definição do conjunto de dados, estes são normalizados de acordo com

$$z = \frac{x - \mu}{\sigma} \quad (3.19)$$

onde z é o valor normalizado; x é o valor original; μ é a média dos valores originais; e σ é o desvio padrão dos valores originais.

Para resumir o procedimento de pré-processamento de dados descrito nesta seção, a Figura 5 apresenta um fluxograma detalhando as etapas realizadas para preparar o conjunto de dados para o treinamento dos modelos. Este conjunto de dados, devidamente estruturado e normalizado, será utilizado na próxima etapa, conforme discutido na Seção 3.8.

Figura 5 – Fluxograma do pré-processamento de dados para o treinamento dos modelos NARX.



Fonte: Autoria Própria (2024).

3.8 Predição com *Non-linear AutoRegressive eXogenous (NARX)*

O processo de evolução de um algoritmo evolucionário (EA) pode ser descrito como um sistema dinâmico representado por uma coleção de indivíduos da população \mathcal{P}_t que é tomada sequencialmente ao longo do tempo t , evolui conforme premissas de movimento (Guerrero-Peña; Araújo, 2019). O modelo NARX é capaz de capturar essa dinâmica, possibilitando a predição de valores futuros de uma dada série (Ljung, 1999; Guerrero-Peña; Araújo, 2019). Este modelo envolve os regressores das variáveis de entrada e saída no momento t denotado como

$$y(t) = f(\mathbf{y}_t, \mathbf{u}_t) + e(t), \quad (3.20)$$

considerando:

$$\mathbf{y}_t = \left[y(t-1) \quad y(t-2) \quad \cdots \quad y(t-n_a) \right]^T, \quad \mathbf{u}_t = \left[u(t-1) \quad u(t-2) \quad \cdots \quad u(t-n_b) \right]^T,$$

onde f é uma função não linear; \mathbf{u}_t e \mathbf{y}_t são os vetores de entrada e saída do modelo, respectivamente. Os parâmetros n_a e n_b são as ordens do modelo NARX, e $e(t)$ é o termo de ruído branco de média zero. Neste estudo, os parâmetros do NARX foram mantidos com valores padrão, justificando-se por sua simplicidade e adequação a cenários iniciais. Essa escolha busca evitar a complexidade excessiva em situações onde interações ou variáveis externas não têm impacto significativo. Além disso, funciona como um

ponto de partida simples e eficiente para entender o comportamento autoregressivo básico antes de aplicar ajustes mais avançados.

Para capturar comportamentos não lineares complexos, a função não linear f é modelada como um Processo Gaussiano (GP). Esta abordagem permite a modelagem flexível de relações não lineares, sem pressupor uma forma funcional específica.

A entrada do processo Gaussiano consiste no vetor de regressão $\varphi(t)$, e a saída é correspondente à previsão do modelo:

$$\hat{y}(t) = \mathcal{GP}(\varphi(t)) + e(t), \quad (3.21)$$

onde $\mathcal{GP}(\varphi(t))$ denota a realização de um processo Gaussiano com entrada $\varphi(t)$. O vetor de regressão $\varphi(t)$ é definido como:

$$\varphi(t) = \left[-y(t-1) \quad \dots \quad -y(t-n_a) \quad u(t-n_t) \quad \dots \quad u(t-n_t-n_b+1) \right]^T, \quad (3.22)$$

onde $y(t-1), \dots, y(t-n_a)$ representam os valores passados da saída do sistema até a ordem n_a ; e $u(t-n_t), \dots, u(t-n_t-n_b+1)$ representam os valores passados da entrada do sistema, começando de um atraso n_t até a ordem n_b .

Os processos Gaussianos se caracterizam por seus hiperparâmetros, que definem a função de covariância $k(x, x')$. A estimativa desses hiperparâmetros é crucial para a eficácia do modelo.

Para realizar previsões usando processos Gaussianos, assume-se que as observações seguem uma distribuição normal multivariada, ou seja:

$$y \sim \mathcal{N}(\mu, \mathbf{K}) \quad (3.23)$$

onde y é um vetor de observações, μ um vetor de médias e \mathbf{K} é a matriz de covariância entre os dados de treinamento.

No contexto do algoritmo P-NSBiDiCo, são realizadas duas predições: uma no espaço objetivo e outra no espaço de decisão.

Cada uma dessas predições é realizada para um horizonte de predição específico, sendo h_{os} e h_{ds} os horizontes de predição para os espaços objetivo ($\hat{\mathbf{y}}_{os}(t_*)$) e de decisão ($\hat{\mathbf{y}}_{ds}(t_*)$), respectivamente. Uma nova entrada $\varphi_{os}(t_*)$ ou $\varphi_{ds}(t_*)$ corresponde a $t + h$, onde h é o horizonte de predição. Assim, a predição para o espaço objetivo é dada por:

$$\hat{\mathbf{y}}_{os}(t_*) = \mathbf{k}_{os*}^T \mathbf{K}_{os}^{-1} \mathbf{y}, \quad (3.24)$$

e a predição para o espaço de decisão é dada por:

$$\hat{\mathbf{y}}_{ds}(t_*) = \mathbf{k}_{ds*}^T \mathbf{K}_{ds}^{-1} \mathbf{y}, \quad (3.25)$$

onde \mathbf{k}_*^T é o vetor transposto de covariância entre a nova entrada $\varphi(t_*)$ e os dados de treinamento; \mathbf{K}^{-1} é a inversa da matriz de covariância do conjunto de treinamento para cada espaço.

Dado que os valores do conjunto de treinamento foram normalizados de acordo com a equação (3.19), os novos valores preditos, $\hat{y}_{os}(t_*)$ e $\hat{y}_{ds}(t_*)$, são transformados de volta para a escala original utilizando a média e o desvio padrão dos valores de treinamento de acordo com a Equação (3.26).

$$\mathbf{x} = \mathbf{z} \cdot \boldsymbol{\sigma} + \boldsymbol{\mu}, \quad (3.26)$$

onde \mathbf{x} é o valor na escala original; \mathbf{z} é o valor na escala padronizada; $\boldsymbol{\mu}$ é a média dos valores originais; e $\boldsymbol{\sigma}$ é o desvio padrão dos valores originais.

Após o treinamento e predição de novas soluções, o algoritmo P-NSBiDiCo emprega um procedimento de seleção de soluções promissoras para identificar soluções candidatas que serão utilizadas para a geração de novas soluções tentativas. A próxima seção detalha o procedimento de seleção de soluções promissoras.

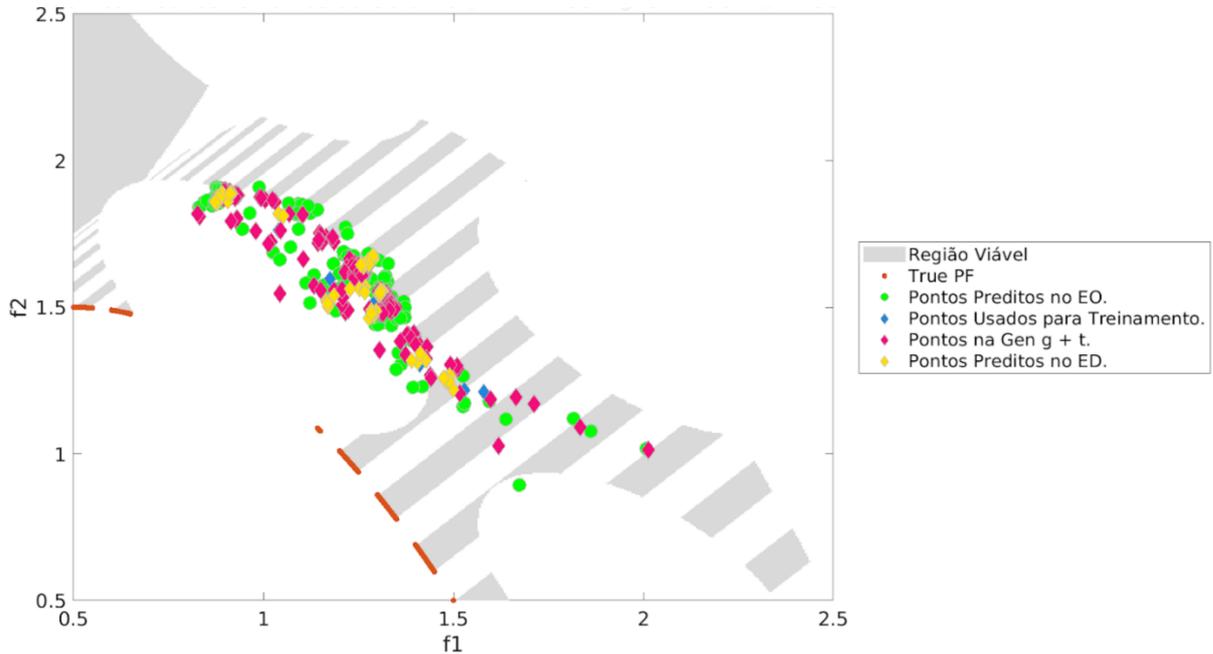
3.9 Exemplos de Funcionamento da Predição

Para ilustrar o funcionamento do modelo preditivo, a Figura 6 e a Figura 7 apresentam dois exemplos: um com predição no espaço objetivo e outro no espaço de decisão, respectivamente. Cada exemplo é acompanhado de figuras que mostram o conjunto de pontos (soluções) utilizados para treinar o modelo em uma geração g , os pontos preditos para uma geração futura $g + t$, e a população observada na mesma geração $g + t$. Esses exemplos têm como objetivo demonstrar a capacidade do modelo de capturar padrões e prever soluções em ambos os espaços, destacando a relação entre as soluções treinadas, preditas e observadas.

O problema DASCMP4 possui uma Fronteira de Pareto (PF) com geometria descontínua e poucas regiões viáveis, o que torna sua otimização especialmente complexa. No exemplo da Figura 6, a predição foi realizada na geração 541, com um horizonte de predição de 10 gerações. Observa-se que o modelo conseguiu capturar as tendências e, após 10 gerações, a população está próxima das regiões preditas. Algumas dessas soluções preditas apresentaram maior capacidade de explorar novas áreas de busca, evidenciando o potencial do modelo para gerar soluções exploratórias e promover a diversidade.

Já o DOC1 possui uma Fronteira de Pareto côncava e contínua, além de uma região de inviabilidade não linear. Este problema foi escolhido devido ao seu menor número de dimensões, o que facilita a visualização gráfica. A predição no espaço de decisão é particularmente desafiadora, especialmente em problemas de alta dimensio-

Figura 6 – Conjunto de treinamento, previsões e população observada no espaço objetivo para DASCPOP4



Fonte: Autoria Própria (2024).

nalidade. No entanto, como ilustrado na Figura 7, observa-se que as soluções preditas no espaço de decisão são muito próximas dos valores observados na população futura, após transcorrida a janela de predição.

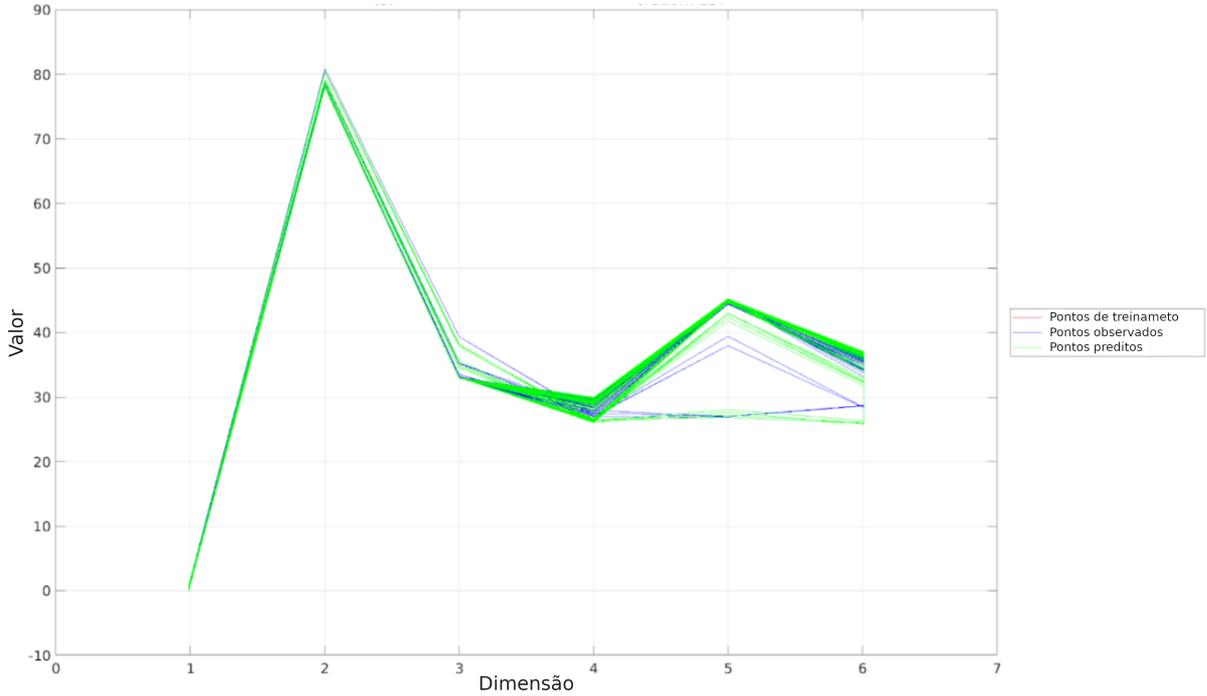
Os exemplos apresentados permitem observar como o modelo preditivo se aproxima da população observada em $g + t$, fornecendo *insights* valiosos sobre a precisão das previsões. Além disso, eles validam visualmente a eficácia do modelo preditivo ao mapear soluções de treinamento para previsões futuras e compará-las com a população real observada, destacando sua capacidade de lidar com diferentes estruturas de problemas e espaços de busca.

3.10 Seleção de Soluções Promissoras Preditas

A estimação dos modelos e a predição para cada solução em \mathcal{D} resultam em um conjunto de valores preditos para o espaço objetivo $\tilde{\mathcal{Y}}_{obj}$ e para o espaço de decisão \mathcal{Y}_{dec} .

As soluções preditas no espaço de decisão \mathcal{Y}_{dec} são mapeadas para o espaço objetivo através das funções objetivo $f : \mathcal{Y}_{dec} \rightarrow \tilde{\mathcal{Y}}_{obj}$. Em seguida, os valores objetivo de \mathcal{Y}_{dec} e $\tilde{\mathcal{Y}}_{obj}$ são normalizados de acordo com a Equação (3.8).

Figura 7 – Conjunto de treinamento, predições e população observada no espaço de decisão para DOC1



Fonte: Autoria Própria (2024).

As soluções preditas no espaço objetivo $\mathcal{Y}_{obj,i}$ são selecionadas as 5 mais próximas em \mathcal{Y}_{dec} com base na distância cosseno para compor \mathcal{C} conforme Equação (3.27),

$$\mathcal{C}_i = \arg \min_{\mathcal{Y}_{dec} \in \mathcal{Y}_{dec}} \left(1 - \frac{\mathcal{Y}_{dec} \cdot \hat{\mathcal{Y}}_{obj,i}}{\|\mathcal{Y}_{dec}\| \|\hat{\mathcal{Y}}_{obj,i}\|} \right) \quad (3.27)$$

onde \mathcal{C}_i representa o conjunto dos 5 índices das soluções mais próximas. Após isso, uma única solução é aleatoriamente selecionada do conjunto \mathcal{C}_i para que seja atribuída a \mathcal{C} . Formalmente, esta seleção pode ser dada por

$$\mathcal{C}_i^* = \mathcal{C}_i(\text{random}([1, \dots, 5])) \quad (3.28)$$

onde \mathcal{C}_i^* é o índice da solução selecionada aleatoriamente. Em seguida, o conjunto \mathcal{C} é atualizado para conter apenas os índices únicos:

$$\mathcal{C} \leftarrow \text{unique}(\mathcal{C}). \quad (3.29)$$

onde $\text{unique}(\mathcal{C})$ refere-se à operação de atualização do conjunto \mathcal{C} para que contenha apenas os índices únicos das soluções selecionadas.

Selecionar 1 entre as 5 soluções mais próximas via métrica cosseno visa promover maior diversidade entre as soluções selecionadas, pois as soluções mais próximas possuem direção de busca semelhante, dessa forma possibilitando explorar com maior eficácia o espaço de busca.

Com o conjunto \mathcal{C} de soluções promissoras definido, novas soluções são geradas com o intuito de promover a troca de informações entre soluções preditas e a soluções da população principal. Para isso, os seguintes passos são realizados:

1. Para cada solução em \mathcal{P} e \mathcal{C} , normalize os valores objetivos de acordo com a Equação (3.30).
2. Para cada solução $\mathbf{x}_i \in \mathcal{P}$, identifica-se a solução mais próxima \mathbf{y}_i^* em \mathcal{C} usando a distância cosseno dado pela Equação 3.31.
3. Um fator aleatório α é amostrado uniformemente no intervalo $[0, 1]$.
4. Uma nova solução tentativa \mathbf{z}_i é gerada como uma combinação linear ponderada entre a solução \mathbf{x}_i da população principal \mathcal{P} e a solução \mathbf{y}_i^{sel} de \mathcal{C} .

Tanto os objetivos de cada solução da população principal \mathcal{P} , quanto as do conjunto de soluções preditas \mathcal{C} são normalizados de acordo com a Equação (3.8) a seguir:

$$\mathbf{y}'_{obj} = \frac{\mathbf{y}_{obj} - \min(\mathbf{y}_{obj})}{\max(\mathbf{y}_{obj}) - \min(\mathbf{y}_{obj})} \quad (3.30)$$

onde \mathbf{y}'_{obj} é o vetor de valores objetivos normalizados; \mathbf{y}_{obj} é o vetor de valores objetivos original; $\min(\mathbf{y}_{obj})$ e $\max(\mathbf{y}_{obj})$ são os valores mínimos e máximo dos objetivos, respectivamente.

Em seguida, para cada solução normalizada no espaço objetivo $\mathcal{Y}'_{obj,i}$ da população principal, escolhe-se suas 5 soluções mais próximas no conjunto de soluções preditas \mathcal{C} utilizando a distância do cosseno calculada como

$$\mathcal{C}_i = \arg \min_{\mathbf{p}_{obj}^{C_j} \in \mathcal{C}} \left(1 - \frac{\mathbf{p}_{obj}^{C_j} \cdot \mathbf{q}'_{obj,i}}{\|\mathbf{p}_{obj}^{C_j}\| \|\mathbf{q}'_{obj,i}\|} \right) \quad \text{para } j = 1, 2, \dots, 5 \quad (3.31)$$

onde \mathcal{C}_i representa o conjunto das 5 soluções preditas mais próximas da i -ésima solução da população principal, selecionadas com base na distância cosseno; $\mathbf{p}_{obj}^{C_j}$ é o vetor objetivo correspondente à j -ésima solução predita no conjunto \mathcal{C} ; $\mathbf{q}'_{obj,i}$ é o vetor objetivo da i -ésima solução da população principal, normalizado; $\|\mathbf{p}_{obj}^{C_j}\|$ e $\|\mathbf{q}'_{obj,i}\|$ são as normas dos vetores objetivos correspondentes às soluções preditas e da população principal, respectivamente.

Após selecionar as 5 soluções mais próximas, uma solução é aleatoriamente selecionada do conjunto \mathcal{C}_i para compor o conjunto \mathcal{C}

$$\mathcal{C}_i^{sel} = \mathbf{p}_{obj}^{C_{j^*}} \quad \text{onde } j^* = \text{random}(1, 5) \quad (3.32)$$

onde \mathcal{C}_i^{sel} é a solução predita selecionada aleatoriamente entre as 5 mais próximas da i -ésima solução da população principal. $\mathbf{p}_{obj}^{C_{j^*}}$ é o vetor objetivo da j^* -ésima solução predita no conjunto \mathcal{C} , onde j^* é um índice selecionado aleatoriamente entre 1 e 5.

Essa equação formaliza o processo de escolha aleatória de uma solução entre as 5 mais próximas, garantindo variação e diversidade na seleção das soluções que serão utilizadas no processo evolucionário.

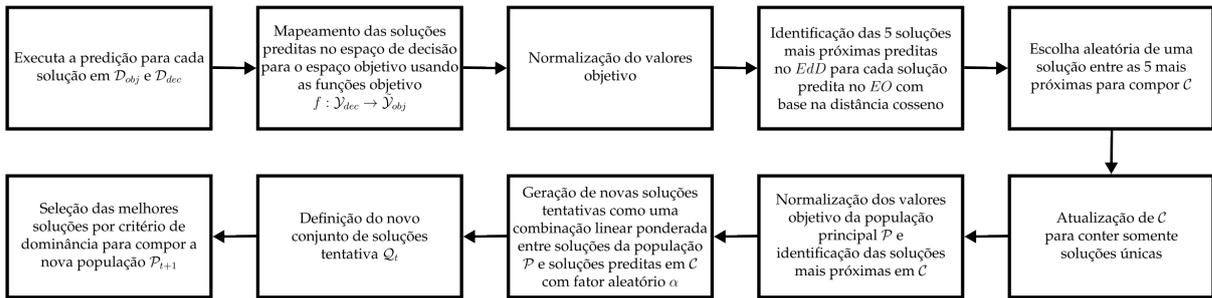
Novas soluções \mathbf{z}_i são geradas de acordo com a Equação (3.33) a seguir:

$$\mathbf{z}_i = \mathbf{x}_i + \alpha \times \mathbf{y}_i^{\text{sel}} \quad (3.33)$$

onde \mathbf{z}_i representa a i -ésima nova solução tentativa; \mathbf{x}_i é a i -ésima solução da população principal \mathcal{P} ; $\mathbf{y}_i^{\text{sel}}$ é a solução predita mais próxima em \mathcal{C} , selecionada de acordo com a distância cosseno; e $\alpha \in [0, 1]$ é uma taxa de aprendizado selecionado uniformemente no intervalo entre 0 e 1. As novas soluções geradas são atribuídas ao conjunto de soluções tentativas \mathcal{Q}_t .

A Figura 8 apresenta um fluxograma que resume todas as etapas descritas para a seleção de soluções promissoras preditas e a geração de novas soluções tentativas.

Figura 8 – Fluxograma para seleção de soluções promissoras preditas e a geração de novas soluções tentativas.



Fonte: Autoria Própria (2024).

Após a definição do conjunto de soluções tentativas \mathcal{Q}_t , o procedimento de seleção ambiental é executado para selecionar as melhores soluções candidatas para compor a população principal \mathcal{P}_{t+1} . A próxima seção detalha o procedimento de seleção ambiental.

3.11 Seleção Ambiental

A seleção ambiental, cujo procedimento está sendo demonstrado no Algoritmo 4, é realizada para selecionar as melhores soluções candidatas para compor a população principal \mathcal{P}_{t+1} . Para isso, o conjunto de soluções candidatas $\overline{\mathcal{P}}_t$ é unido com as soluções tentativas \mathcal{Q}_t , ou seja, $\mathcal{P}_{t+1} \leftarrow \overline{\mathcal{P}}_t \cup \mathcal{Q}_t$. Em seguida, as estas soluções são ordenadas de acordo com o método de ordenação por não-dominância e a população principal \mathcal{P} é atualizada com as N melhores soluções.

Entrada: \mathcal{P}_t e \mathcal{Q}_t

Saída : \mathcal{P}_{t+1}

- 1 $\mathcal{U}_t = \mathcal{P}_t \cup \mathcal{Q}_t$;
- 2 $\mathcal{F} \leftarrow$ execute o procedimento de ordenação por não dominância em \mathcal{U}_t ;
- 3 $\mathcal{P}_{t+1} \leftarrow \emptyset$ e $i \leftarrow 1$;
- 4 **while** $|\mathcal{P}_{t+1}| + |\mathcal{F}_i| \leq N$ **do**
- 5 $\mathcal{C}_i \leftarrow$ distância de aglomeração de \mathcal{F}_i ;
- 6 $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} \cup \mathcal{F}_i$;
- 7 $i \leftarrow i + 1$;
- 8 $\mathcal{F}_i \leftarrow$ ordena de acordo com valores \mathcal{C}_i em ordem decrescente;
- 9 $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} \cup \mathcal{F}_i[1 : (N - |\mathcal{P}_{t+1}|)]$;

Algoritmo 4: Seleção Ambiental

3.12 Considerações do Capítulo

Este capítulo apresenta um novo CMOEA com evolução diferencial de ordenação por não-dominância com predição nos espaços objetivo e decisão, denominado P-NSBiDiCo. A abordagem proposta combina a busca bi-objetivo com a evolução diferencial para resolver problemas de otimização multiobjetivo com restrições. A etapa de busca é composta por um mecanismo de seleção restrita de cruzamento e um novo mecanismo de seleção dos parâmetros F e Cr . Além disso uma estratégia de predição emprega soluções inviáveis não-dominadas para compor o conjunto de dados de treinamento e predição de áreas de busca promissoras no EO e EdD através da estimação de modelos NARX. Soluções promissoras são identificadas e utilizadas para a geração de novas soluções candidatas com o intuito de promover a troca de informações entre soluções preditas e a soluções da população principal.

A seguir, o Capítulo 4 apresenta os resultados obtidos com a aplicação do algoritmo proposto em problemas de otimização multiobjetivo com restrições em diversas bases de *benchmark* e problemas do mundo real.

4

Validação e Testes da Abordagem Proposta

Este capítulo apresenta os resultados experimentais obtidos com a aplicação do algoritmo P-NSBiDiCo em um conjunto diversificado de problemas de otimização multiobjetivo com restrições. O objetivo principal é avaliar o desempenho do algoritmo proposto em comparação com outros métodos de estado da arte e identificar suas principais vantagens e desvantagens. Para tanto, foram conduzidos experimentos em quatro conjuntos de funções de teste e em problemas do mundo real, utilizando métricas de desempenho amplamente utilizadas na literatura.

4.1 Configuração dos Experimentos

Para a avaliação do modelo foram conduzidos experimentos em quatro conjuntos de funções de testes, LIRCMOP, DASC-MOP, DOC e MW, totalizando 46 funções de *benchmark* e em 23 problemas do mundo-real. O desempenho do modelo proposto P-NSBiDiCo foi comparado com 7 algoritmos do estado da arte, sendo o CMOEA/D (Jain; Deb, 2014), CTAEA (Li et al., 2019), CCMO (Tian et al., 2021), CMOEA-MS (Tian et al., 2022), DSPCMDE (Yu et al., 2022), BiCo (Liu; Wang; Tang, 2022) e o modelo de referência NSBiDiCo (Mendes; Araújo; Farias, 2023).

Os parâmetros foram configurados da seguinte forma:

1. *Tamanho da População*: O tamanho da população foi configurado para 100.
2. *Número de Execuções Independentes e Critério de Parada*: Os experimentos foram executados independentemente 20 vezes em cada função de teste até atingir um número máximo de avaliações de função $maxFE$.
3. *Número Máximo de Avaliações de Função (FEs)*: Para todas as bases de teste (DASC-MOP, MW, RWMOP, LIRCMOP e DOC) o número máximo de avaliações de função foi configurado para 3×10^5 .

4. *Probabilidade de Mutação p_m e Índice de Distribuição η_m* : Para a mutação polinomial, a probabilidade de mutação p_m foi configurado para $1/D$, onde D é o número de variáveis de decisão; e o índice da distribuição polinomial η_m para 20.
5. *Número de Soluções Vizinhas Não-Dominadas no Espaço Objetivo n_{os} e Espaço de Decisão n_{ds}* : O número de soluções vizinhas usadas para o treinamento do modelo no espaço decisão n_{ds} foi configurado para 5 e, para o espaço objetivo, n_{os} foi setado como $2 \times n_{ds}$.
6. *Horizonte de Predição no Espaço Objetivo h_{os} e Espaço de Decisão h_{ds}* : Os valores de h_{os} e h_{ds} foram configurados para 5 e 3, respectivamente.
7. *Janela de Gerações para Aplicação da Predição ω* : O valor de ω foi configurado para 30.
8. Para os demais algoritmos, os parâmetros foram mantidos idênticos aos seus respectivos artigos originais.

Todos os experimentos foram executados em um servidor com o sistema operacional Ubuntu 22.04.4 LTS, processador Intel Xeon Silver 4310 de 48 núcleos e com 128Gb de RAM.

Neste trabalho, utilizou-se a plataforma PlatEMO (Tian et al., 2023; Tian et al., 2017) para a criação e avaliação dos modelos. O PlatEMO é uma ferramenta *open source* desenvolvida em MATLAB, projetada para resolver problemas de otimização complexos. A plataforma oferece uma ampla gama de meta-heurísticas, incluindo algoritmos evolucionários, algoritmos de inteligência de enxames, algoritmos multiobjetivo e de otimização assistida por substitutos (*surrogate-assisted*), entre outros. Além disso, ela disponibiliza recursos para a criação e avaliação de novos algoritmos, facilitando a comparação de desempenho entre diversos modelos. A ferramenta também oferece suporte a realização de testes estatísticos para avaliar a significância das diferenças de desempenho entre os algoritmos testados.

As métricas utilizadas para a comparação de desempenho dos algoritmos foram o hipervolume (HV) (Zitzler; Thiele, 1999) e distância geracional invertida (IGD) (Bosman; Thierens, 2003). Mais detalhes sobre cada métrica podem ser encontrados na Seção 2.3. Todas as métricas foram calculadas de acordo com a implementação original da plataforma PlatEMO, considerando apenas o conjunto de soluções viáveis da última geração. Assim, caso o algoritmo não encontre soluções viáveis ao longo de 20 execuções independentes, o valor atribuído para a métrica é *NaN*.

Para avaliar as diferenças estatísticas entre o desempenho dos modelos utilizou-se o teste de soma de *ranks* de Wilcoxon (Wilcoxon, 1945) com nível de significância de 0,05. Os símbolos +, - e \approx nas Tabelas apresentadas na Seção 4.2 e Seção 4.3, a seguir, indicam que o resultado encontrado pelo modelo são, respectivamente, melhor, pior, ou

estatisticamente equivalente ao obtido pelo P-NSBiDiCo. A métrica HV será a única utilizada para avaliar os problemas do mundo-real por não se ter a fronteira do Pareto real.

O objetivo de testar os algoritmos em funções de *benchmarks* e em problemas do mundo real consistem em avaliar sua eficiência, robustez e capacidade de generalização em diferentes cenários. As funções de benchmarks são projetados para representar problemas desafiadores com diversas propriedades, como regiões de viabilidade complexas, frentes de Pareto difíceis de mapear e diferentes dimensões de objetivos e variáveis. Esses testes fornecem uma base comparativa sólida, permitindo verificar como um algoritmo se comporta em relação a outros modelos existentes, tanto em métricas de qualidade de soluções (como IGD e HV) quanto em termos de convergência.

Incluir problemas do mundo real também é muito importante, pois eles representam desafios práticos com aplicações diretas em áreas como engenharia, logística e biomedicina e etc. Testar algoritmos nessas situações garante que eles não apenas funcionem bem em ambientes controlados, mas também sejam eficazes em resolver problemas que envolvem múltiplas restrições e complexidades encontradas em aplicações reais do cotidiano.

As seções a seguir apresentam os resultados comparativos para funções de *benchmark* e problemas do mundo real.

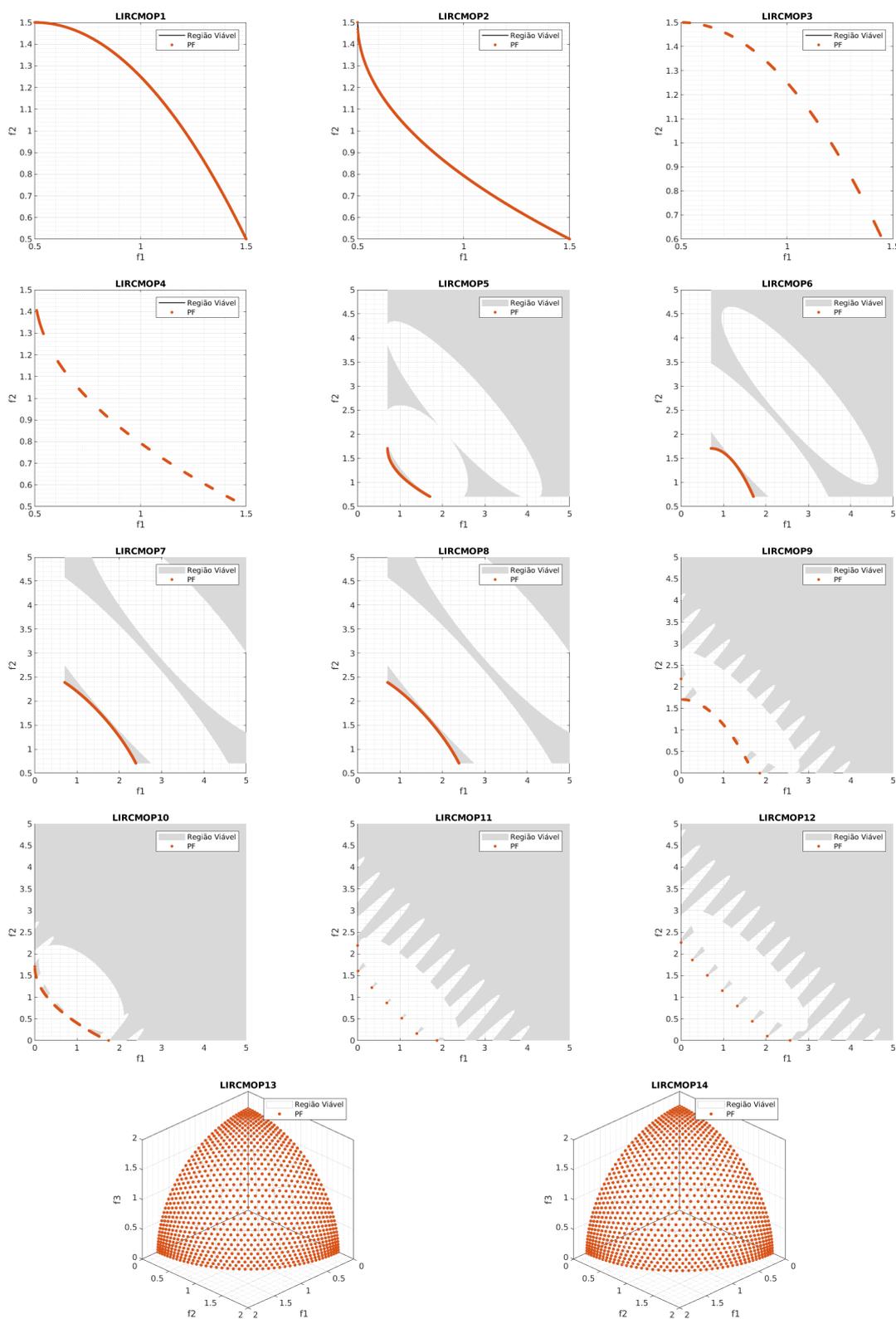
4.2 Resultados Experimentais para os *Benchmarks*

Esta seção descreve os problemas de *benchmarks* utilizados para validar o modelo proposto e comparar seu desempenho na resolução de problemas multiobjetivos com restrições com modelos do estado da arte.

4.2.1 LIRCMOP

O conjunto de problemas de *benchmark* LIRCMOP (Fan et al., 2019a) contém um total de 14 funções de teste com diferentes propriedades. Este conjunto de testes tem como característica principal suas extensas regiões de inviabilidade com pequenas regiões viáveis. A fronteira de Pareto restrita (CPF) é bloqueada por um grande número de regiões inviáveis tornando estas regiões difíceis de serem descobertas durante o processo evolucionário. As funções de restrições do conjunto LIRCMOP é composta por uma função de forma controlável, usada para tornar as formas das CPFs côncavas e convexas, e uma função de distância para ajustar a dificuldade de convergência (Fan et al., 2019a; Liu et al., 2023). A Figura 9 mostra o espaço objetivo 2D das funções de teste da base LIRCMOP, apresentando as suas respectivas fronteiras de Pareto (PF) bem como também a região de viabilidade.

Figura 9 – Ilustração dos LIRCMOPs.



Fonte: Autoria Própria (2024).

As Tabelas 1 e 2 apresentam o desempenho do P-NSBiDiCo e contém a média e o desvio padrão para a métrica do IGD e HV, respectivamente. As colunas M e D correspondem ao número de objetivos do problema e a dimensão do espaço de decisão, respectivamente. Os resultados destacados em cinza representam o melhor resultado encontrado entre os algoritmos comparados.

Para os problemas LIRCMOP1-4, caracterizados por uma região de inviabilidade larga e curva como região de viabilidade e com suas frente de Pareto não restritas (UPF) distantes da sua frente de Pareto restrita (CPF), o NSBiDiCo obteve melhor desempenho para ambas as métricas de IGD e HV. O P-NSBiDiCo obteve melhor desempenho do que COMOEA, CTAEA, CCMO, CMOEA-MS e DSPCMDE. Já o BiCo obteve melhor desempenho para ambas as métricas no LIRCMOP1, LIRCMOP3 e LIRCMOP4 e desempenho estatisticamente similar no LIRCMOP2.

LIRCMOP5 e LIRCMOP6 têm PF convexas e concavas, respectivamente e suas CPFs são as mesmas que UPFs. Nestes problemas o CMOEA-MS obteve o melhor resultado; o P-NSBiDiCo obteve melhor desempenho quando comparado com COMOEA, CTAEA, CMOEA-MS, DSPCMDE e NSBiDiCo, com exceção LIRCMOP6 em que o CCMO obteve melhor desempenho e o DSPCMDE obtiveram resultado equivalente.

LIRCMOP7 e LIRCMOP8 possuem suas UPFs localizadas na região de inviabilidade, e suas CPFs são localizadas nos limites de restrições. Nestes problemas o CCMO e CMOEA-MS obtiveram o melhor desempenho no LIRCMOP7 para ambas as métricas; e o CCMO obteve desempenho equivalente ao P-NSBiDiCo. Todos os demais algoritmos obtiveram desempenho inferior.

LIRCMOP9-12 possuem regiões de inviabilidade largas e um número de segmentos desconectados. LIRCMOP9-10, suas UPF são parte da CPF; LIRCMOP11-12, suas CPF são situadas nos limites de viabilidade de restrições. No LIRCMOP9, o CCMO obteve melhor desempenho na métrica IGD. Já para o HV, CCMO, CMOEA-MS e NSBiDiCo obtiveram melhor desempenho. No LIRCMOP10, o CCMO e CMOEA-MS obtiveram o melhor desempenho para ambas as métricas. No LIRCMOP11 o CCMO, DSPCMDE obtiveram desempenho equivalente ao P-NSBiDiCo para o IGD. Já para o HV, o NSBiDiCo obteve melhor resultado. Para LIRCMOP12, CCMO, DSPCMDE e NSBiDiCo obtiveram melhor desempenho para IGD e CCMO e NSBiDiCo tiveram melhor performance no HV.

LIRCMOP13-14 contêm 3 objetivos. LIRCMOP13 contém CPF igual a sua UPF e LIRCMOP14 contém CPF localizada nos limites das restrições. Para LIRCMOP13-14 os algoritmos CMOEA/D, CTAEA, CCMO, CMOEA-MS e BiCo obtiveram melhor desempenho em ambas as métricas.

Ao comparar o P-NSBiDiCo ao NSBiDiCo, pode-se observar que o P-NSBiDiCo obteve melhor desempenho em 9 dos 14 problemas para a métrica do IGD, e melhor desempenho em 7 de 14 problemas para o HV. Isso sugere uma melhora de desempenho com a inclusão das estratégias de seleção de parâmetros e de predição.

No geral, quando comparados os algoritmos um a um, o P-NSBiDiCo se mantém competitivo para os problemas LIRCMOP. Para ambas as métricas o P-NSBiDiCo têm desempenho superior em metade, ou mais da metade, dos problemas e perde apenas para o CCMO.

A Figura 10 apresenta o perfil de convergência média dos algoritmos ao longo do processo evolucionário nos problemas LIRCMOP. Em problemas como o DASCMP1, pode-se notar que o algoritmo P-NSBiDiCo apresentou uma rápida convergência, sendo superior ao NSBiDiCo e equivalente ao CCMO em termos de velocidade. Já para DASCMP2 observa-se comportamento semelhante, sendo a convergência do P-NSBiDiCo ligeiramente mais rápida que a do CCMO no início do processo evolucionário.

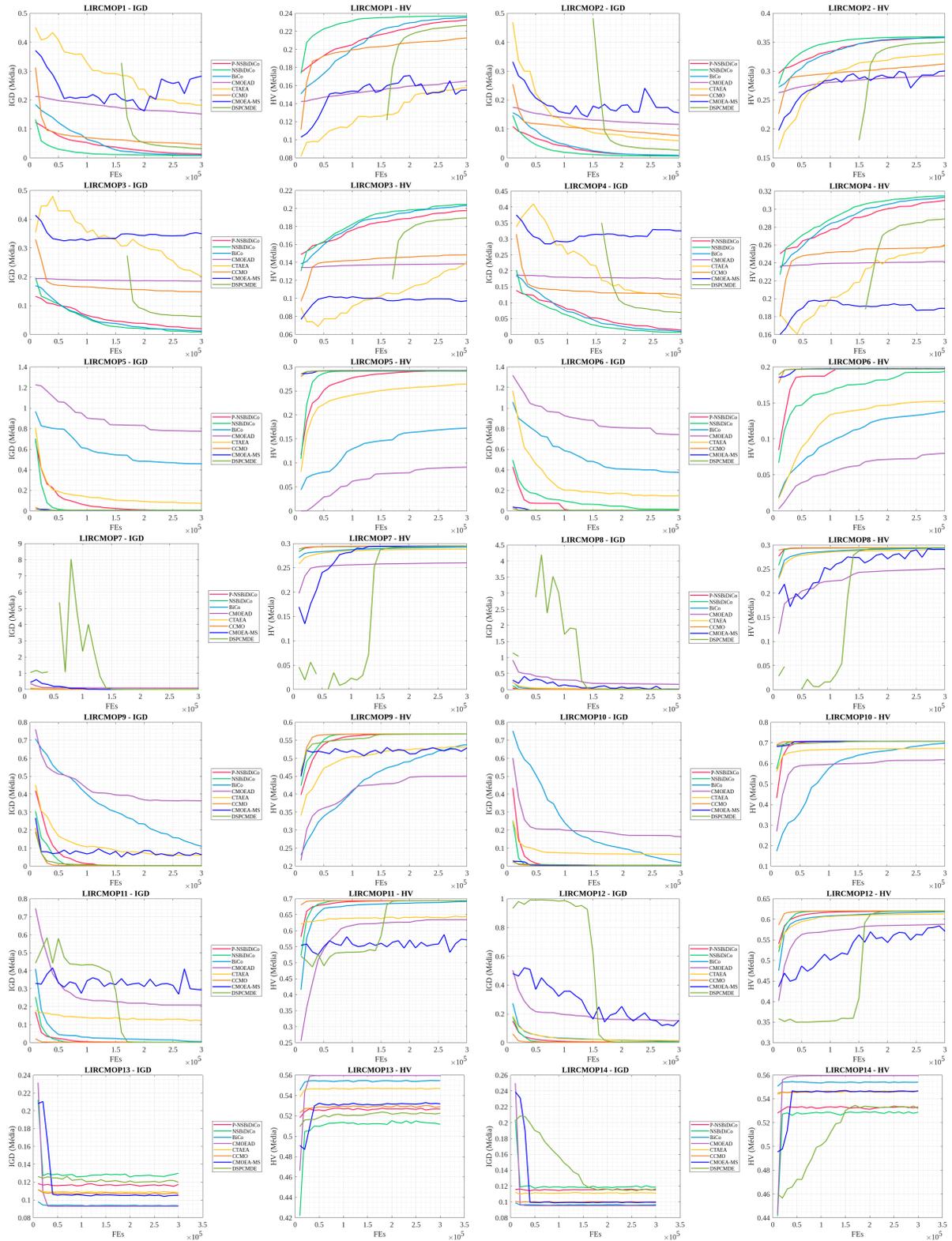
Tabela 1 – Média e desvio padrão de IGD para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas LIRCMOP.

| Problema | M | D | CMOEAD | CTAEA | CCMO | CMOEA_MS | DSPCMDE | BiCo | NSBiDiCo | P-NSBiDiCo |
|-----------|---|----|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| LIRCMOP1 | 2 | 10 | 1.5156e-1 (5.84e-2) | 1.8211e-1 (1.31e-1) | 4.5523e-2 (2.65e-2) | 2.8248e-1 (1.73e-1) | 3.1918e-2 (2.12e-2) | 9.4891e-3 (1.99e-3) | 8.0311e-3 (9.16e-4) | 1.2794e-2 (3.67e-3) |
| LIRCMOP2 | 2 | 10 | 1.1554e-1 (2.76e-2) | 5.8987e-2 (2.37e-2) | 7.6979e-2 (3.52e-2) | 1.5560e-1 (6.03e-2) | 2.6895e-2 (1.33e-2) | 8.6359e-3 (2.53e-3) | 6.3407e-3 (6.21e-4) | 7.3888e-3 (1.17e-3) |
| LIRCMOP3 | 2 | 10 | 1.8526e-1 (5.61e-2) | 2.0157e-1 (1.57e-1) | 1.4781e-1 (6.12e-2) | 3.4983e-1 (1.19e-1) | 6.1958e-2 (1.13e-1) | 1.1705e-2 (6.25e-3) | 8.6873e-3 (7.11e-3) | 1.9790e-2 (9.46e-3) |
| LIRCMOP4 | 2 | 10 | 1.7437e-1 (5.67e-2) | 1.1399e-1 (5.51e-2) | 1.2304e-1 (4.26e-2) | 3.2554e-1 (1.06e-1) | 6.8845e-2 (2.70e-2) | 9.6907e-3 (5.69e-3) | 6.6131e-3 (4.71e-3) | 1.3674e-2 (9.55e-3) |
| LIRCMOP5 | 2 | 10 | 7.7521e-1 (5.37e-1) | 7.1967e-2 (2.10e-2) | 5.0392e-3 (2.05e-4) | 4.5378e-3 (1.58e-4) | 5.8601e-3 (2.46e-4) | 4.5777e-1 (5.30e-1) | 6.6582e-3 (3.78e-4) | 5.7425e-3 (3.69e-4) |
| LIRCMOP6 | 2 | 10 | 7.4159e-1 (6.22e-1) | 1.4736e-1 (1.49e-1) | 5.2617e-3 (1.78e-4) | 4.8810e-3 (1.42e-4) | 6.0882e-3 (2.37e-4) | 3.7371e-1 (5.81e-1) | 1.3077e-2 (3.01e-2) | 5.5941e-3 (1.37e-4) |
| LIRCMOP7 | 2 | 10 | 8.8754e-2 (2.98e-2) | 1.9820e-2 (6.11e-3) | 7.1742e-3 (1.63e-4) | 6.9710e-3 (3.57e-4) | 7.8352e-3 (2.71e-4) | 8.9291e-3 (2.16e-3) | 8.6048e-3 (3.23e-4) | 7.6778e-3 (2.38e-4) |
| LIRCMOP8 | 2 | 10 | 1.6603e-1 (3.61e-1) | 1.6313e-2 (2.08e-3) | 7.2637e-3 (2.30e-4) | 1.7912e-2 (4.81e-2) | 7.8632e-3 (2.12e-4) | 8.2463e-3 (2.76e-3) | 8.6347e-3 (3.69e-4) | 7.6311e-3 (2.42e-4) |
| LIRCMOP9 | 2 | 10 | 3.6319e-1 (6.95e-2) | 6.1076e-2 (3.45e-2) | 2.6554e-3 (6.36e-5) | 6.3889e-2 (6.05e-2) | 3.1299e-3 (2.30e-4) | 1.1096e-1 (6.55e-2) | 3.4273e-3 (1.29e-4) | 2.9663e-3 (9.52e-5) |
| LIRCMOP10 | 2 | 10 | 1.6383e-1 (5.41e-2) | 6.6099e-2 (6.80e-2) | 4.6015e-3 (1.32e-4) | 4.8311e-3 (2.55e-4) | 5.0158e-3 (1.57e-4) | 1.9379e-2 (1.79e-2) | 5.7280e-3 (1.66e-4) | 5.0223e-3 (1.81e-4) |
| LIRCMOP11 | 2 | 10 | 2.0911e-1 (9.96e-2) | 1.2221e-1 (3.98e-2) | 2.3909e-3 (5.36e-5) | 2.9348e-1 (1.17e-1) | 2.3422e-3 (7.35e-5) | 6.2249e-3 (7.71e-3) | 2.4334e-3 (5.21e-5) | 2.3710e-3 (8.22e-5) |
| LIRCMOP12 | 2 | 10 | 1.5320e-1 (6.25e-2) | 1.4539e-2 (3.71e-3) | 2.9767e-3 (1.23e-4) | 1.5543e-1 (3.13e-1) | 3.3896e-3 (2.81e-4) | 7.1189e-3 (8.48e-3) | 3.1654e-3 (1.21e-4) | 3.7163e-3 (5.70e-4) |
| LIRCMOP13 | 3 | 10 | 9.2899e-2 (1.16e-7) | 1.0837e-1 (1.79e-3) | 1.0658e-1 (1.54e-3) | 1.0522e-1 (1.53e-3) | 1.1975e-1 (3.73e-3) | 9.3655e-2 (1.10e-3) | 1.2998e-1 (5.11e-3) | 1.1749e-1 (4.72e-3) |
| LIRCMOP14 | 3 | 10 | 9.5313e-2 (7.70e-7) | 1.1102e-1 (9.80e-4) | 1.0014e-1 (1.40e-3) | 9.9582e-2 (1.77e-3) | 1.1672e-1 (3.76e-3) | 9.6510e-2 (1.09e-3) | 1.1869e-1 (4.82e-3) | 1.1510e-1 (3.75e-3) |
| + / - / ≈ | | | 2/12/0 | 2/12/0 | 9/4/1 | 6/7/1 | 1/7/6 | 4/6/4 | 5/9/0 | |

Tabela 2 – Média e desvio padrão de HV para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas LIRCMOP.

| Problema | M | D | CMOEAD | CTAEA | CCMO | CMOEA_MS | DSPCMDE | BiCo | NSBiDiCo | P-NSBiDiCo |
|-----------|---|----|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| LIRCMOP1 | 2 | 10 | 1.6503e-1 (2.27e-2) | 1.5742e-1 (3.82e-2) | 2.1274e-1 (1.74e-2) | 1.5508e-1 (3.59e-2) | 2.2642e-1 (7.23e-3) | 2.3542e-1 (1.99e-3) | 2.3696e-1 (5.65e-4) | 2.3289e-1 (3.00e-3) |
| LIRCMOP2 | 2 | 10 | 2.9185e-1 (1.17e-2) | 3.2987e-1 (1.21e-2) | 3.1260e-1 (2.34e-2) | 3.0026e-1 (2.29e-2) | 3.5013e-1 (5.21e-3) | 3.5778e-1 (1.41e-3) | 3.5938e-1 (3.72e-4) | 3.5829e-1 (1.07e-3) |
| LIRCMOP3 | 2 | 10 | 1.3844e-1 (1.60e-2) | 1.3942e-1 (3.53e-2) | 1.4850e-1 (2.05e-2) | 9.7370e-2 (2.14e-2) | 1.8935e-1 (3.02e-2) | 2.0339e-1 (3.20e-3) | 2.0447e-1 (4.43e-3) | 1.9770e-1 (4.71e-3) |
| LIRCMOP4 | 2 | 10 | 2.4149e-1 (2.83e-2) | 2.5873e-1 (3.14e-2) | 2.5958e-1 (2.06e-2) | 1.8925e-1 (3.38e-2) | 2.8901e-1 (1.10e-2) | 3.1326e-1 (2.49e-3) | 3.1474e-1 (2.30e-3) | 3.0974e-1 (4.39e-3) |
| LIRCMOP5 | 2 | 10 | 9.1173e-2 (1.15e-1) | 2.6478e-1 (7.01e-3) | 2.9218e-1 (1.00e-4) | 2.9245e-1 (8.81e-5) | 2.9177e-1 (1.21e-4) | 1.7263e-1 (1.30e-1) | 2.9145e-1 (1.87e-4) | 2.9189e-1 (1.75e-4) |
| LIRCMOP6 | 2 | 10 | 7.9966e-2 (8.28e-2) | 1.5230e-1 (4.13e-2) | 1.9730e-1 (1.12e-4) | 1.9756e-1 (7.15e-5) | 1.9693e-1 (1.05e-4) | 1.3797e-1 (8.34e-2) | 1.9356e-1 (1.45e-2) | 1.9726e-1 (9.01e-5) |
| LIRCMOP7 | 2 | 10 | 2.6025e-1 (9.72e-3) | 2.8849e-1 (2.36e-3) | 2.9458e-1 (7.48e-5) | 2.9472e-1 (1.89e-4) | 2.9436e-1 (1.27e-4) | 2.9295e-1 (1.79e-3) | 2.9389e-1 (1.41e-4) | 2.9444e-1 (1.23e-4) |
| LIRCMOP8 | 2 | 10 | 2.5118e-1 (6.05e-2) | 2.9090e-1 (8.49e-4) | 2.9454e-1 (1.21e-4) | 2.9061e-1 (1.79e-2) | 2.9439e-1 (9.37e-5) | 2.9347e-1 (2.82e-3) | 2.9388e-1 (1.67e-4) | 2.9448e-1 (1.22e-4) |
| LIRCMOP9 | 2 | 10 | 4.5012e-1 (3.02e-2) | 5.3132e-1 (2.20e-2) | 5.6754e-1 (5.34e-5) | 5.2881e-1 (2.86e-2) | 5.6733e-1 (2.18e-4) | 5.3802e-1 (1.68e-2) | 5.6745e-1 (5.74e-5) | 5.6734e-1 (1.42e-4) |
| LIRCMOP10 | 2 | 10 | 6.1802e-1 (3.97e-2) | 6.7255e-1 (2.83e-2) | 7.0769e-1 (9.29e-5) | 7.0775e-1 (1.41e-4) | 7.0753e-1 (1.07e-4) | 6.9844e-1 (1.01e-2) | 7.0734e-1 (1.16e-4) | 7.0752e-1 (1.07e-4) |
| LIRCMOP11 | 2 | 10 | 6.3452e-1 (6.09e-2) | 6.4439e-1 (1.33e-2) | 6.9401e-1 (8.31e-6) | 5.7155e-1 (5.53e-2) | 6.9398e-1 (2.12e-5) | 6.9133e-1 (5.50e-3) | 6.9403e-1 (6.97e-6) | 6.9397e-1 (2.64e-5) |
| LIRCMOP12 | 2 | 10 | 5.8786e-1 (2.08e-2) | 6.1308e-1 (1.66e-3) | 6.2032e-1 (1.90e-5) | 5.7078e-1 (9.52e-2) | 6.2012e-1 (1.06e-4) | 6.1845e-1 (3.72e-3) | 6.2033e-1 (1.24e-5) | 6.2004e-1 (1.88e-4) |
| LIRCMOP13 | 3 | 10 | 5.5962e-1 (3.29e-7) | 5.4681e-1 (1.40e-3) | 5.2903e-1 (2.30e-3) | 5.3183e-1 (2.53e-3) | 5.2267e-1 (3.96e-3) | 5.5459e-1 (1.06e-3) | 5.1173e-1 (3.91e-3) | 5.2664e-1 (3.88e-3) |
| LIRCMOP14 | 3 | 10 | 5.5959e-1 (8.36e-6) | 5.4651e-1 (7.50e-4) | 5.4637e-1 (1.41e-3) | 5.4673e-1 (2.23e-3) | 5.3184e-1 (3.91e-3) | 5.5403e-1 (1.08e-3) | 5.2881e-1 (3.60e-3) | 5.3323e-1 (2.26e-3) |
| + / - / ≈ | | | 2/12/0 | 2/12/0 | 8/4/2 | 6/6/2 | 0/7/7 | 5/5/4 | 7/7/0 | |

Figura 10 – Perfil de convergência (IGD e HV) dos algoritmos para os LIRCMOPs.



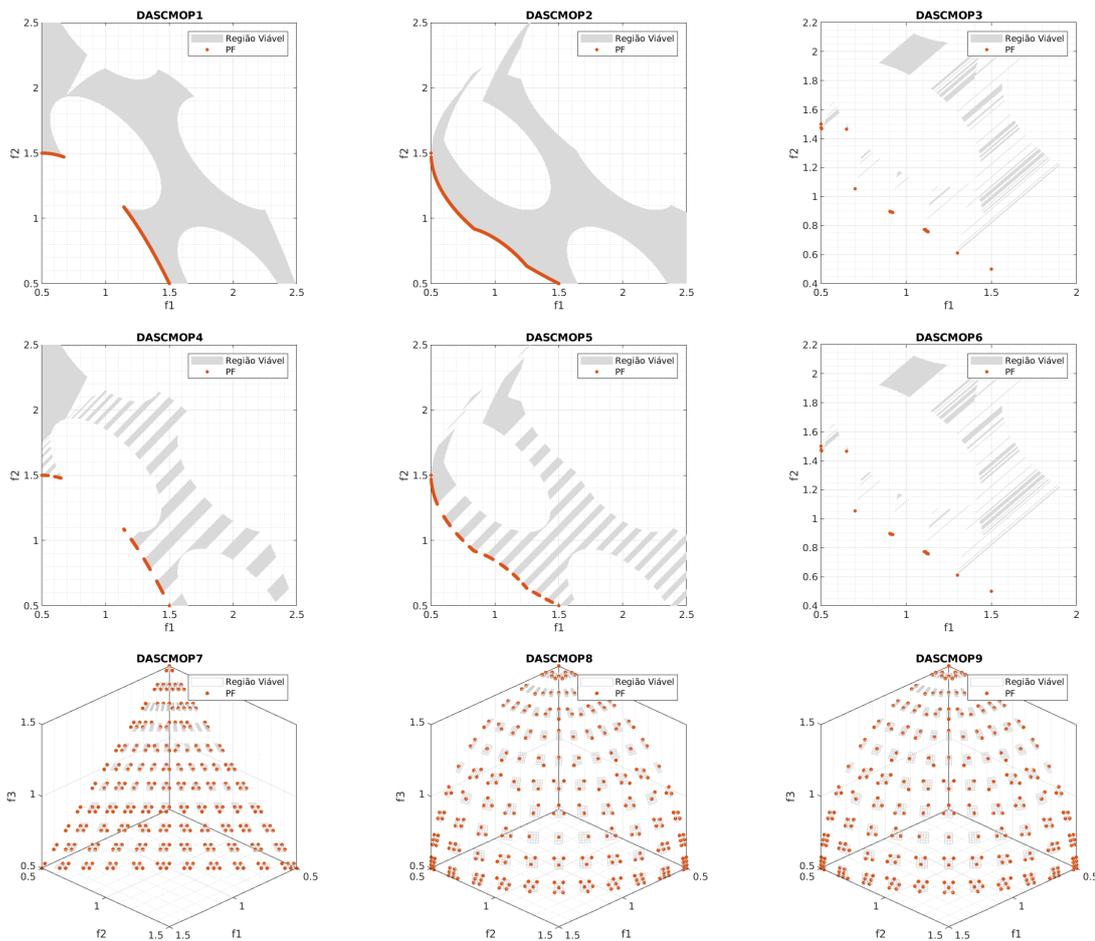
Fonte: Autoria Própria (2024).

4.2.2 DASC MOP

A base DASC MOP (Fan et al., 2020) contém 9 funções de testes com dificuldades escaláveis e adaptáveis, permitindo a flexibilidade de ajustar a complexidade dos problemas em termos de dificuldade e a escalabilidade do número de objetivos. A principal característica do DASC MOP consiste em simular desafios típicos em problemas de otimização, com três tipos de dificuldades essenciais: dificuldade de viabilidade, convergência e diversidade. Esses aspectos são controlados por meio de funções de restrições parametrizadas, o que torna a base altamente desafiadora para algoritmos evolucionários.

A Figura 11 apresenta o espaço 2D das funções de teste da base DASC MOP, destacando suas respectivas fronteiras de Pareto (PF) e a região de viabilidade.

Figura 11 – Ilustração dos DASC MOPs.



Fonte: Autoria Própria (2024).

As Tabelas 7 e 8 mostram o desempenho do P-NSBiDiCo comparado aos demais modelos para os problemas DASC MOP, contendo a média e o desvio padrão para a métrica do IGD e HV, respectivamente. As colunas M e D correspondem ao número

de objetivos do problema e à dimensão do espaço de decisão, respectivamente. Os resultados destacados em cinza representam os melhores resultados encontrados entre os algoritmos comparados.

Os problemas DASC MOP são classificados de acordo com os tipos de dificuldade e geometria da PF. O DASC MOP1, por exemplo, apresenta dificuldade de viabilidade com uma PF restrita com muitos pontos nas bordas das regiões viáveis, enquanto o DASC MOP2 traz dificuldade de convergência, com a PF obscurecida por regiões inviáveis, dificultando a convergência para a PF. Já o DASC MOP3 possui dificuldade de diversidade, com uma PF fragmentada, com segmentos discretos e distribuição esparsa de soluções viáveis. O DASC MOP4 caracteriza-se pela dificuldade de diversidade e viabilidade, com geometria da PF descontínua e restrita e com poucas soluções viáveis. Problemas mais complexos, como o DASC MOP7, envolvem todas as três dificuldades, tornando-se extremamente desafiador obter uma frente de Pareto completa.

O DASC MOP5 tem dificuldade de convergência e de viabilidade, com PF obscurecida e restrita, tornando-se desafiador encontrar soluções viáveis. Já o DASC MOP6 possui dificuldade de diversidade e convergência, com uma PF fragmentada e de difícil acesso e com muitas regiões de inviabilidade. No DASC MOP8 observa-se a dificuldade de viabilidade e de diversidade. Sua PF é restrita, com segmentos discretos, dificultando a obtenção de uma PF completa. E, por fim, o DASC MOP9 possui dificuldade de convergência e de diversidade. Sua PF é obscurecida e fragmentada, complicando tanto a convergência quanto a distribuição de soluções ao longo da PF.

Para este conjunto de problemas, o algoritmo BiCo obteve desempenho superior em ambas as métricas de IGD e HV, especialmente em cinco dos nove problemas (DASC MOP4-DASC MOP8). No entanto, o P-NSBiDiCo se destacou em problemas específicos como DASC MOP2, que envolve dificuldade de convergência. O P-NSBiDiCo, também superou o CMOEA/D, o BiCo e o NSBiDiCo em DASC MOP1 a DASC MOP3. Isso sugere que o P-NSBiDiCo é eficiente em problemas com dificuldade de convergência, mas enfrenta desafios maiores em cenários onde a diversidade e a viabilidade podem ser mais exigentes, como nos problemas DASC MOP4 a DASC MOP8.

O P-NSBiDiCo obteve melhor desempenho que o CMOEA/D, CTAEA, BiCo e NSBiDiCo no DASC MOP9, um problema com dificuldade de convergência e de diversidade. O CCMO obteve melhor resultado para este problema em ambas as métricas. Para os demais problemas (DASC MOP4-DASC MOP8), evidencia-se que os P-NSBiDiCo obteve maior dificuldade e os demais algoritmos obtiveram melhor desempenho em ambas as métricas.

A Figura 12 apresenta o perfil de convergência média dos algoritmos ao longo do processo evolucionário nos problemas DASC MOP. Em problemas como o DASC MOP1, pode-se notar que o algoritmo P-NSBiDiCo apresentou uma rápida convergência, sendo

superior ao NSBiDiCo e equivalente ao CCMO em termos de velocidade. Já para DASC-MOP2 observa-se comportamento semelhante, sendo a convergência do P-NSBiDiCo ligeiramente mais rápida que a do CCMO no início do processo evolucionário.

No DASC-MOP3, o P-NSBiDiCo converge ligeiramente mais rápido que o CCMO no início, mas acaba sendo superado pelo CCMO e DSPCMDE. Ainda assim, fica evidente a melhoria em seu desempenho quando comparado com o NSBiDiCo e BiCo. Nos demais problemas, as curvas de convergência evidenciam a dificuldade do P-NSBiDiCo em problemas mais fragmentados.

Tabela 3 – Média e desvio padrão de IGD para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas DASC-MOP.

| Problem | M | D | CMOEAD | CTAEA | CCMO | CMOEA_MS | DSPCMDE | BiCo | NSBiDiCo | P-NSBiDiCo |
|-----------|---|----|--|---|---|------------------------------|------------------------------|----------------------------|---------------------|------------|
| DASC-MOP1 | 2 | 30 | 6.9606e-1 (4.61e-2) - 1.8062e-1 (1.60e-2) - 3.1372e-3 (2.86e-4) + | 3.5728e-2 (1.46e-1) - 3.5812e-3 (5.76e-4) + | 7.1285e-1 (3.09e-2) - 3.2275e-3 (1.50e-4) + | 5.7121e-3 (2.29e-3) | | | | |
| DASC-MOP2 | 2 | 30 | 2.2057e-1 (2.48e-2) - 7.1368e-2 (2.92e-2) - 4.2936e-3 (1.53e-4) - | 4.0845e-2 (7.55e-2) ≈ | 4.5917e-3 (9.95e-5) - | 2.3487e-1 (2.31e-2) - | 1.2837e-2 (3.50e-2) - | 4.0770e-3 (8.82e-5) | | |
| DASC-MOP3 | 2 | 30 | 3.4390e-1 (2.50e-4) - 1.4114e-1 (2.84e-2) ≈ | 2.5703e-2 (2.74e-2) + | 2.9061e-1 (9.97e-2) - | 1.8518e-2 (2.44e-3) + | 2.7819e-1 (6.09e-2) - | 1.9126e-1 (1.38e-1) - | 7.4595e-2 (6.36e-2) | |
| DASC-MOP4 | 2 | 30 | 1.6870e-2 (5.33e-2) + 9.1437e-3 (1.82e-3) + | 2.1961e-2 (8.75e-2) + | 1.5325e-1 (1.43e-1) + | 5.3566e-1 (4.08e-1) ≈ | 1.2747e-3 (1.41e-4) + | 4.9798e-2 (1.00e-1) + | 3.9514e-1 (5.96e-2) | |
| DASC-MOP5 | 2 | 30 | 5.0495e-3 (8.05e-4) + 7.1488e-3 (4.77e-4) + | 3.5326e-3 (3.30e-4) + | 1.6984e-1 (1.78e-1) + | 4.2785e-1 (3.44e-1) ≈ | 2.7275e-3 (5.01e-5) + | 9.5313e-2 (2.34e-1) + | 5.3662e-1 (2.30e-1) | |
| DASC-MOP6 | 2 | 30 | 1.1031e-1 (1.87e-1) + 2.4837e-2 (2.96e-3) + | 8.5774e-2 (2.00e-1) + | 2.0781e-1 (2.07e-1) + | 5.2579e-1 (2.66e-1) ≈ | 1.5709e-2 (5.33e-3) + | 1.2179e-1 (2.13e-1) + | 6.9759e-1 (8.06e-2) | |
| DASC-MOP7 | 3 | 30 | 4.3817e-2 (3.09e-3) + 3.8651e-2 (8.18e-4) + | 7.7161e-2 (1.65e-1) + | 2.6412e-1 (7.30e-2) ≈ | 6.6936e-1 (4.72e-1) ≈ | 3.1871e-2 (8.35e-4) + | 9.5598e-2 (1.01e-1) + | 3.6796e-1 (2.48e-1) | |
| DASC-MOP8 | 3 | 30 | 6.3765e-2 (2.68e-3) + 6.2254e-2 (1.44e-2) + | 1.2191e-1 (1.54e-1) + | 3.0618e-1 (7.14e-2) ≈ | 5.0246e-1 (3.45e-1) ≈ | 4.1165e-2 (8.36e-4) + | 1.0915e-1 (1.17e-1) + | 3.3899e-1 (3.61e-1) | |
| DASC-MOP9 | 3 | 30 | 2.4463e-1 (1.58e-1) - 1.9573e-1 (6.58e-2) - | 4.0697e-2 (7.37e-4) + | 5.4896e-2 (7.03e-2) + | 5.6443e-2 (3.02e-3) ≈ | 3.2342e-1 (6.06e-2) - | 6.3035e-2 (3.34e-3) - | 5.6162e-2 (3.22e-3) | |
| + / - / ≈ | | | 5/4/0 | 5/3/1 | 8/1/0 | 4/2/3 | 2/1/6 | 5/4/0 | 6/3/0 | |

Tabela 4 – Média e desvio padrão de HV para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas DASC-MOP.

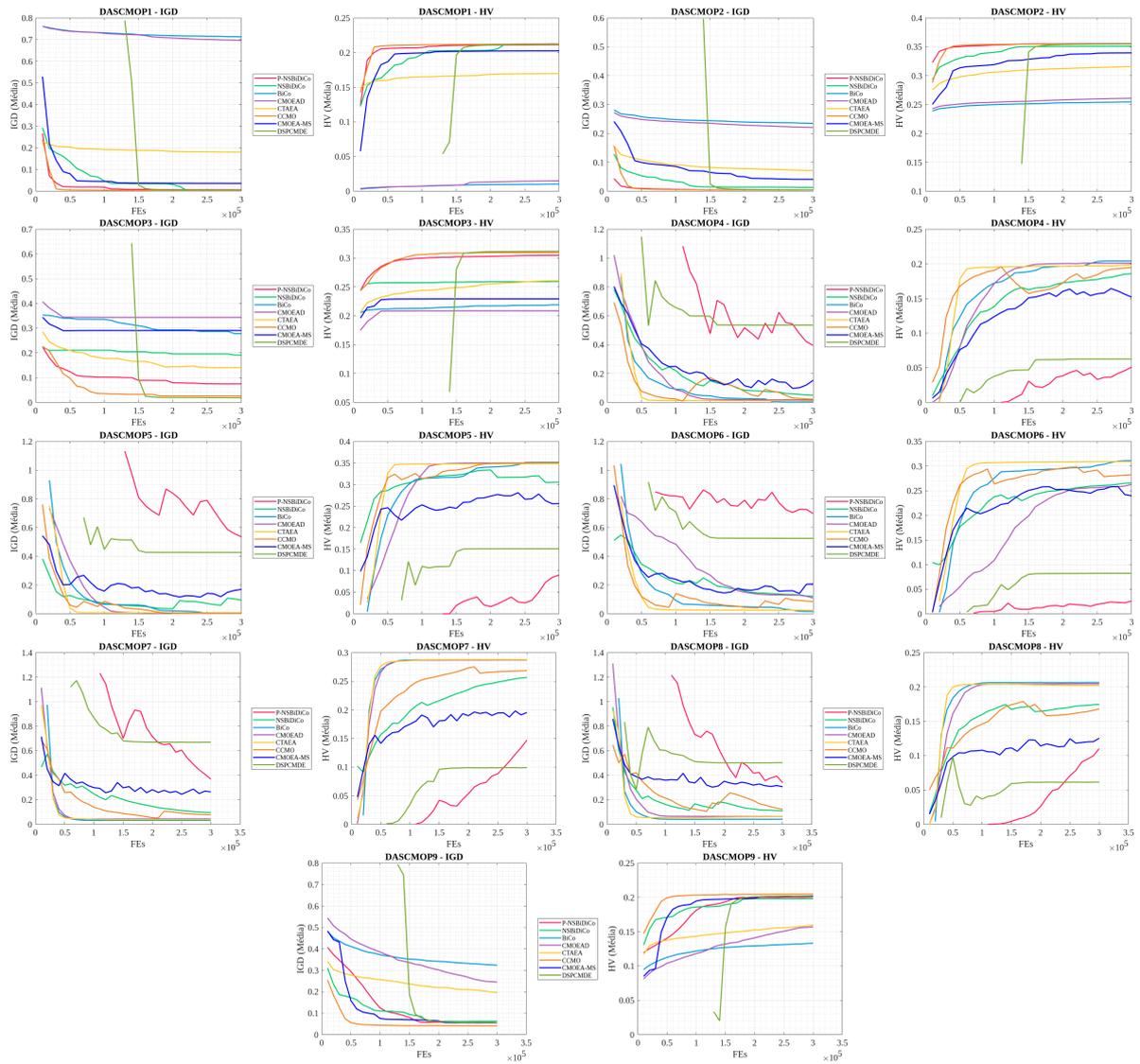
| Problema | M | D | CMOEAD | CTAEA | CCMO | CMOEA-MS | DSPCMDE | BiCo | NSBiDiCo | P-NSBiDiCo |
|-----------|---|----|---|------------------------------|-----------------------|------------------------------|------------------------------|------------------------------|----------------------------|------------|
| DASC-MOP1 | 2 | 30 | 1.4898e-2 (1.44e-2) - 1.6969e-1 (9.39e-4) - | 2.1247e-1 (4.13e-4) + | 2.0251e-1 (4.31e-2) ≈ | 2.1219e-1 (3.22e-4) ≈ | 1.0227e-2 (5.66e-3) - | 2.1268e-1 (2.84e-4) + | 2.1144e-1 (1.74e-3) | |
| DASC-MOP2 | 2 | 30 | 2.6095e-1 (5.16e-3) - 3.1573e-1 (9.54e-3) - | 3.5515e-1 (1.16e-4) - | 3.3951e-1 (3.25e-2) - | 3.5501e-1 (7.76e-5) - | 2.5464e-1 (4.60e-3) - | 3.5111e-1 (1.67e-2) - | 3.5557e-1 (6.79e-5) | |
| DASC-MOP3 | 2 | 30 | 2.0870e-1 (3.05e-5) - 2.6089e-1 (6.61e-3) - | 3.0980e-1 (1.02e-2) + | 2.2929e-1 (3.37e-2) - | 3.1194e-1 (3.81e-4) + | 2.1909e-1 (1.12e-2) - | 2.5954e-1 (4.57e-2) ≈ | 3.0500e-1 (6.41e-3) | |
| DASC-MOP4 | 2 | 30 | 2.0118e-1 (8.71e-3) + 1.9865e-1 (3.29e-3) + | 1.9459e-1 (3.40e-2) + | 1.5201e-1 (2.59e-2) + | 6.2649e-2 (7.69e-2) ≈ | 2.0415e-1 (1.96e-4) + | 1.8586e-1 (3.85e-2) + | 5.0980e-2 (1.73e-2) | |
| DASC-MOP5 | 2 | 30 | 3.5036e-1 (3.36e-4) + 3.4837e-1 (4.26e-4) + | 3.5042e-1 (4.13e-4) + | 2.5590e-1 (8.44e-2) + | 1.5088e-1 (1.34e-1) ≈ | 3.5168e-1 (8.44e-5) + | 3.0574e-1 (1.10e-1) + | 9.0041e-2 (1.08e-1) | |
| DASC-MOP6 | 2 | 30 | 2.6292e-1 (8.71e-2) + 3.0917e-1 (9.70e-4) + | 2.8232e-1 (8.77e-2) + | 2.3964e-1 (8.42e-2) + | 8.2255e-2 (1.03e-1) ≈ | 3.1119e-1 (4.32e-3) + | 2.6587e-1 (9.95e-2) + | 2.5862e-2 (1.14e-2) | |
| DASC-MOP7 | 3 | 30 | 2.8704e-1 (5.90e-4) + 2.8774e-1 (2.33e-4) + | 2.6855e-1 (5.59e-2) + | 1.9549e-1 (2.73e-2) ≈ | 9.9086e-2 (1.21e-1) ≈ | 2.8778e-1 (4.36e-4) + | 2.5667e-1 (4.79e-2) + | 1.4727e-1 (8.38e-2) | |
| DASC-MOP8 | 3 | 30 | 2.0460e-1 (4.23e-4) + 2.0211e-1 (3.13e-3) + | 1.6811e-1 (6.69e-2) + | 1.2531e-1 (2.21e-2) ≈ | 6.1415e-2 (8.29e-2) ≈ | 2.0661e-1 (4.69e-4) + | 1.7468e-1 (5.00e-2) + | 1.1022e-1 (7.91e-2) | |
| DASC-MOP9 | 3 | 30 | 1.5729e-1 (3.99e-2) - 1.5977e-1 (1.77e-2) - | 2.0490e-1 (4.01e-4) + | 2.0204e-1 (1.75e-2) + | 2.0121e-1 (1.10e-3) + | 1.3310e-1 (1.12e-2) - | 1.9806e-1 (1.16e-3) - | 2.0046e-1 (9.49e-4) | |
| + / - / ≈ | | | 5/4/0 | 5/4/0 | 8/1/0 | 4/2/3 | 2/1/6 | 5/4/0 | 6/2/1 | |

4.2.3 DOC

O conjunto de testes DOC (Liu; Wang, 2019) contém um total de 9 funções de testes. Este conjunto de testes considera restrições no espaço objetivo e também no espaço de decisão simultaneamente e possui restrições de desigualdade e restrições de igualdade. Devido a isso, os autores argumentam que as funções na base DOC exibem potencial para simular cenários de problemas do mundo-real. A fronteira de Pareto para este conjunto de testes exibe diferentes características, tais como contínua, desconectada, convexa, concava, linear, mista, degenerada e multimodal. As regiões de viabilidade também possuem propriedades, tais como não-linear, muito pequenas e multimodais.

A Figura 13 mostra o gráfico das funções de teste da base DOC, apresentando a suas respectivas fronteiras de Pareto (PF) bem como também a região de viabilidade.

Figura 12 – Gráficos de Convergência dos DASC-MOPs.

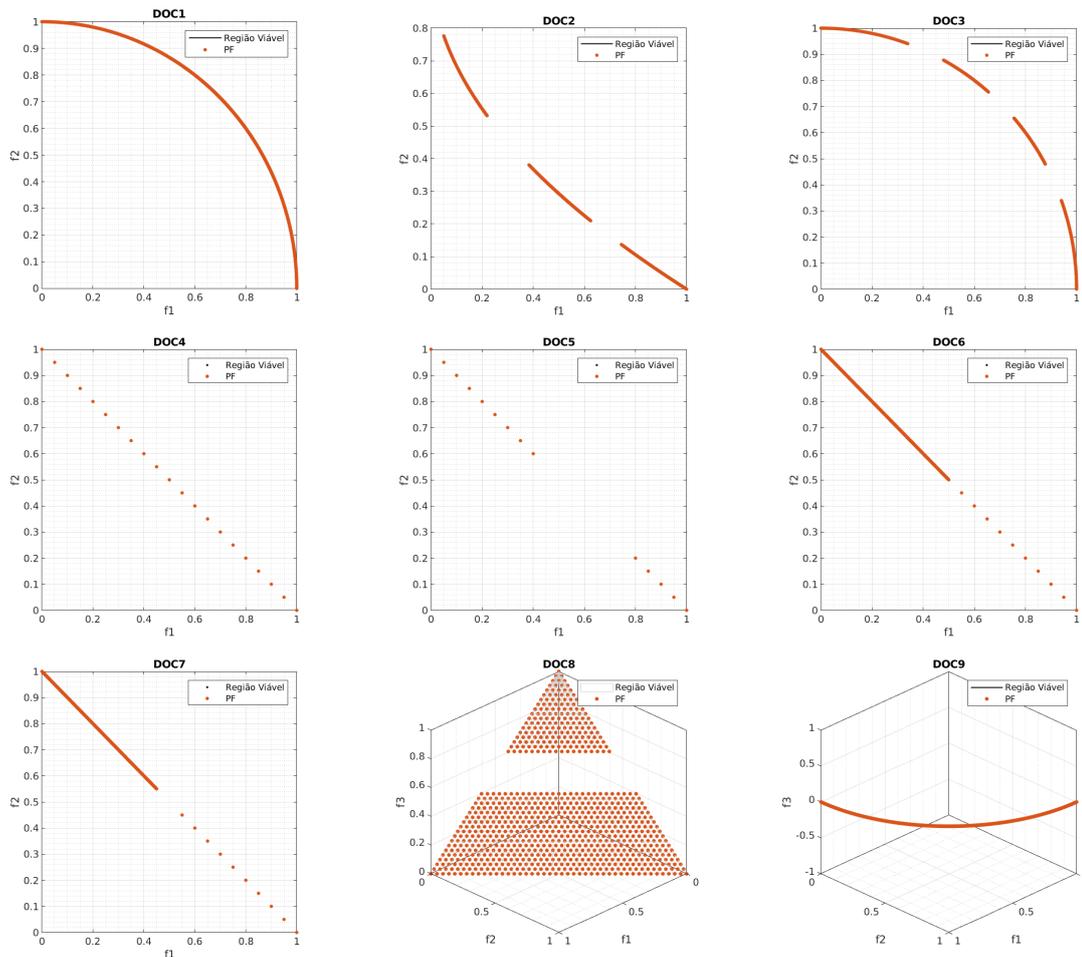


Fonte: Autoria Própria (2024).

As Tabelas 5 e 6 mostram o desempenho do P-NSBiDiCo comparado aos demais modelos para os problemas DOC e contém a média e o desvio padrão para a métrica do IGD e HV, respectivamente. As colunas *M* e *D* correspondem ao número de objetivos do problema e a dimensão do espaço de decisão, respectivamente. Os resultados destacados em cinza representam o melhor resultado encontrado entre os algoritmos comparados.

No DOC1, que possui PF côncava e contínua e região de viabilidade não-linear, o CMOEA-MS obteve o melhor desempenho para ambas as métricas. O CMOEA/D, CTAEA e BiCo apresentaram os piores resultados em ambas as métricas, exceto o BiCo, que obteve desempenho equivalente no HV.

O DOC2, possui PF convexa e descontínua e uma região viável muito pequena e não-linear. Neste problema o CMOEA/D, CTAEA, CCMO, CMOEA-MS e BiCo

Figura 13 – Ilustração das funções de *benchmark* da base DOC.

Fonte: Autoria Própria (2024).

não encontraram soluções viáveis. Já o CCMO, DSPCMDE e NSBiDiCo obtiveram desempenho estatisticamente similar, sendo que o NSBiDiCo e CCMO obtiveram o melhor resultado para IGD e HV, respectivamente.

Para o DOC3, que possui uma PF côncava, descontínua e multimodal com espaço de busca muito pequeno, não-linear e multi-multimodal, o CTAEA não obteve soluções viáveis. Os modelos CMOEA/D, CCMO, CMOEA-MS, BiCo e NSBiDiCo outros modelos obtiveram desempenho inferior no IGD, mas equivalente na métrica HV. O DSPCMDE obteve o melhor resultado para ambas as métricas.

Os problemas DOC4 e DOC8 contêm PF linear e descontínua com região de viabilidade muito pequena e não-linear. No DOC4 e DOC8 os algoritmos CMOEA/D, CTAEA e BiCo obtiveram pior desempenho em ambas as métricas. No DOC4 o CCMO obteve desempenho equivalente em ambas as métricas; o CMOEA-MS, DSPCMDE e NSBiDiCo obtiveram o melhores desempenho em IGD e HV, sendo que o NSBiDiCo obteve o melhor resultado encontrado para ambas as métricas. Já no DOC8, o CCMO,

CMOEA-MS, DSPCMDE e NSBiDiCo obtiveram o melhor desempenho, sendo o do CMOEA-MS o melhor entre todos para este problema.

O DOC5 possui PF descontínua e multi-modal com espaço de viabilidade não-linear e muito pequeno. Neste problema os algoritmos CMOEA/D, CTAEA, DSPCMDE e BiCo não encontraram soluções viáveis. O CMOEA-MS obteve desempenho equivalente em ambas as métricas, enquanto o NSBiDiCo e CCMO obtiveram os melhores desempenhos, sendo o do CCMO o melhor resultado encontrado para ambas as métricas.

O DOC6 apresenta PF mista e multimodal, mas com um espaço viável muito pequeno e não linear. Neste problema o DSPCMDE obteve o melhor desempenho para ambas as métricas. Já o CMOEA/D, CTAEA e BiCo obtiveram pior desempenho para IGD e HV. O CCMO, CMOEA-MS e NSBiDiCo obtiveram desempenho equivalente na métrica IGD. O CMOEA-MS e NSBiDiCo obtiveram melhor desempenho no HV.

Já o DOC7 tem PF multimodal, porém com um espaço viável muito pequeno e multimodal. O CTAEA não obteve soluções viáveis para este problema. O NSBiDiCo obteve o melhor resultado encontrado para ambas as métricas. O CMOEA/D e BiCo obtiveram desempenho inferior para ambas as métricas. CCMO e DSPCMDE obtiveram desempenho superior ao P-NSBiDiCo no IGD e HV, ao passo que CMOEA-MS obteve desempenho equivalente.

O DOC9 contém PF multimodal e degenerada com espaço de viabilidade muito pequeno, não-linear e multimodal. O CCMO e CMOEA-MS obtiveram o melhor desempenho, sendo o do CMOEA-MS o melhor resultado para este problema na métrica IGD. CMOEA/D e NSBiDiCo obtiveram desempenho inferior, ao passo que DSPCMDE e BiCo obtiveram desempenho similar. Embora os algoritmos tenham encontrado soluções viáveis no DOC9, com exceção do CTAEA, conforme evidenciado pelos resultados do IGD, o valor do HV foi reportado como *NaN*. Isso pode ser atribuído à dificuldade no cálculo do hipervolume, pois o DOC9 possui uma fronteira de Pareto multimodal e degenerada, além de uma região viável muito pequena e não-linear. Essas características podem ter impactado a distribuição das soluções, resultando em um valor de HV não computável, apesar de a qualidade das soluções ter sido suficiente para gerar um IGD válido.

A Figura 16 apresenta o perfil de convergência média dos algoritmos ao longo do processo evolucionário nos problemas DOC. No DOC2 o P-NSBiDiCo apresenta uma convergência ligeiramente mais rápida, mas logo é superado pelo NSBiDiCo. No DOC3, o IGD mostra uma convergência estável, melhor que a do NSBiDiCo, mas é superado pelo DSPCMDE. Para o DOC6 observa-se uma rápida convergência no início do processo evolucionário, no entanto em estágios finais outros modelos superam o P-NSBiDiCo, evidenciando a necessidade de melhor exploração neste problema.

Observa-se ainda que em problemas como o DOC1, DOC4 e DOC6 o P-NSBiDiCo converge para valores que são próximos dos valores encontrados pelos modelos com melhor desempenho.

No geral, o algoritmo P-NSBiDiCo demonstrou competitividade ao superar ou igualar outros algoritmos em diversos problemas, especialmente nos mais complexos, como DOC6, DOC7 e DOC8; a partir dos resultados observa-se que o P-NSBiDiCo supera os modelos CMOEA/D, CTAEA e BiCo em quase todos os problemas. Além disso, para os demais modelos, o P-NSBiDiCo obteve desempenho estatisticamente equivalente ou melhor em aproximadamente metade dos problemas, demonstrando que apesar de não ter obtido o melhor resultado encontrado para nenhum dos problemas, este ainda consegue encontrar soluções viáveis de alta qualidade e é capaz de explorar eficientemente o espaço de soluções.

Tabela 5 – Média e desvio padrão de IGD para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas DOC.

| Problema | M | D | CMOEAD | CTAEA | CCMO | CMOEA_MS | DSPCMDE | BiCo | NSBiDiCo | P-NSBiDiCo |
|-----------|---|----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|
| DOC1 | 2 | 6 | 1.4151e+0 (1.04e+0) - | 5.6647e+2 (3.04e+2) - | 5.7838e-3 (4.96e-4) + | 5.5039e-3 (5.34e-4) + | 2.9804e+2 (4.62e+2) ≈ | 1.5468e-2 (2.14e-2) - | 6.3701e-3 (6.32e-4) + | 7.1169e-3 (5.50e-4) |
| DOC2 | 2 | 16 | NaN (NaN) | NaN (NaN) | 3.4336e-2 (9.88e-2) ≈ | NaN (NaN) | 4.4124e-1 (3.19e-4) ≈ | NaN (NaN) | 3.3399e-2 (1.32e-1) ≈ | 3.5178e-2 (0.00e+0) |
| DOC3 | 2 | 10 | 6.0155e+2 (2.06e+2) - | NaN (NaN) | 5.5870e+2 (3.89e+2) - | 7.9792e+2 (4.45e+2) - | 1.0705e+2 (1.76e+2) + | 6.0536e+2 (1.43e+2) - | 3.5149e+2 (1.22e+2) - | 2.3558e+2 (1.33e+2) |
| DOC4 | 2 | 8 | 5.9367e-1 (2.71e-1) - | 1.7655e+2 (2.04e+2) - | 2.5109e-2 (6.32e-3) ≈ | 1.8739e-2 (6.75e-3) + | 1.9835e-2 (1.90e-3) + | 2.7202e-1 (1.64e-1) - | 1.5848e-2 (2.19e-3) + | 2.4950e-2 (5.29e-3) |
| DOC5 | 2 | 8 | NaN (NaN) | NaN (NaN) | 5.8204e-2 (8.85e-2) + | 3.6480e+2 (5.08e+2) ≈ | NaN (NaN) | NaN (NaN) | 4.8966e+1 (6.53e+1) + | 6.5389e+1 (6.83e+1) |
| DOC6 | 2 | 11 | 6.7228e-1 (5.05e-1) - | 3.3821e+1 (3.92e+1) - | 4.1452e-2 (1.08e-1) ≈ | 3.5813e-3 (2.21e-3) ≈ | 2.7317e-3 (1.15e-4) + | 8.5070e-1 (9.78e-1) - | 3.6498e-3 (2.22e-4) ≈ | 1.3695e-2 (3.34e-2) |
| DOC7 | 2 | 11 | 5.5827e+0 (2.00e+0) - | NaN (NaN) | 3.5180e-2 (1.46e-1) + | 1.9640e+0 (1.86e+0) ≈ | 1.8647e-2 (7.07e-2) + | 5.4266e+0 (1.50e+0) - | 3.0581e-3 (1.51e-4) + | 1.5399e+0 (1.03e+0) |
| DOC8 | 3 | 10 | 3.6918e+1 (4.62e+1) - | 4.1879e+2 (2.28e+2) - | 7.2051e-2 (6.41e-3) + | 6.0351e-2 (3.10e-3) + | 1.1930e-1 (9.63e-3) + | 4.1586e+1 (2.82e+1) - | 1.0764e-1 (6.74e-3) + | 3.4334e+0 (3.61e+0) |
| DOC9 | 3 | 11 | 3.8910e-1 (5.52e-2) - | NaN (NaN) | 8.3153e-2 (1.23e-2) + | 4.7686e-2 (5.22e-3) + | 1.1496e-1 (1.34e-2) ≈ | 1.3680e-1 (8.87e-2) ≈ | 1.3918e-1 (2.00e-2) - | 1.1982e-1 (1.49e-2) |
| + / - / ≈ | | | 0/7/0 | 0/4/0 | 5/1/3 | 4/1/3 | 5/0/3 | 0/6/1 | 5/2/2 | |

Tabela 6 – Média e desvio padrão de HV para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas DOC.

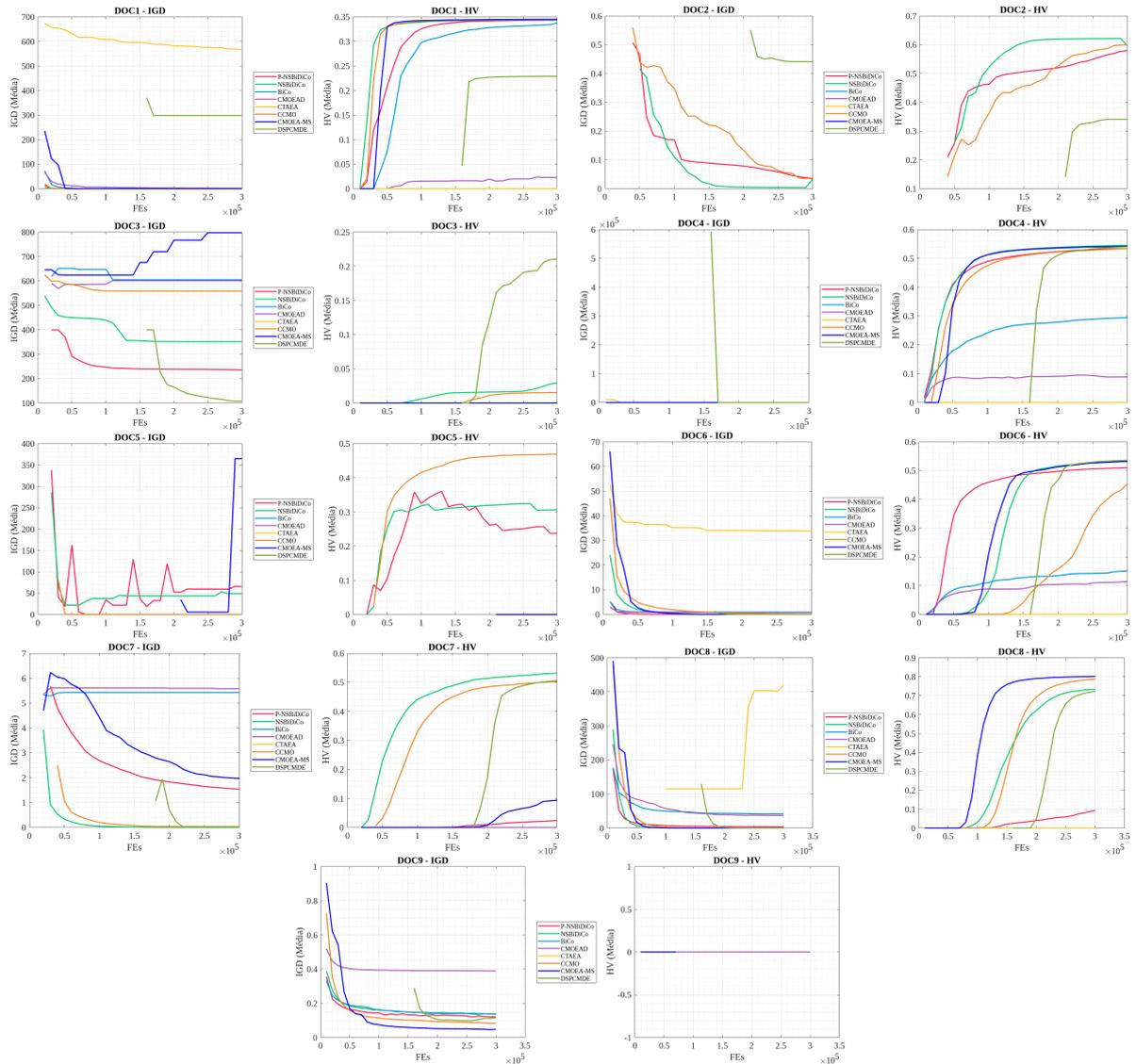
| Problema | M | D | CMOEAD | CTAEA | CCMO | CMOEA_MS | DSPCMDE | BiCo | NSBiDiCo | P-NSBiDiCo |
|-----------|---|----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|
| DOC1 | 2 | 6 | 2.2919e-2 (4.77e-2) - | 0.0000e+0 (0.00e+0) - | 3.4470e-1 (8.80e-4) + | 3.4532e-1 (4.33e-4) + | 2.2904e-1 (1.77e-1) ≈ | 3.3769e-1 (1.69e-2) ≈ | 3.4386e-1 (6.82e-4) + | 3.4315e-1 (5.39e-4) |
| DOC2 | 2 | 16 | NaN (NaN) | NaN (NaN) | 5.9963e-1 (6.28e-2) ≈ | NaN (NaN) | 3.4083e-1 (3.32e-4) ≈ | NaN (NaN) | 5.9549e-1 (1.17e-1) ≈ | 5.7987e-1 (0.00e+0) |
| DOC3 | 2 | 10 | 0.0000e+0 (0.00e+0) ≈ | NaN (NaN) | 1.5410e-2 (6.89e-2) ≈ | 0.0000e+0 (0.00e+0) ≈ | 2.1080e-1 (1.58e-1) + | 0.0000e+0 (0.00e+0) ≈ | 2.9155e-2 (9.04e-2) ≈ | 0.0000e+0 (0.00e+0) |
| DOC4 | 2 | 8 | 8.8762e-2 (9.83e-2) - | 0.0000e+0 (0.00e+0) - | 5.3392e-1 (7.15e-3) ≈ | 5.4177e-1 (7.64e-3) + | 5.4048e-1 (2.37e-3) + | 2.9398e-1 (1.43e-1) - | 5.4519e-1 (2.80e-3) + | 5.3427e-1 (6.11e-3) |
| DOC5 | 2 | 8 | NaN (NaN) | NaN (NaN) | 4.6916e-1 (4.32e-2) + | 0.0000e+0 (0.00e+0) ≈ | NaN (NaN) | NaN (NaN) | 3.0654e-1 (2.45e-1) ≈ | 2.3819e-1 (2.49e-1) |
| DOC6 | 2 | 11 | 1.1499e-1 (1.51e-1) - | 0.0000e+0 (0.00e+0) - | 4.5433e-1 (1.25e-1) - | 5.3143e-1 (1.15e-2) + | 5.3548e-1 (5.00e-3) + | 1.5074e-1 (1.97e-1) - | 5.3351e-1 (6.29e-3) + | 5.1019e-1 (4.86e-2) |
| DOC7 | 2 | 11 | 0.0000e+0 (0.00e+0) - | NaN (NaN) | 5.0246e-1 (1.18e-1) + | 9.3724e-2 (1.89e-1) ≈ | 5.604e-1 (9.80e-2) + | 0.0000e+0 (0.00e+0) - | 5.3177e-1 (1.23e-2) + | 2.4286e-2 (6.12e-2) |
| DOC8 | 3 | 10 | 0.0000e+0 (0.00e+0) - | 0.0000e+0 (0.00e+0) ≈ | 7.8753e-1 (7.84e-3) + | 8.0164e-1 (3.88e-3) + | 7.2217e-1 (1.11e-2) + | 0.0000e+0 (0.00e+0) - | 7.3363e-1 (8.98e-3) + | 9.3604e-2 (1.96e-1) |
| DOC9 | 3 | 11 | 0.0000e+0 (0.00e+0) ≈ | NaN (NaN) | NaN (NaN) |
| + / - / ≈ | | | 0/5/1 | 0/3/1 | 4/1/3 | 4/0/3 | 5/0/2 | 0/4/2 | 5/0/3 | |

4.2.4 MW

A base de testes MW (Ma; Wang, 2019) contém 14 problemas com diversas propriedades, incluindo diversos níveis de dificuldade, regiões viáveis irregularmente distribuídas e diferentes configurações geométricas, tais como regiões viáveis desconectas e estrutura da frente de Pareto complexas. Este conjunto de problemas é escalável em termos do número de objetivos e variáveis de decisão.

A Figura 15 ilustra o gráfico das funções de teste da base MW, apresentando a suas respectivas fronteiras de Pareto (PF) bem como também a região de viabilidade.

Figura 14 – Gráficos de Convergência dos DOC.

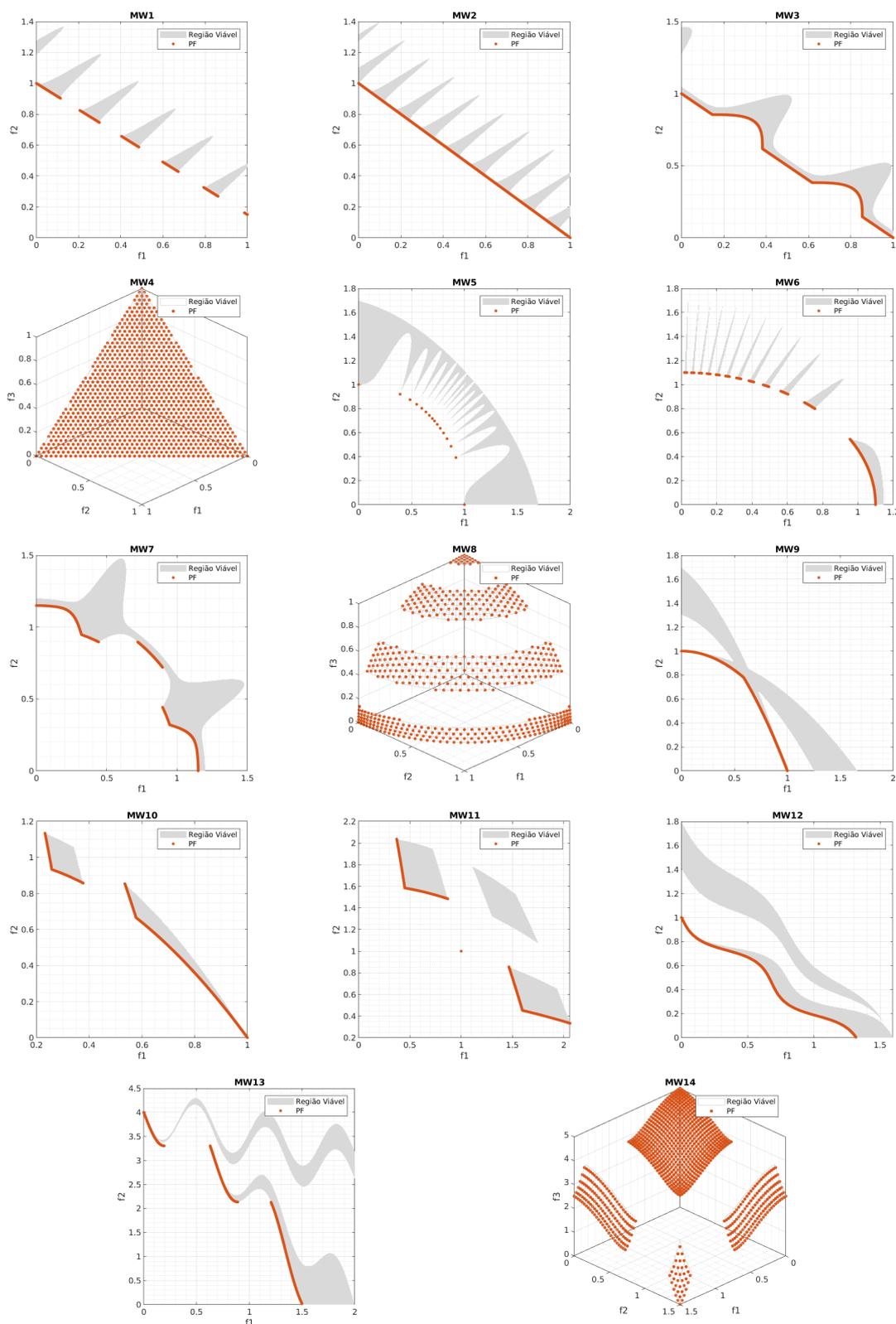


Fonte: Autoria Própria (2024).

As Tabelas 7 e 8 mostram o desempenho do P-NSBiDiCo comparado aos demais modelos para os problemas MW, contendo a média e o desvio padrão para a métrica do IGD e HV, respectivamente. As colunas *M* e *D* correspondem ao número de objetivos do problema e à dimensão do espaço de decisão, respectivamente. Os resultados destacados em cinza representam os melhores resultados encontrados entre os algoritmos comparados.

Os problemas MW podem ser classificados em quatro tipos (ver Subseção 2.1.2): Tipo I (MW2, MW4, MW14), Tipo II (MW1, MW5, MW6, MW8), Tipo III (MW3, MW7, MW10, MW13) e Tipo IV (MW9, MW11, MW12). Cada tipo apresenta diferentes combinações de CPFs e regiões de viabilidade conectadas ou desconectadas.

Figura 15 – Ilustração dos MWs.



Fonte: Autoria Própria (2024).

Para problemas do Tipo I, o MW2 possui CPF linear com região de viabilidade desconectada; MW4 possui CPF linear com região de viabilidade conectada; MW14 possui CPF desconectada com região de viabilidade conectada.

Já os problemas do Tipo II, o MW1 possui CPF desconectada com região de viabilidade desconectada; MW5 possui CPF discreta com região de viabilidade conectada; MW6 possui CPF desconectada com região de viabilidade desconectada; MW8 possui CPF desconectada com região de viabilidade desconectada.

No Tipo III, o MW3 possui CPF mista com região de viabilidade conectada; MW7 possui CPF côncava com região de viabilidade conectada; MW10 possui CPF desconectada com região de viabilidade desconectada; MW13 possui CPF desconectada com região de viabilidade desconectada.

Para problemas do Tipo IV, MW9 possui CPF côncava com região de viabilidade conectada; MW11 possui CPF desconectada com região de viabilidade desconectada; MW12 possui CPF mista com região de viabilidade desconectada.

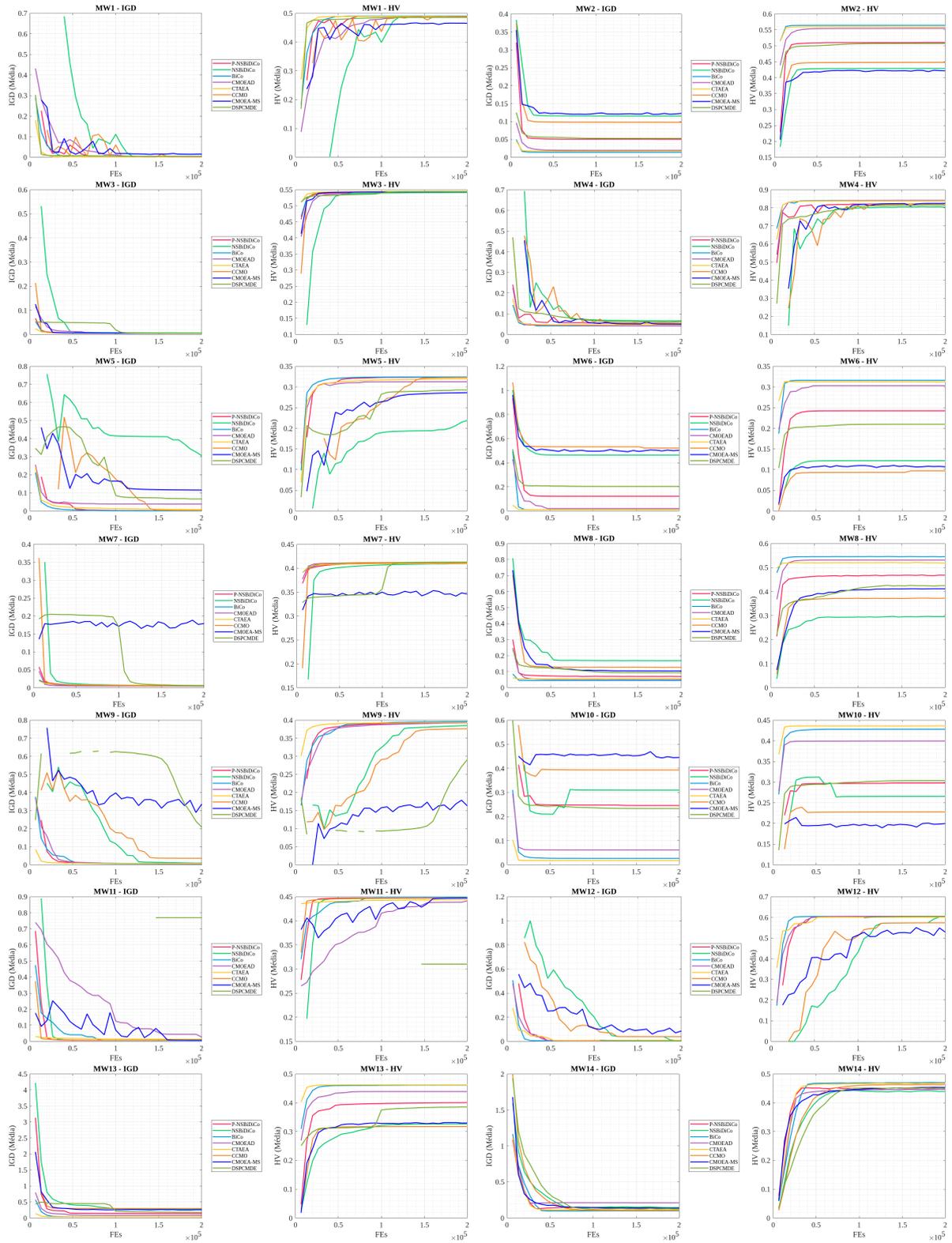
Para este conjunto de testes o algoritmo BiCo superou todos os modelos em ambas as métricas de HV e IGD. Pode-se observar que o modelo P-NSBiDiCo supera o NSBiDiCo em quase todos os problemas, evidenciando melhora de desempenho com a utilização das abordagens de controle de parâmetros e predição. Além deste, o P-NSBiDiCo apresenta resultado superior aos modelos CCMO, CMOEA-MS e DSPCMDE, para ambas as métricas IGD e HV.

A Figura 16 apresenta o perfil de convergência média dos algoritmos ao longo do processo evolucionário nos problemas MW. Com exceção do MW10, o P-NSBiDiCo demonstrou uma convergência mais rápida em comparação ao NSBiDiCo e a outros algoritmos, como o CCMO, CMOEA-MS, DSPCMDE e CMOEA/D. Esse resultado pode sugerir que a melhora identificada pode estar relacionada à inclusão dos novos mecanismos de ajuste paramétrico e de predição.

Tabela 7 – Média e desvio padrão de IGD para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas MW.

| Problema | M | D | CMOEAD | CTAEA | CCMO | CMOEA_MS | DSPCMDE | BiCo | NSBiDiCo | P-NSBiDiCo |
|-----------|---|----|---------------------|-----------------------|-----------------------|-----------------------|------------------------|-----------------------|-----------------------|-----------------------|
| MW1 | 2 | 15 | 3.1888e-3 (1.27e-5) | - 2.0099e-3 (7.21e-5) | - 2.1382e-3 (1.24e-4) | - 1.5273e-2 (4.88e-3) | - 2.1111e-3 (7.80e-5) | - 1.6167e-3 (1.51e-5) | 2.3150e-3 (9.49e-5) | - 1.8218e-3 (3.92e-5) |
| MW2 | 2 | 15 | 1.8460e-2 (7.94e-3) | + 1.4614e-2 (6.33e-3) | + 1.2422e-1 (8.14e-2) | - 1.3182e-1 (7.69e-2) | - 5.7871e-2 (2.63e-2) | ≈ 1.4172e-2 (6.58e-3) | + 9.9110e-2 (4.30e-2) | - 5.0227e-2 (1.50e-2) |
| MW3 | 2 | 15 | 4.6645e-3 (2.78e-4) | + 4.7129e-3 (2.40e-4) | + 5.1630e-3 (1.49e-4) | - 5.6480e-3 (2.10e-4) | - 5.2174e-3 (1.90e-4) | - 4.6216e-3 (1.52e-4) | + 6.3196e-3 (3.54e-4) | - 4.9781e-3 (1.69e-4) |
| MW4 | 3 | 15 | 4.1125e-2 (1.42e-5) | + 4.6706e-2 (2.75e-4) | + 5.3873e-2 (1.57e-3) | ≈ 5.1911e-2 (1.37e-3) | + 6.2007e-2 (4.54e-3) | - 4.0944e-2 (3.49e-4) | + 6.5713e-2 (1.81e-3) | - 5.3658e-2 (1.69e-3) |
| MW5 | 2 | 15 | 7.3348e-4 (1.11e-3) | + 8.0441e-3 (1.91e-3) | - 2.6665e-3 (1.37e-3) | - 1.1522e-1 (2.70e-1) | - 5.782e-2 (4.71e-2) | - 5.2626e-4 (1.69e-4) | + 4.1179e-1 (3.76e-1) | - 1.0901e-3 (5.90e-4) |
| MW6 | 2 | 15 | 2.1402e-2 (1.94e-2) | + 1.2508e-2 (6.92e-3) | + 4.5247e-1 (1.84e-1) | - 5.0204e-1 (1.24e-1) | - 2.1095e-1 (1.88e-1) | - 1.0549e-2 (7.88e-3) | + 4.0025e-1 (2.25e-1) | - 1.2125e-1 (1.51e-1) |
| MW7 | 2 | 15 | 4.5838e-3 (7.86e-5) | + 6.3212e-3 (5.11e-4) | - 5.1305e-3 (4.31e-4) | ≈ 1.7965e-1 (2.35e-2) | - 5.9230e-3 (9.67e-4) | - 4.2757e-3 (2.16e-4) | + 5.9787e-3 (3.00e-4) | - 5.0003e-3 (3.44e-4) |
| MW8 | 3 | 15 | 5.3310e-2 (5.11e-3) | + 5.4553e-2 (1.87e-3) | + 1.4052e-1 (5.14e-2) | - 1.3143e-1 (7.28e-2) | - 1.1052e-1 (4.42e-2) | - 4.4833e-2 (1.19e-3) | + 1.4574e-1 (4.57e-2) | - 6.9274e-2 (7.81e-3) |
| MW9 | 2 | 15 | 5.7028e-3 (8.04e-4) | ≈ 8.1567e-3 (6.72e-4) | - 7.1297e-2 (2.04e-1) | ≈ 3.8477e-1 (1.98e-1) | - 2.2527e-1 (2.73e-1) | - 4.8928e-3 (5.48e-4) | + 1.1230e-2 (2.05e-3) | - 5.3685e-3 (8.24e-4) |
| MW10 | 2 | 15 | 8.7738e-2 (1.48e-1) | + 1.2572e-2 (1.08e-2) | + 2.3000e-1 (1.02e-1) | ≈ 3.3367e-1 (1.49e-1) | ≈ 2.8906e-1 (1.79e-1) | ≈ 2.9994e-2 (2.76e-2) | + 3.1609e-1 (1.67e-1) | ≈ 2.4595e-1 (1.57e-1) |
| MW11 | 2 | 15 | 2.7036e-2 (7.64e-2) | - 1.2452e-2 (1.18e-3) | - 6.4411e-3 (1.86e-4) | - 6.5229e-3 (2.28e-4) | - 7.6944e-1 (2.28e-16) | - 5.9345e-3 (9.39e-5) | + 7.1719e-3 (2.86e-4) | - 6.0678e-3 (1.25e-4) |
| MW12 | 2 | 15 | 4.7639e-3 (9.91e-6) | + 7.7158e-3 (7.11e-4) | - 3.7923e-2 (1.46e-1) | - 1.1959e-1 (1.33e-1) | - 6.3776e-3 (5.99e-3) | - 4.6353e-3 (1.14e-4) | + 4.4917e-2 (1.48e-1) | - 4.8804e-3 (1.18e-4) |
| MW13 | 2 | 15 | 7.7871e-2 (2.38e-2) | + 4.1935e-2 (2.90e-2) | + 2.4645e-1 (1.04e-1) | - 3.0242e-1 (2.32e-1) | - 1.6827e-1 (5.88e-2) | ≈ 3.5034e-2 (2.24e-2) | + 2.7356e-1 (2.23e-1) | - 1.3952e-1 (2.79e-2) |
| MW14 | 3 | 15 | 2.1115e-1 (8.38e-4) | - 1.1190e-1 (4.66e-3) | + 1.0330e-1 (3.02e-3) | + 1.4001e-1 (1.20e-2) | ≈ 1.3078e-1 (1.05e-2) | + 9.8590e-2 (1.67e-3) | + 1.4576e-1 (1.32e-2) | ≈ 1.4211e-1 (1.15e-2) |
| + / - / ≈ | | | 10/3/1 | 8/6/0 | 1/9/4 | 1/11/2 | 1/10/3 | 14/0/0 | 0/12/2 | |

Figura 16 – Gráficos de Convergência dos MW.



Fonte: Autoria Própria (2024).

Tabela 8 – Média e desvio padrão de HV para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas MW.

| Problema | <i>M</i> | <i>D</i> | CMOEAD | CTAEA | CCMO | CMOEA_MS | DSPCMDE | BiCo | NSBiDiCo | P-NSBiDiCo |
|----------|----------|----------|---------------------|---------------------|---------------------|---------------------|----------------------|---------------------|---------------------|---------------------|
| MW1 | 2 | 15 | 4.8893e-1 (9.40e-5) | 4.8894e-1 (2.11e-4) | 4.8875e-1 (3.04e-4) | 4.6330e-1 (9.34e-3) | 4.8938e-1 (9.02e-5) | 4.9013e-1 (1.51e-5) | 4.8895e-1 (1.92e-4) | 4.8999e-1 (2.21e-5) |
| MW2 | 2 | 15 | 5.5669e-1 (1.36e-2) | 5.6352e-1 (1.14e-2) | 4.2084e-1 (8.86e-2) | 4.0793e-1 (8.87e-2) | 5.0039e-1 (3.52e-2) | 5.6422e-1 (1.15e-2) | 4.4682e-1 (5.50e-2) | 5.1048e-1 (2.06e-2) |
| MW3 | 2 | 15 | 5.4515e-1 (3.34e-4) | 5.4519e-1 (2.26e-4) | 5.4347e-1 (2.20e-4) | 5.4268e-1 (3.49e-4) | 5.4402e-1 (2.74e-4) | 5.4464e-1 (2.35e-4) | 5.4167e-1 (5.05e-4) | 5.4412e-1 (2.82e-4) |
| MW4 | 3 | 15 | 8.4167e-1 (2.88e-5) | 8.3814e-1 (1.98e-4) | 8.2070e-1 (1.97e-3) | 8.2173e-1 (1.68e-3) | 8.1314e-1 (6.64e-3) | 8.4143e-1 (4.50e-4) | 8.0535e-1 (3.86e-3) | 8.2354e-1 (2.54e-3) |
| MW5 | 2 | 15 | 3.2444e-1 (3.68e-4) | 3.1971e-1 (1.22e-3) | 3.2285e-1 (8.04e-4) | 2.8620e-1 (8.42e-2) | 2.9222e-1 (3.08e-2) | 3.2436e-1 (1.04e-4) | 1.9332e-1 (1.16e-1) | 3.2397e-1 (3.81e-4) |
| MW6 | 2 | 15 | 3.0212e-1 (2.43e-2) | 3.1041e-1 (1.02e-2) | 1.1901e-1 (6.97e-2) | 1.0161e-1 (4.06e-2) | 2.0465e-1 (5.47e-2) | 3.1633e-1 (1.15e-2) | 1.4033e-1 (7.74e-2) | 2.4245e-1 (4.43e-2) |
| MW7 | 2 | 15 | 4.1141e-1 (1.05e-4) | 4.0960e-1 (7.17e-4) | 4.1218e-1 (1.53e-4) | 3.4558e-1 (1.07e-2) | 4.1297e-1 (9.48e-5) | 4.1248e-1 (3.81e-4) | 4.0917e-1 (6.56e-4) | 4.1180e-1 (3.28e-4) |
| MW8 | 3 | 15 | 5.2407e-1 (1.72e-2) | 5.2645e-1 (1.08e-2) | 3.4856e-1 (8.25e-2) | 3.6825e-1 (1.15e-1) | 3.9526e-1 (7.77e-2) | 5.4678e-1 (6.01e-3) | 3.3127e-1 (7.67e-2) | 4.6823e-1 (1.88e-2) |
| MW9 | 2 | 15 | 3.9360e-1 (2.73e-3) | 3.9372e-1 (1.40e-3) | 3.5646e-1 (1.21e-1) | 1.4786e-1 (7.38e-2) | 2.7833e-1 (1.33e-1) | 3.9600e-1 (2.16e-3) | 3.8340e-1 (2.88e-3) | 3.9373e-1 (2.09e-3) |
| MW10 | 2 | 15 | 3.8825e-1 (7.42e-2) | 4.4219e-1 (1.31e-2) | 3.0351e-1 (5.49e-2) | 2.4806e-1 (6.60e-2) | 2.7689e-1 (8.39e-2) | 4.2587e-1 (2.26e-2) | 2.6194e-1 (7.89e-2) | 2.9800e-1 (7.62e-2) |
| MW11 | 2 | 15 | 4.4141e-1 (2.61e-2) | 4.4401e-1 (7.64e-4) | 4.4715e-1 (2.33e-4) | 4.4724e-1 (2.59e-4) | 3.1024e-1 (5.70e-17) | 4.4811e-1 (9.45e-5) | 4.4723e-1 (1.38e-4) | 4.4797e-1 (1.74e-4) |
| MW12 | 2 | 15 | 6.0565e-1 (1.08e-4) | 6.0094e-1 (7.44e-4) | 5.7436e-1 (1.28e-1) | 5.0146e-1 (1.13e-1) | 6.0246e-1 (8.96e-3) | 6.0531e-1 (1.26e-4) | 5.6448e-1 (1.31e-1) | 6.0490e-1 (2.12e-4) |
| MW13 | 2 | 15 | 4.4322e-1 (1.28e-2) | 4.5996e-1 (1.48e-2) | 3.3425e-1 (6.13e-2) | 3.1659e-1 (9.28e-2) | 3.9178e-1 (4.36e-2) | 4.6310e-1 (1.08e-2) | 3.3145e-1 (8.51e-2) | 4.0073e-1 (1.95e-2) |
| MW14 | 3 | 15 | 4.4618e-1 (1.06e-3) | 4.6825e-1 (1.93e-3) | 4.6336e-1 (2.35e-3) | 4.4980e-1 (4.10e-3) | 4.4981e-1 (4.34e-3) | 4.6969e-1 (1.54e-3) | 4.4094e-1 (5.54e-3) | 4.4995e-1 (4.40e-3) |
| | | | + / - / ≈ | 9/4/1 | 8/5/1 | 2/10/2 | 0/12/2 | 1/8/5 | 14/0/0 | 0/13/1 |

A seção a seguir apresenta os resultados obtidos nos testes em problemas do mundo real.

4.3 Resultados Experimentais em Problemas do Mundo Real

Os problemas do mundo real, RWMOPs, consistem em problemas de otimização de aplicações práticas do cotidiano. Estes problemas possuem dois ou três objetivos *M* e diferentes números de variáveis de decisão *D*. Além disso, estes problemas podem possuir diferentes complexidades e diferentes tipos de restrições de desigualdade ou de igualdade ou ambas.

Para este teste, foram selecionados problemas da base de problemas do mundo real proposta em (Kumar et al., 2021). Esta base inclui problemas de projeto mecânico, problemas de engenharia química e problemas de processos, design e síntese. Ao todo foram selecionados 23 problemas do mundo real.

A Tabela 9 mostra o desempenho do P-NSBiDiCo comparado aos demais algoritmos para os problemas RWMOP, contendo a média e o desvio padrão para a métrica do HV. As colunas *M* e *D* correspondem ao número de objetivos do problema e à dimensão do espaço de decisão. Os resultados destacados em cinza indicam os melhores resultados encontrados entre os algoritmos comparados.

Os resultados da Tabela 9 evidenciam a superioridade do P-NSBiDiCo em resolver problemas do mundo real. De 23 problemas, o P-NSBiDiCo obteve o melhor resultado em 13 (RWMOP3-RWMOP9, RWMOP10, RWMOP12-RWMOP14 E RWMOP21-RWMOP25). Ao analisar o perfil de convergência na Figura 17, fica ainda mais evidente a superioridade do P-NSBiDiCo em diversos problemas, como o RWMOP3, RWMOP5, RWMOP6 e RWMOP21.

Tabela 9 – Média e desvio padrão de HV para comparações de desempenho entre CMOEAD, CTAEA, CCMO, CMOEA-MS, DSPCMDE, BiCo, NSBiDiCo e P-NSBiDiCo para os problemas do mundo real.

| Problema | M | D | CMOEAD | CTAEA | CCMO | CMOEA_MS | DSPCMDE | BiCo | NSBiDiCo | P-NSBiDiCo |
|----------|-----|-----|-----------------------|-----------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|---------------------|
| RWMOP1 | 2 | 4 | 1.0813e-1 (2.89e-5) + | 6.0668e-1 (1.04e-3) + | 6.0817e-1 (2.75e-4) + | 5.8736e-1 (4.79e-3) + | 6.0752e-1 (3.11e-4) + | 6.0924e-1 (2.15e-4) + | 6.0785e-1 (2.87e-4) + | 9.0897e-2 (5.21e-6) |
| RWMOP2 | 2 | 5 | 3.3247e-1 (1.15e-1) ≈ | 2.3629e-1 (1.82e-1) - | 3.9289e-1 (1.54e-5) ≈ | 3.9282e-1 (2.67e-5) ≈ | 3.5366e-1 (1.21e-1) + | 3.4877e-1 (1.03e-1) ≈ | 3.9296e-1 (1.13e-5) ≈ | 3.2314e-1 (1.34e-1) |
| RWMOP3 | 2 | 3 | 9.7158e-2 (1.59e-2) - | 8.9598e-1 (2.13e-3) - | 8.9946e-1 (5.51e-4) - | 8.7179e-1 (1.56e-2) - | 9.0211e-1 (1.41e-4) - | 8.9973e-1 (3.62e-4) - | 9.0194e-1 (1.74e-4) - | 9.6736e-1 (8.48e-2) |
| RWMOP4 | 2 | 4 | 2.7230e-2 (6.53e-2) - | 8.5288e-1 (8.62e-3) - | 8.5872e-1 (1.32e-3) - | 4.9394e-1 (1.71e-1) - | 8.4969e-1 (8.07e-3) - | 8.6023e-1 (8.96e-4) - | 8.6272e-1 (2.14e-4) - | 8.9300e-1 (3.82e-2) |
| RWMOP5 | 2 | 4 | 4.2589e-1 (4.59e-3) - | 4.3277e-1 (1.36e-3) - | 4.3468e-1 (1.34e-4) - | 4.3534e-1 (6.80e-5) - | 2.7704e-1 (1.06e-4) - | 4.3494e-1 (8.30e-5) - | 4.3465e-1 (9.35e-5) - | 5.1932e-1 (1.14e-1) |
| RWMOP6 | 2 | 7 | 2.7650e-1 (5.33e-6) - | 2.2651e-1 (5.49e-2) - | 2.7642e-1 (1.67e-4) - | 2.7724e-1 (8.24e-6) - | 2.7062e-1 (2.01e-2) - | 2.7726e-1 (9.81e-6) - | 2.7711e-1 (2.14e-5) - | 5.4188e-1 (2.86e-1) |
| RWMOP7 | 2 | 4 | 4.8228e-1 (7.96e-4) - | 4.8396e-1 (2.43e-4) - | 4.8462e-1 (3.65e-5) - | 4.8451e-1 (4.46e-4) - | 4.8438e-1 (4.46e-5) - | 4.8461e-1 (3.67e-5) - | 4.8444e-1 (6.98e-5) - | 4.8500e-1 (2.15e-3) |
| RWMOP8 | 3 | 7 | 1.0542e-2 (2.36e-4) - | 2.6091e-2 (3.58e-5) - | 2.5977e-2 (4.37e-5) - | 2.6194e-2 (3.04e-5) - | 2.5877e-2 (9.70e-5) - | 2.6143e-2 (2.78e-5) - | 2.5912e-2 (7.46e-5) - | 4.8073e-2 (1.66e-2) |
| RWMOP9 | 2 | 4 | 5.3088e-2 (5.34e-5) - | 4.0871e-1 (5.23e-4) - | 4.0951e-1 (9.85e-5) - | 4.0894e-1 (2.16e-4) - | 4.0931e-1 (1.46e-4) - | 4.0985e-1 (9.26e-5) - | 4.0921e-1 (1.68e-4) - | 6.4596e-1 (2.93e-1) |
| RWMOP10 | 2 | 2 | 7.9706e-2 (4.02e-4) - | 8.4201e-1 (3.92e-3) ≈ | 8.4073e-1 (2.09e-3) ≈ | 8.3855e-1 (1.12e-3) ≈ | 8.4725e-1 (2.43e-4) + | 8.4750e-1 (3.63e-5) + | 8.4737e-1 (1.15e-4) + | 7.9584e-1 (2.00e-1) |
| RWMOP13 | 3 | 7 | 8.8166e-2 (3.40e-5) - | 8.5788e-2 (4.40e-3) - | 8.7239e-2 (1.90e-4) - | 3.4350e-2 (1.93e-2) - | 8.7835e-2 (8.55e-5) - | 8.7998e-2 (9.13e-5) - | 8.7748e-2 (7.25e-5) - | 2.1834e-1 (1.93e-1) |
| RWMOP14 | 2 | 5 | 1.0189e-1 (1.71e-2) - | 6.1777e-1 (4.74e-4) - | 6.1520e-1 (9.96e-4) - | 5.4902e-1 (1.20e-1) - | 5.8572e-1 (7.99e-2) - | 6.1821e-1 (4.78e-4) - | 6.1778e-1 (7.53e-4) - | 7.1687e-1 (1.56e-1) |
| RWMOP15 | 2 | 3 | 6.6015e-2 (9.41e-7) - | 5.4169e-1 (4.85e-3) - | 5.4362e-1 (1.34e-5) - | 4.9819e-1 (4.26e-2) - | 5.4356e-1 (2.26e-5) - | 5.4369e-1 (6.75e-5) + | 5.4368e-1 (9.06e-6) + | 5.4364e-1 (1.95e-5) |
| RWMOP16 | 2 | 2 | 7.9059e-2 (7.30e-6) - | 7.6261e-1 (5.79e-4) + | 7.6257e-1 (2.50e-4) + | 7.5495e-1 (1.69e-3) + | 7.6378e-1 (1.25e-4) + | 7.6286e-1 (1.91e-4) + | 7.6373e-1 (2.34e-4) + | 7.2412e-1 (2.02e-1) |
| RWMOP18 | 2 | 3 | 4.0339e-2 (6.78e-7) - | 4.0333e-2 (6.56e-5) - | 4.0513e-2 (2.78e-6) + | 4.0518e-2 (1.93e-6) + | 4.0499e-2 (5.18e-6) ≈ | 4.0521e-2 (1.53e-6) + | 4.0496e-2 (3.81e-6) + | 4.0499e-2 (2.84e-6) |
| RWMOP21 | 2 | 6 | 2.9321e-2 (3.26e-6) - | 3.1625e-2 (3.55e-5) - | 3.1761e-2 (5.76e-7) - | 3.1761e-2 (5.66e-7) - | 3.1757e-2 (1.14e-6) - | 3.1759e-2 (7.54e-7) - | 3.1757e-2 (1.15e-6) - | 1.7752e-1 (2.27e-1) |
| RWMOP22 | 2 | 9 | NaN (NaN) | NaN (NaN) | 7.3162e-1 (1.99e-1) ≈ | 5.6206e-1 (3.07e-1) - | NaN (NaN) | NaN (NaN) | 7.5130e-1 (2.07e-1) ≈ | 1.1783e+0 (1.73e+0) |
| RWMOP12 | 2 | 4 | 6.8251e-2 (4.91e-3) - | 5.4899e-1 (5.90e-3) - | 5.5624e-1 (2.05e-3) - | 4.7425e-1 (6.08e-2) - | 5.5799e-1 (2.34e-4) - | 5.5620e-1 (1.77e-3) - | 5.6057e-1 (2.20e-4) - | 6.5632e-1 (1.71e-1) |
| RWMOP25 | 2 | 2 | 2.3799e-1 (2.01e-4) ≈ | 2.4116e-1 (1.29e-4) ≈ | 2.4152e-1 (7.34e-6) ≈ | 2.3159e-1 (8.28e-6) ≈ | 2.4111e-1 (8.69e-5) ≈ | 2.4153e-1 (8.61e-6) ≈ | 2.4124e-1 (4.87e-5) ≈ | 5.2256e-1 (3.80e-1) |
| RWMOP26 | 2 | 3 | 1.3910e-1 (7.41e-3) - | 1.2496e-1 (1.38e-2) - | 1.4697e-1 (4.45e-4) + | 1.1713e-1 (1.44e-2) - | 1.4675e-1 (6.79e-5) + | 1.4603e-1 (2.71e-3) - | 1.4705e-1 (4.51e-5) + | 1.4624e-1 (6.62e-4) |
| RWMOP27 | 2 | 3 | 2.0463e+2 (2.19e+2) + | 4.7266e+2 (7.42e+2) + | 5.3163e+10 (1.50e+11) + | 1.1930e+10 (2.66e+10) + | 1.2211e+12 (4.09e+12) + | 5.8777e+11 (1.47e+12) + | 1.1697e+11 (3.89e+11) + | 9.9208e-1 (1.56e-2) |
| RWMOP28 | 2 | 7 | 2.4016e+2 (2.08e-2) ≈ | NaN (NaN) | 2.3093e-2 (2.01e-2) ≈ | 2.5665e-2 (1.08e-2) ≈ | NaN (NaN) | 3.5653e-2 (2.13e-2) ≈ | 3.1287e-2 (1.73e-2) ≈ | 3.1890e-2 (1.57e-2) |
| RWMOP29 | 2 | 7 | 7.8074e-1 (1.10e-2) - | 7.1020e-1 (4.38e-2) - | 7.8644e-1 (7.81e-4) ≈ | 7.8659e-1 (8.29e-4) ≈ | NaN (NaN) | 7.8717e-1 (4.86e-4) + | 7.8708e-1 (1.90e-4) ≈ | 7.8670e-1 (8.55e-4) |
| | | | + / - / ≈ | 2/17/3 | 3/16/2 | 5/12/6 | 4/14/5 | 6/12/2 | 7/12/3 | 6/12/5 |

4.4 Estudo de Ablation

O estudo *ablation* apresentado nesta seção tem como objetivo analisar como diferentes modificações no modelo base, o NSBiDiCo, afetam o seu desempenho geral. Para isso, foram realizadas variações específicas do modelo, incluindo componentes-chave, como a predição e ajustes de parâmetros.

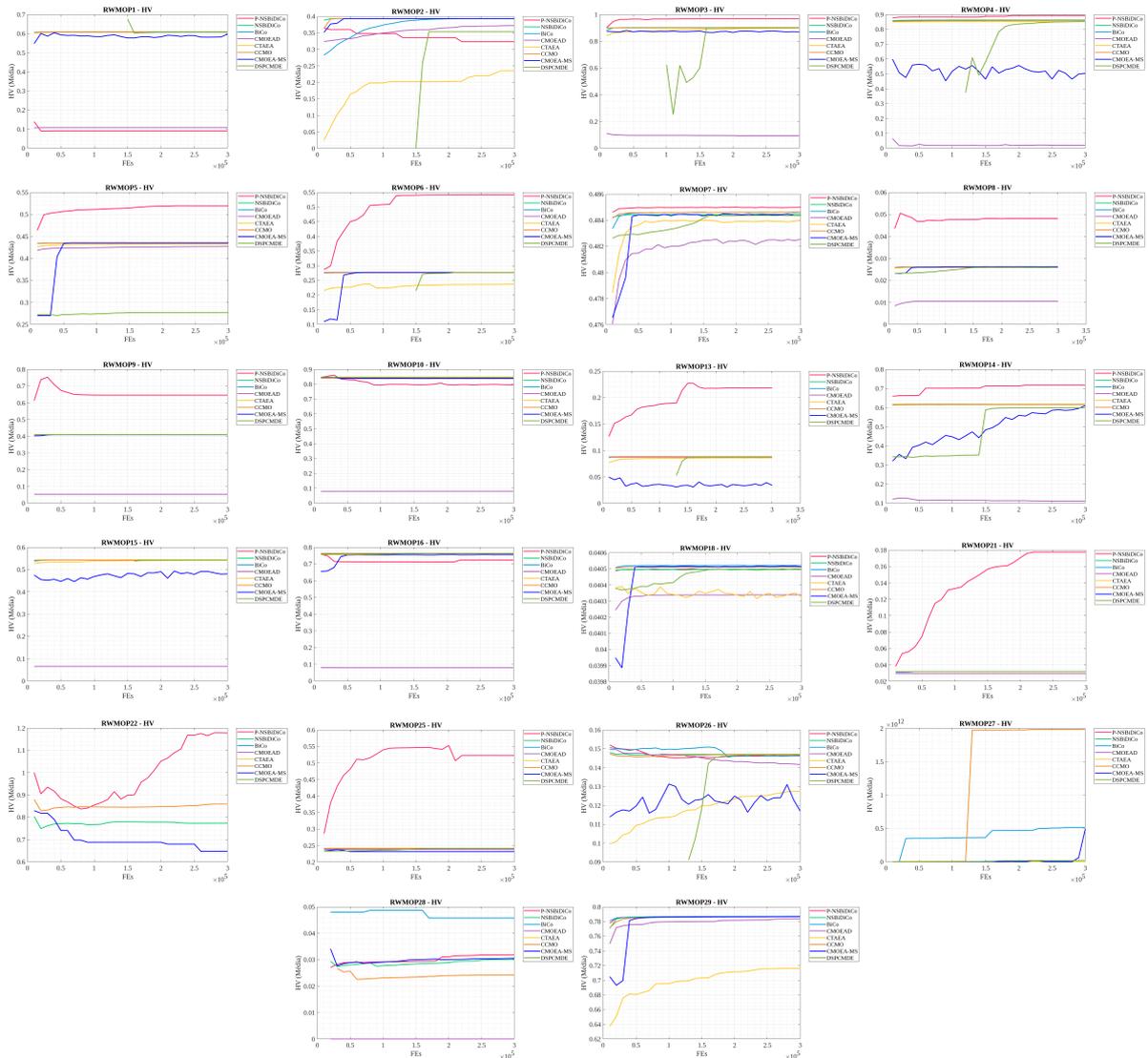
Essas modificações foram selecionadas devido a sua importância e impacto no desempenho do modelo com o uso do algoritmo de evolução diferencial e à potencial melhoria na convergência e exploração de espaço de soluções. O impacto dessas modificações foi avaliado por meio da métrica IGD em relação ao desempenho do NSBiDiCo em problemas de *benchmark*.

4.4.1 Variações do Modelo

As variações do modelo analisadas neste estudo incluem:

- **NSBiDiCo-V1:** Modelo base sem alterações utilizado como ponto de partida.
- **NSBiDiCo-V2:** Esta variação incorpora ajuste automático dos parâmetros F e Cr da evolução diferencial, visando aumentar a capacidade do algoritmo em se adaptar a diferentes características do espaço de busca do problema ao longo das gerações.
- **NSBiDiCo-V3:** Modelo NSBiDiCo-V2 que implementa adicionalmente a estratégia de predição no espaço objetivo e espaço de decisão.

Figura 17 – Convergência média dos modelos para o conjunto de problemas do mundo real - RWMOPs.



Fonte: Autoria Própria (2024).

Os testes para verificar o desempenho das variações do modelo foram realizados em 46 problemas de *benchmark*. Foram realizadas 20 execuções independentes para cada problema, com um limite de 3×10^5 avaliações de função objetivo.

Para avaliar as diferenças estatísticas entre o desempenho dos modelos utilizou-se o teste de soma de *ranks* de Wilcoxon (Wilcoxon, 1945) com nível de significância de 0,05. Os símbolos +, – e \approx na Tabela 10 e Tabela 11 apresentadas na Subseção 4.4.2, a seguir, indicam que o resultado encontrado pelo modelo são, respectivamente, melhor, pior, ou estatisticamente equivalente ao obtido pelo NSBiDiCo-V1 e NSBiDiCo-V3, respectivamente.

A subseção a seguir apresenta os resultados obtidos para cada variação do modelo.

4.4.2 Análise das Variações dos Modelos

A Tabela 10 mostra a comparação entre o NSBiDiCo-V1 e o NSBiDiCo-V2. Os resultados indicam que a inclusão da estratégia de ajuste automático dos parâmetros F e Cr resultou em uma melhora significativa no desempenho do modelo, especialmente em problemas da base MW.

A Tabela 11 mostra a comparação entre o NSBiDiCo-V2 e o NSBiDiCo-V3. Os resultados indicam que a inclusão da estratégia de predição no espaço objetivo e espaço de decisão resultou em uma melhora adicional no desempenho do modelo. Os gráficos de convergência média para as variações NSBiDiCo-V1 e NSBiDiCo-V3 são apresentados nas Figuras 18, 19, 20 e 21. Pode-se observar que o NSBiDiCo-V3 apresenta uma convergência mais rápida em comparação ao NSBiDiCo-V1 em diversos problemas testados, tais como o LIRCMOP6-8, LIRCMOP13-14, DASCMP1-3, DOC3-4, DOC6, DOC9 e MW1-14. Além disso, nos testes realizados, constatou-se que em 25 dos 46 problemas analisados, o NSBiDiCo-V3 apresentou uma convergência significativamente mais rápida que o NSBiDiCo-V1.

Dessa forma, os resultados obtidos sugerem que a inclusão de estratégias de ajuste automático dos parâmetros F e Cr , aliada a predição no espaço objetivo e espaço de decisão podem melhorar o desempenho do modelo NSBiDiCo em problemas de otimização multiobjetivo com restrições. Assim, o modelo NSBiDiCo-V3 apresenta o melhor desempenho em relação aos demais modelos testados e por este motivo foi selecionado como modelo final e chamado P-NSBiDiCo. Os resultados do estudo de *ablation* reforçam a importância das estratégias introduzidas no NSBiDiCo, evidenciando melhor capacidade de convergência e eficiência em problemas complexos, tornando-o uma solução promissora para aplicações práticas.

4.5 Considerações do Capítulo

Este capítulo apresentou os resultados comparativos do algoritmo proposto, o P-NSBiDiCo, com sete algoritmos do estado da arte. O modelo proposto demonstrou ser uma ferramenta promissora para a resolução de problemas de otimização multiobjetivo com restrições, especialmente em cenários do mundo real.

Os testes realizados nas diferentes bases de dados revelaram um desempenho variável do P-NSBiDiCo. Nas aplicações práticas do mundo real, o algoritmo demonstrou resultados significativos, superando as expectativas em termos de eficiência e eficácia. Essa superioridade pode ser atribuída à sua capacidade de adaptar-se a dados complexos e ruidosos, que são muito comuns em cenários reais, o que resulta em soluções mais robustas e aplicáveis.

Figura 18 – Comparação da Convergência Média entre as Variações dos Modelo – NSBiDiCo-V1 vs NSBiDiCo-V3 para LIRC MOP.

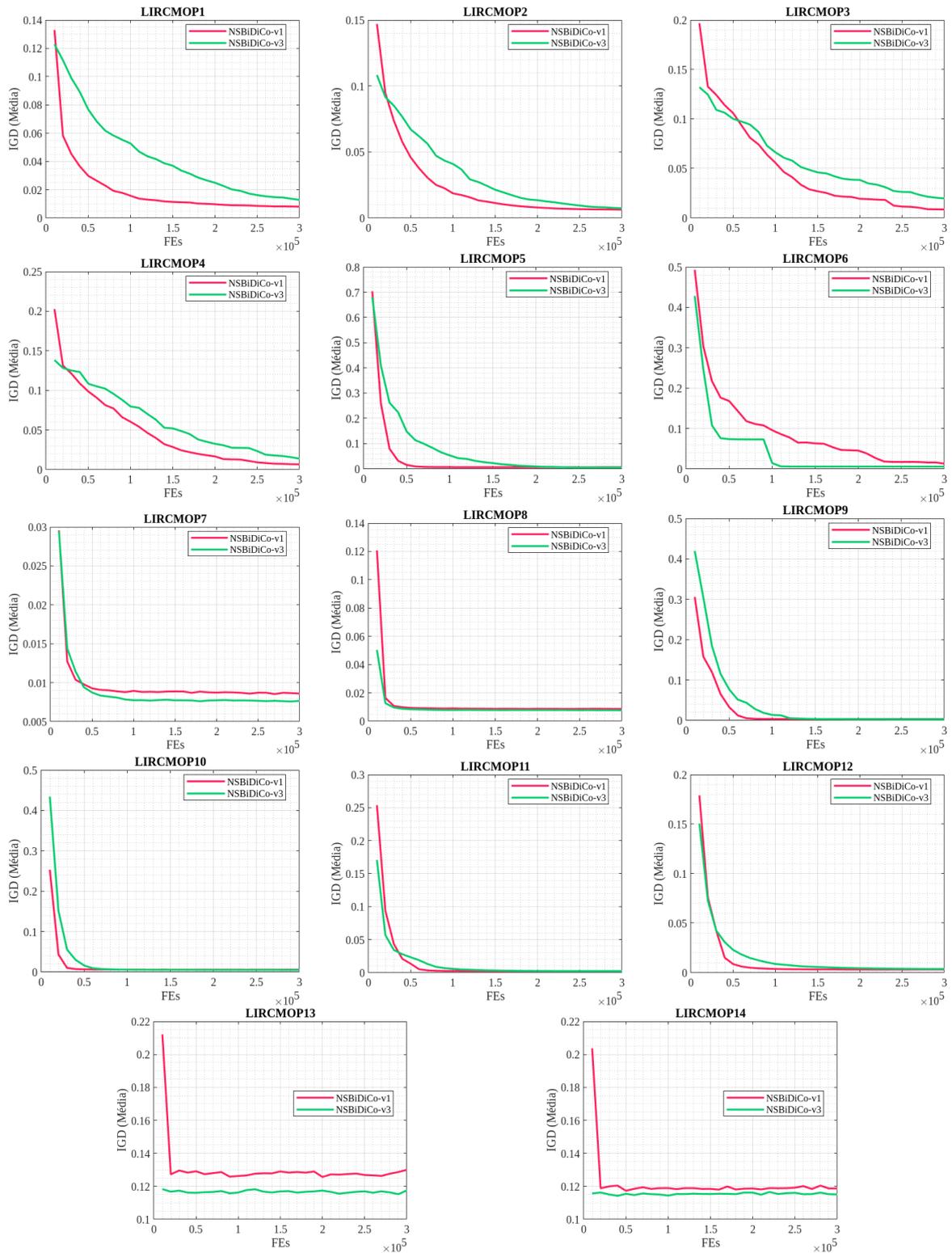
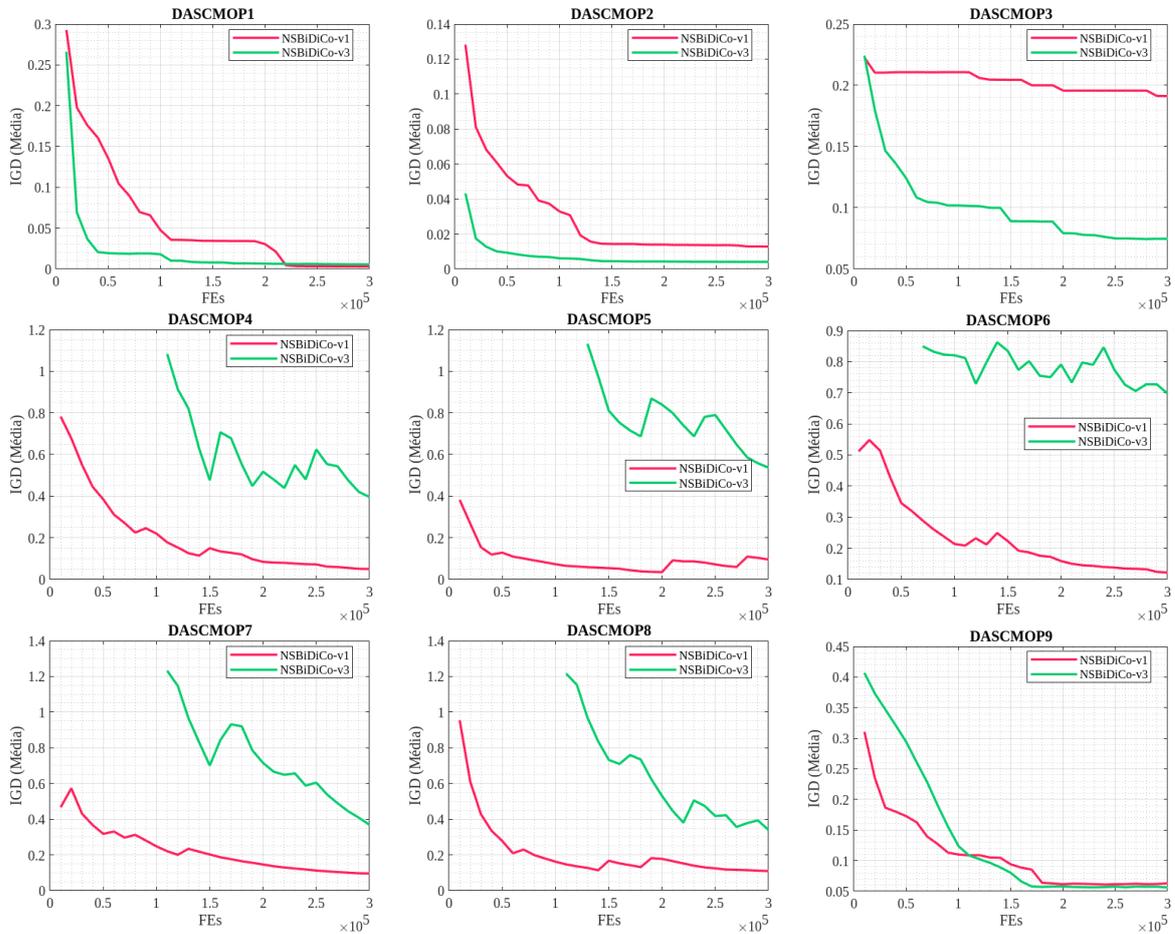


Tabela 10 – Comparação do Desempenho do NSBiDiCo-V1 e NSBiDiCo-V2 em Diversos Problemas de *Benchmark* Multiobjetivo com Restrições.

| Problema | M | D | NSBiDiCo-V2 | NSBiDiCo-V1 |
|-----------|-----|-----|-------------------------------|---------------------|
| LIRCMOP1 | 2 | 10 | 1.1752e-2 (3.37e-3) – | 8.0311e-3 (9.16e-4) |
| LIRCMOP2 | 2 | 10 | 8.4141e-3 (2.03e-3) – | 6.3407e-3 (6.21e-4) |
| LIRCMOP3 | 2 | 10 | 2.2249e-2 (1.18e-2) – | 8.6873e-3 (7.11e-3) |
| LIRCMOP4 | 2 | 10 | 2.4761e-2 (1.62e-2) – | 6.6131e-3 (4.71e-3) |
| LIRCMOP5 | 2 | 10 | 7.2831e-2 (2.55e-1) – | 6.6582e-3 (3.78e-4) |
| LIRCMOP6 | 2 | 10 | 5.6160e-3 (2.02e-4) + | 1.3077e-2 (3.01e-2) |
| LIRCMOP7 | 2 | 10 | 7.5819e-3 (2.31e-4) + | 8.6048e-3 (3.23e-4) |
| LIRCMOP8 | 2 | 10 | 7.6597e-3 (2.10e-4) + | 8.6347e-3 (3.69e-4) |
| LIRCMOP9 | 2 | 10 | 2.8621e-3 (8.83e-5) + | 3.4273e-3 (1.29e-4) |
| LIRCMOP10 | 2 | 10 | 4.9035e-3 (2.03e-4) + | 5.7280e-3 (1.66e-4) |
| LIRCMOP11 | 2 | 10 | 2.3773e-3 (5.65e-5) + | 2.4334e-3 (5.21e-5) |
| LIRCMOP12 | 2 | 10 | 3.6816e-3 (2.58e-4) – | 3.1654e-3 (1.21e-4) |
| LIRCMOP13 | 3 | 10 | 1.1614e-1 (2.92e-3) + | 1.2998e-1 (5.11e-3) |
| LIRCMOP14 | 3 | 10 | 1.1626e-1 (4.27e-3) \approx | 1.1869e-1 (4.82e-3) |
| DASCMP1 | 2 | 30 | 3.7857e-2 (7.56e-2) – | 3.2275e-3 (1.50e-4) |
| DASCMP2 | 2 | 30 | 2.1083e-2 (3.69e-2) \approx | 1.2837e-2 (3.50e-2) |
| DASCMP3 | 2 | 30 | 2.7968e-1 (3.82e-2) \approx | 1.9126e-1 (1.38e-1) |
| DASCMP4 | 2 | 30 | 4.2865e-1 (1.83e-1) – | 4.9798e-2 (1.00e-1) |
| DASCMP5 | 2 | 30 | 4.5151e-1 (2.44e-1) – | 9.5313e-2 (2.34e-1) |
| DASCMP6 | 2 | 30 | 6.8632e-1 (8.05e-2) – | 1.2179e-1 (2.13e-1) |
| DASCMP7 | 3 | 30 | 3.3956e-1 (1.61e-1) – | 9.5598e-2 (1.01e-1) |
| DASCMP8 | 3 | 30 | 2.3522e-1 (3.15e-1) \approx | 1.0915e-1 (1.17e-1) |
| DASCMP9 | 3 | 30 | 5.7365e-2 (2.70e-3) + | 6.3035e-2 (3.34e-3) |
| DOC1 | 2 | 6 | 7.1830e-3 (4.92e-4) – | 6.3701e-3 (6.32e-4) |
| DOC2 | 2 | 16 | 4.9760e-1 (0.00e+0) \approx | 3.3399e-2 (1.32e-1) |
| DOC3 | 2 | 10 | 2.4162e+2 (1.15e+2) + | 3.5149e+2 (1.22e+2) |
| DOC4 | 2 | 8 | 2.4694e-2 (4.55e-3) – | 1.5848e-2 (2.19e-3) |
| DOC5 | 2 | 8 | 6.2911e+1 (6.60e+1) – | 4.8966e+1 (6.53e+1) |
| DOC6 | 2 | 11 | 9.9637e-3 (2.75e-2) \approx | 3.6498e-3 (2.22e-4) |
| DOC7 | 2 | 11 | 1.2153e+0 (1.20e+0) – | 3.0581e-3 (1.51e-4) |
| DOC8 | 3 | 10 | 3.6414e+0 (3.76e+0) – | 1.0764e-1 (6.74e-3) |
| DOC9 | 3 | 11 | 1.3624e-1 (4.56e-2) + | 1.3918e-1 (2.00e-2) |
| MW1 | 2 | 15 | 1.8595e-3 (5.60e-5) + | 2.3402e-3 (1.06e-4) |
| MW2 | 2 | 15 | 4.6011e-2 (1.24e-2) + | 1.1525e-1 (5.93e-2) |
| MW3 | 2 | 15 | 4.7838e-3 (1.08e-4) + | 6.3319e-3 (2.85e-4) |
| MW4 | 3 | 15 | 5.2923e-2 (1.95e-3) + | 6.6624e-2 (2.00e-3) |
| MW5 | 2 | 15 | 1.2719e-3 (9.42e-4) + | 3.0551e-1 (3.50e-1) |
| MW6 | 2 | 15 | 8.7513e-2 (9.58e-2) + | 4.6361e-1 (2.35e-1) |
| MW7 | 2 | 15 | 4.5998e-3 (1.69e-4) + | 5.9954e-3 (2.74e-4) |
| MW8 | 3 | 15 | 8.3089e-2 (1.60e-2) + | 1.6865e-1 (5.26e-2) |
| MW9 | 2 | 15 | 4.8124e-3 (4.22e-4) + | 9.8652e-3 (2.19e-3) |
| MW10 | 2 | 15 | 3.0571e-1 (2.26e-1) \approx | 3.0963e-1 (1.98e-1) |
| MW11 | 2 | 15 | 6.0466e-3 (1.45e-4) + | 7.1197e-3 (2.39e-4) |
| MW12 | 2 | 15 | 4.8160e-3 (1.40e-4) + | 5.9665e-3 (4.52e-4) |
| MW13 | 2 | 15 | 1.3106e-1 (2.02e-2) + | 2.6798e-1 (1.57e-1) |
| MW14 | 3 | 15 | 1.4004e-1 (9.14e-3) \approx | 1.4600e-1 (1.37e-2) |
| | | | + / - / \approx | 22/16/8 |

Em contrapartida, quando o P-NSBiDiCo foi avaliado em funções de benchmark, o algoritmo apresentou um desempenho insatisfatório. As limitações encontradas durante esses testes podem ser atribuídas à estrutura controlada das funções de benchmark, que podem não refletir adequadamente os desafios do mundo real ou mesmo

Figura 19 – Comparação da Convergência Média entre as Variações dos Modelo – NSBiDiCo-V1 vs NSBiDiCo-V3 para DASC MOP.



a sensibilidade a configuração dos parâmetros atrelado a natureza do problema. Essa discrepância ressalta a importância em avaliar os algoritmos em contextos práticos do mundo real para obter uma visão mais precisa de sua aplicabilidade e desempenho.

Figura 20 – Comparação da Convergência Média entre as Variações dos Modelo – NSBiDiCo-V1 vs NSBiDiCo-V3 para DOC.

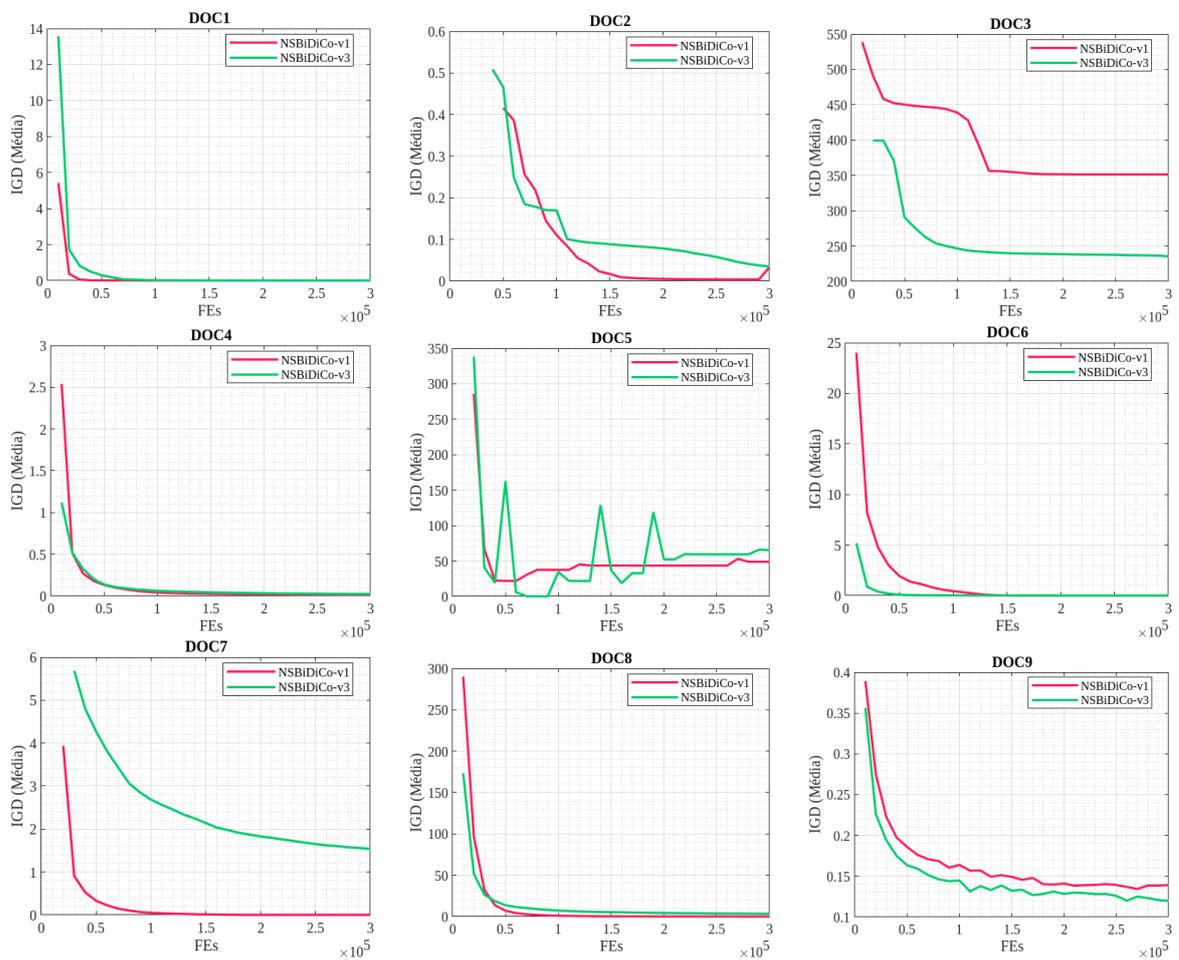


Figura 21 – Comparação da Convergência Média entre as Variações dos Modelo – NSBiDiCo-V1 vs NSBiDiCo-V3 para MW.

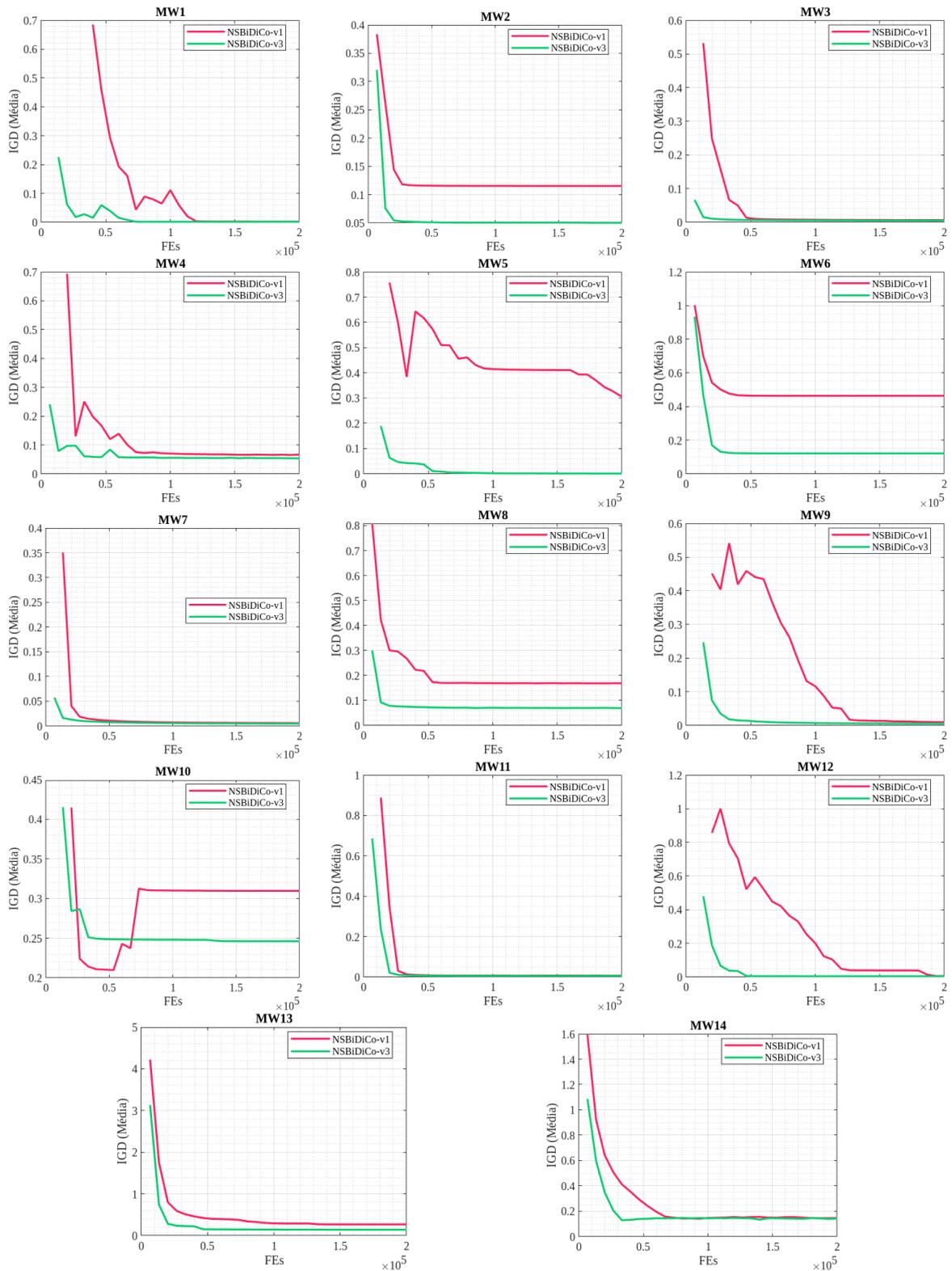


Tabela 11 – Comparação do Desempenho do NSBiDiCo-V2 e NSBiDiCo-V3 em Diversos Problemas de *Benchmark* Multiobjetivo com Restrições.

| Problema | M | D | NSBiDiCo-V2 | NSBiDiCo-V3 |
|-------------------|-----|-----|-------------------------------|---------------------|
| LIRCMOP1 | 2 | 10 | 1.1752e-2 (3.37e-3) \approx | 1.2794e-2 (3.67e-3) |
| LIRCMOP2 | 2 | 10 | 8.4141e-3 (2.03e-3) \approx | 7.3888e-3 (1.17e-3) |
| LIRCMOP3 | 2 | 10 | 2.2249e-2 (1.18e-2) \approx | 1.9790e-2 (9.46e-3) |
| LIRCMOP4 | 2 | 10 | 2.4761e-2 (1.62e-2) $-$ | 1.3674e-2 (9.55e-3) |
| LIRCMOP5 | 2 | 10 | 7.2831e-2 (2.55e-1) \approx | 5.7425e-3 (3.69e-4) |
| LIRCMOP6 | 2 | 10 | 5.6160e-3 (2.02e-4) \approx | 5.5941e-3 (1.37e-4) |
| LIRCMOP7 | 2 | 10 | 7.5819e-3 (2.31e-4) \approx | 7.6778e-3 (2.38e-4) |
| LIRCMOP8 | 2 | 10 | 7.6597e-3 (2.10e-4) \approx | 7.6311e-3 (2.42e-4) |
| LIRCMOP9 | 2 | 10 | 2.8621e-3 (8.83e-5) $+$ | 2.9663e-3 (9.52e-5) |
| LIRCMOP10 | 2 | 10 | 4.9035e-3 (2.03e-4) \approx | 5.0223e-3 (1.81e-4) |
| LIRCMOP11 | 2 | 10 | 2.3773e-3 (5.65e-5) \approx | 2.3710e-3 (8.22e-5) |
| LIRCMOP12 | 2 | 10 | 3.6816e-3 (2.58e-4) \approx | 3.7163e-3 (5.70e-4) |
| LIRCMOP13 | 3 | 10 | 1.1614e-1 (2.92e-3) \approx | 1.1749e-1 (4.72e-3) |
| LIRCMOP14 | 3 | 10 | 1.1626e-1 (4.27e-3) \approx | 1.1510e-1 (3.75e-3) |
| DASCMOP1 | 2 | 30 | 3.7857e-2 (7.56e-2) $-$ | 5.7121e-3 (2.29e-3) |
| DASCMOP2 | 2 | 30 | 2.1083e-2 (3.69e-2) $-$ | 4.0770e-3 (8.82e-5) |
| DASCMOP3 | 2 | 30 | 2.7968e-1 (3.82e-2) $-$ | 7.4595e-2 (6.36e-2) |
| DASCMOP4 | 2 | 30 | 4.2865e-1 (1.83e-1) \approx | 3.9514e-1 (5.96e-2) |
| DASCMOP5 | 2 | 30 | 4.5151e-1 (2.44e-1) \approx | 5.3662e-1 (2.30e-1) |
| DASCMOP6 | 2 | 30 | 6.8632e-1 (8.05e-2) \approx | 6.9759e-1 (8.06e-2) |
| DASCMOP7 | 3 | 30 | 3.3956e-1 (1.61e-1) \approx | 3.6796e-1 (2.48e-1) |
| DASCMOP8 | 3 | 30 | 2.3522e-1 (3.15e-1) \approx | 3.3899e-1 (3.61e-1) |
| DASCMOP9 | 3 | 30 | 5.7365e-2 (2.70e-3) \approx | 5.6162e-2 (3.22e-3) |
| DOC1 | 2 | 6 | 7.1830e-3 (4.92e-4) \approx | 7.1169e-3 (5.50e-4) |
| DOC2 | 2 | 16 | 4.9760e-1 (0.00e+0) \approx | 3.5178e-2 (0.00e+0) |
| DOC3 | 2 | 10 | 2.4162e+2 (1.15e+2) \approx | 2.3558e+2 (1.33e+2) |
| DOC4 | 2 | 8 | 2.4694e-2 (4.55e-3) \approx | 2.4950e-2 (5.29e-3) |
| DOC5 | 2 | 8 | 6.2911e+1 (6.60e+1) \approx | 6.5389e+1 (6.83e+1) |
| DOC6 | 2 | 11 | 9.9637e-3 (2.75e-2) \approx | 1.3695e-2 (3.34e-2) |
| DOC7 | 2 | 11 | 1.2153e+0 (1.20e+0) \approx | 1.5399e+0 (1.03e+0) |
| DOC8 | 3 | 10 | 3.6414e+0 (3.76e+0) \approx | 3.4334e+0 (3.61e+0) |
| DOC9 | 3 | 11 | 1.3624e-1 (4.56e-2) \approx | 1.1982e-1 (1.49e-2) |
| MW1 | 2 | 15 | 1.8595e-3 (5.60e-5) \approx | 1.8278e-3 (3.78e-5) |
| MW2 | 2 | 15 | 4.6011e-2 (1.24e-2) \approx | 5.0227e-2 (1.50e-2) |
| MW3 | 2 | 15 | 4.7838e-3 (1.08e-4) $+$ | 4.9781e-3 (1.69e-4) |
| MW4 | 3 | 15 | 5.2923e-2 (1.95e-3) \approx | 5.3658e-2 (1.69e-3) |
| MW5 | 2 | 15 | 1.2719e-3 (9.42e-4) \approx | 1.0901e-3 (5.90e-4) |
| MW6 | 2 | 15 | 8.7513e-2 (9.58e-2) \approx | 1.2125e-1 (1.51e-1) |
| MW7 | 2 | 15 | 4.5998e-3 (1.69e-4) $+$ | 5.0003e-3 (3.44e-4) |
| MW8 | 3 | 15 | 8.3089e-2 (1.60e-2) $-$ | 6.9274e-2 (7.81e-3) |
| MW9 | 2 | 15 | 4.8124e-3 (4.22e-4) $+$ | 5.3685e-3 (8.24e-4) |
| MW10 | 2 | 15 | 3.0571e-1 (2.26e-1) \approx | 2.4595e-1 (1.57e-1) |
| MW11 | 2 | 15 | 6.0466e-3 (1.45e-4) \approx | 6.0678e-3 (1.25e-4) |
| MW12 | 2 | 15 | 4.8160e-3 (1.40e-4) \approx | 4.8804e-3 (1.18e-4) |
| MW13 | 2 | 15 | 1.3106e-1 (2.02e-2) \approx | 1.3952e-1 (2.79e-2) |
| MW14 | 3 | 15 | 1.4004e-1 (9.14e-3) \approx | 1.4211e-1 (1.15e-2) |
| + / - / \approx | | | 4/5/37 | |

5

Conclusão

O estudo apresentado neste documento propõe um novo algoritmo evolucionário para tratar problemas multiobjetivo com restrições (CMOPs). A abordagem proposta, denominada P-NSBiDiCo, avança o algoritmo NSBiDiCo (Mendes; Araújo; Farias, 2023), um algoritmo de evolução diferencial bidirecional de ordenamento por não-dominância. A nova abordagem emprega um novo mecanismo de predição no espaço objetivo e de decisão para identificar soluções promissoras no espaço de busca. O algoritmo também emprega uma nova técnica de seleção dos parâmetros F e CR da evolução diferencial por meio da utilização de distribuições de probabilidade.

Para validar o método proposto, foram realizados testes em 46 funções de benchmark e em 23 problemas do mundo real. A análise dos resultados experimentais revela um cenário complexo e com alguns *insights* sobre o desempenho do algoritmo P-NSBiDiCo. Ao comparar o modelo proposto com os algoritmos de estado da arte em uma variedade de problemas, observamos um desempenho promissor em diversas situações, ao mesmo tempo em que identificaram-se algumas limitações.

Um dos aspectos mais notáveis do P-NSBiDiCo é, sem dúvidas, a sua capacidade de resolver problemas do mundo real com alta eficiência. Os resultados obtidos nas bases de dados RWMOP demonstram a superioridade do algoritmo em encontrar soluções de alta qualidade para problemas práticos, superando os demais algoritmos em diversos cenários. Essa performance superior pode ser atribuída à combinação das estratégias de seleção de parâmetros e do mecanismo de predição, que se mostraram eficazes em lidar com a complexidade e a variabilidade dos problemas do mundo real.

No entanto, ao analisar os resultados obtidos nas bases de benchmarks, observa-se um desempenho menos expressivo do P-NSBiDiCo em comparação com alguns algoritmos específicos, como o BiCo. Essa discrepância pode ser explicada por diversos fatores, incluindo:

- **Natureza dos benchmarks:** As funções de benchmark são projetadas para avaliar o desempenho dos algoritmos em condições extremas e idealizadas, com características específicas que podem não ser representativas de problemas reais.
- **Configuração dos parâmetros:** A configuração dos parâmetros dos algoritmos pode ter um impacto significativo no desempenho, e é possível que a configuração utilizada neste estudo não seja a ideal para todos os benchmarks.
- **Complexidade das funções:** Algumas funções de benchmark apresentam características que podem ser mais desafiadoras para o P-NSBiDiCo, como regiões de viabilidade complexas e fronteiras de Pareto altamente não-convexas.

5.1 Considerações

É importante ressaltar que o desempenho em benchmarks não deve ser o único critério para avaliar a qualidade final de um algoritmo. A capacidade de resolver problemas do mundo real, que são frequentemente mais complexos e menos bem definidos do que os benchmarks, é um indicador mais relevante da utilidade prática de um algoritmo.

5.1.1 Pontos Fortes do P-NSBiDiCo

Considerando os resultados experimentais obtidos, destaca-se os seguintes pontos fortes:

- **Eficiência em problemas do mundo real:** O algoritmo demonstrou que possui capacidade superior em encontrar soluções de alta qualidade para problemas práticos.
- **Robustez:** O P-NSBiDiCo se mostrou-se robusto em diferentes tipos de problemas, com diferentes números de objetivos e variáveis de decisão.
- **Combinação de estratégias:** A combinação das estratégias de seleção de parâmetros e predição no espaço objetivo e de decisão contribuíram para o bom desempenho do algoritmo.

5.1.2 Limitações

Pôde-se identificar algumas limitações, como:

- **Desempenho em alguns benchmarks:** O algoritmo apresentou dificuldades em alguns benchmarks específicos, especialmente aqueles com características mais extremas.

- **Sensibilidade a parâmetros:** A performance do algoritmo pode ser sensível à configuração dos parâmetros, o que exige mais estudos para um ajuste cuidadoso para cada problema.

5.1.3 Trabalhos Futuros

Como trabalhos futuros, pretende-se:

- **Explorar novas estratégias de predição:** Investigar o uso de diferentes modelos de aprendizado de máquina para melhorar a precisão e a velocidade das predições.
- **Adaptar os parâmetros de forma dinâmica:** Desenvolver novos mecanismos para ajustar os parâmetros do algoritmo de forma adaptativa durante a execução, a fim de melhorar o desempenho em problemas complexos.
- **Avaliar o desempenho em outras bases de benchmarks:** Expandir a avaliação do algoritmo para outras bases de dados, com o objetivo de identificar novas oportunidades de melhoria.

Em conclusão, o P-NSBiDiCo representa uma contribuição significativa para o campo da otimização multiobjetivo com restrições. Embora existam ainda oportunidades para melhoria, os resultados obtidos demonstram o potencial do algoritmo para resolver problemas reais de grande complexidade.

5.2 Publicação

5.2.1 Conferência Internacional – *Qualis A2*

Como parte do desenvolvimento desta pesquisa, foi publicado o seguinte trabalho:

- **"Non-Dominated Sorting Bidirectional Differential Coevolution"** dos autores *Cicero S. R. Mendes, Aluizio F. R. Araújo, e Lucas R. C. Farias* na **2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)** realizada em Honolulu, Hawaii, EUA, de 1-4 de Outubro de 2023.

Referências

- Angira, R.; Babu, B. Non-dominated sorting differential evolution (nsde): An extension of differential evolution for multi-objective optimization. In: **IICAI**. 2005. p. 1428–1443.
- Bao, Q.; Wang, M.; Dai, G.; Chen, X.; Song, Z.; Li, S. An archive-based two-stage evolutionary algorithm for constrained multi-objective optimization problems. **Swarm and Evolutionary Computation**, v. 75, p. 101161, 2022.
- Bosman, P.; Thierens, D. The balance between proximity and diversity in multiobjective evolutionary algorithms. **IEEE Transactions on Evolutionary Computation**, v. 7, n. 2, p. 174–188, 2003.
- Deb, K.; Goyal, M. A combined genetic adaptive search (geneas) for engineering design. **Computer Science and informatics**, COMPUTER SOCIETY OF INDIA, v. 26, p. 30–45, 1996.
- 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.
- Fan, Z.; Li, W.; Cai, X.; Huang, H.; Fang, Y.; You, Y.; Mo, J.; Wei, C.; Goodman, E. An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. **Soft Computing**, v. 23, n. 23, p. 12491–12510, 2019.
- Fan, Z.; Li, W.; Cai, X.; Li, H.; Wei, C.; Zhang, Q.; Deb, K.; Goodman, E. Push and pull search for solving constrained multi-objective optimization problems. **Swarm and Evolutionary Computation**, v. 44, p. 665–679, 2019.
- Fan, Z.; Li, W.; Cai, X.; Li, H.; Wei, C.; Zhang, Q.; Deb, K.; Goodman, E. Difficulty Adjustable and Scalable Constrained Multiobjective Test Problem Toolkit. **Evolutionary Computation**, v. 28, n. 3, p. 339–378, 2020.
- Farzin, H.; Fotuhi-Firuzabad, M.; Moeini-Aghtaie, M. A stochastic multi-objective framework for optimal scheduling of energy storage systems in microgrids. **IEEE Transactions on Smart Grid**, v. 8, n. 1, p. 117–127, 2017.
- Feng, X.; Ren, Z.; Pan, A.; Hong, J.; Tong, Y. A multi-preference-based constrained multi-objective optimization algorithm. **Swarm and Evolutionary Computation**, v. 83, p. 101389, 2023.
- Gu, Q.; Liu, R.; Hui, Z.; Wang, D. A constrained multi-objective optimization algorithm based on coordinated strategy of archive and weight vectors. **Expert Systems with Applications**, v. 244, p. 122961, 2024.
- Guerrero-Peña, E.; Araújo, A. F. R. Multi-objective evolutionary algorithm with prediction in the objective space. **Information Sciences**, v. 501, p. 293–316, 2019.
- Harada, T. Parallel cooperative multiobjective coevolutionary algorithm for constrained multiobjective optimization problems. **Applied Soft Computing**, v. 153, p. 111290, 2024.

- Hu, Z.; Gong, W.; Pedrycz, W.; Li, Y. Deep reinforcement learning assisted co-evolutionary differential evolution for constrained optimization. **Swarm and Evolutionary Computation**, v. 83, p. 101387, 2023.
- Huang, W.; Zou, J.; Tang, H.; Zheng, J.; Yu, F. Enhanced auxiliary population search for diversity improvement of constrained multiobjective coevolutionary optimization. **Swarm and Evolutionary Computation**, v. 83, p. 101404, 2023.
- Jain, H.; Deb, K. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. **IEEE Transactions on Evolutionary Computation**, v. 18, n. 4, p. 602–622, 2014.
- Jin, Y.; Branke, J. Evolutionary optimization in uncertain environments—a survey. **IEEE Transactions on Evolutionary Computation**, v. 9, n. 3, p. 303–317, 2005.
- Jozefowicz, N.; Semet, F.; Talbi, E.-G. Multi-objective vehicle routing problems. **European Journal of Operational Research**, v. 189, n. 2, p. 293–309, 2008.
- Kumar, A.; Wu, G.; Ali, M. Z.; Luo, Q.; Mallipeddi, R.; Suganthan, P. N.; Das, S. A Benchmark-Suite of real-World constrained multi-objective optimization problems and some baseline results. **Swarm and Evolutionary Computation**, v. 67, p. 100961, 2021.
- Li, G.; Wang, Z.; Gao, W.; Wang, L. Decoupling Constraint: Task Clone-Based Multi-Tasking Optimization for Constrained Multi-Objective Optimization. **IEEE Transactions on Evolutionary Computation**, 2024. Acesso antecipado.
- Li, K.; Chen, R.; Fu, G.; Yao, X. Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization. **IEEE Transactions on Evolutionary Computation**, v. 23, n. 2, p. 303–315, 2019.
- Liang, J.; Ban, X.; Yu, K.; Qu, B.; Qiao, K.; Yue, C.; Chen, K.; Tan, K. C. A Survey on Evolutionary Constrained Multiobjective Optimization. **IEEE Transactions on Evolutionary Computation**, v. 27, n. 2, p. 201–221, 2023.
- Liu, S.; Lin, Q.; Li, J.; Tan, K. C. A Survey on Learnable Evolutionary Algorithms for Scalable Multiobjective Optimization. **IEEE Transactions on Evolutionary Computation**, v. 27, n. 6, p. 1941–1961, 2023.
- Liu, Z.-Z.; Wang, B.-C.; Tang, K. Handling Constrained Multiobjective Optimization Problems via Bidirectional Coevolution. **IEEE Transactions on Cybernetics**, v. 52, n. 10, p. 10163–10176, 2022.
- Liu, Z.-Z.; Wang, Y. Handling Constrained Multiobjective Optimization Problems With Constraints in Both the Decision and Objective Spaces. **IEEE Transactions on Evolutionary Computation**, v. 23, n. 5, p. 870–884, 2019.
- Ljung, L. System identification: theory for the user. **PTR Prentice Hall, Upper Saddle River, NJ**, v. 28, p. 540, 1999.
- Ma, X.; Li, X.; Zhang, Q.; Tang, K.; Liang, Z.; Xie, W.; Zhu, Z. A Survey on Cooperative Co-Evolutionary Algorithms. **IEEE Transactions on Evolutionary Computation**, v. 23, n. 3, p. 421–441, 2019.

- Ma, Z.; Wang, Y. Evolutionary Constrained Multiobjective Optimization: Test Suite Construction and Performance Comparisons. **IEEE Transactions on Evolutionary Computation**, v. 23, n. 6, p. 972–986, 2019.
- Mala-Jetmarova, H.; Sultanova, N.; Savic, D. Lost in optimisation of water distribution systems? A literature review of system operation. **Environmental Modelling & Software**, v. 93, p. 209–254, 2017.
- Mendes, C. S. R.; Araújo, A. F. R.; Farias, L. R. C. Non-dominated sorting bidirectional differential coevolution. In: **2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. 2023. p. 1709–1714.
- Ming, F.; Gong, W.; Wang, L.; Gao, L. Constrained Multiobjective Optimization via Multitasking and Knowledge Transfer. **IEEE Transactions on Evolutionary Computation**, v. 28, n. 1, p. 77–89, 2024.
- Peng, C.; Liu, H.-L.; Goodman, E. D. A Cooperative Evolutionary Framework Based on an Improved Version of Directed Weight Vectors for Constrained Multiobjective Optimization With Deceptive Constraints. **IEEE Transactions on Cybernetics**, v. 51, n. 11, p. 5546–5558, 2021.
- Qiao, K.; Chen, Z.; Qu, B.; Yu, K.; Yue, C.; Chen, K.; Liang, J. A dual-population evolutionary algorithm based on dynamic constraint processing and resources allocation for constrained multi-objective optimization problems. **Expert Systems with Applications**, v. 238, p. 121707, 2024.
- Shi, C.; Wang, Z.; Jin, X.; Xu, Z.; Wang, Z.; Shen, P. A novel three-stage multi-population evolutionary algorithm for constrained multi-objective optimization problems. **Complex & Intelligent Systems**, 2023.
- Song, S.; Zhang, K.; Zhang, L.; Wu, N. A dual-population algorithm based on self-adaptive epsilon method for constrained multi-objective optimization. **Information Sciences**, v. 655, p. 119906, 2024.
- Talbi, E.-G. Machine Learning into Metaheuristics: A Survey and Taxonomy. **ACM Computing Surveys**, v. 54, n. 6, p. 129:1–129:32, 2021.
- Tan, K. C.; Feng, L.; Jiang, M. Evolutionary Transfer Optimization - A New Frontier in Evolutionary Computation Research. **IEEE Computational Intelligence Magazine**, v. 16, n. 1, p. 22–33, 2021.
- Tian, Y.; Cheng, R.; Zhang, X.; Jin, Y. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]. **IEEE Computational Intelligence Magazine**, v. 12, n. 4, p. 73–87, 2017.
- Tian, Y.; Zhang, T.; Xiao, J.; Zhang, X.; Jin, Y. A Coevolutionary Framework for Constrained Multiobjective Optimization Problems. **IEEE Transactions on Evolutionary Computation**, v. 25, n. 1, p. 102–116, 2021.
- Tian, Y.; Zhang, Y.; Su, Y.; Zhang, X.; Tan, K. C.; Jin, Y. Balancing Objective Optimization and Constraint Satisfaction in Constrained Evolutionary Multiobjective Optimization. **IEEE Transactions on Cybernetics**, v. 52, n. 9, p. 9559–9572, 2022.

Tian, Y.; Zhu, W.; Zhang, X.; Jin, Y. A practical tutorial on solving optimization problems via PlatEMO. **Neurocomputing**, v. 518, p. 190–205, 2023. ISSN 0925-2312.

Wang, B.-C.; Li, H.-X.; Li, J.-P.; Wang, Y. Composite Differential Evolution for Constrained Evolutionary Optimization. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. 49, n. 7, p. 1482–1495, 2019.

Wang, C.; Liu, Z.; Qiu, J.; Zhang, L. Adaptive constraint handling technique selection for constrained multi-objective optimization. **Swarm and Evolutionary Computation**, v. 86, p. 101488, 2024.

Wang, J.; Liang, G.; Zhang, J. Cooperative Differential Evolution Framework for Constrained Multiobjective Optimization. **IEEE Transactions on Cybernetics**, v. 49, n. 6, p. 2060–2072, 2019.

Wang, Y.; Cai, Z.; Zhang, Q. Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters. **IEEE Transactions on Evolutionary Computation**, v. 15, n. 1, p. 55–66, 2011.

Wang, Y.; Huang, K.; Gong, W.; Ming, F. Bi-directional search based on constraint relaxation for constrained multi-objective optimization problems with large infeasible regions. **Expert Systems with Applications**, v. 239, p. 122492, 2024.

Wilcoxon, F. Individual comparisons by ranking methods. **Biometrics Bulletin**, International Biometric Society, Wiley, v. 1, n. 6, p. 80–83, 1945.

Xia, M.; Chong, Q.; Dong, M. A constrained multi-objective evolutionary algorithm with two-stage resources allocation. **Swarm and Evolutionary Computation**, v. 79, p. 101313, 2023.

Yang, K.; Zheng, J.; Zou, J.; Yu, F.; Yang, S. A dual-population evolutionary algorithm based on adaptive constraint strength for constrained multi-objective optimization. **Swarm and Evolutionary Computation**, v. 77, p. 101247, 2023.

Yang, Y.; Huang, P.-Q.; Kong, X.; Zhao, J. A constrained multi-objective evolutionary algorithm assisted by an additional objective function. **Applied Soft Computing**, v. 132, p. 109904, 2023.

Yang, Y.; Liu, J.; Tan, S.; Wang, H. A multi-objective differential evolutionary algorithm for constrained multi-objective optimization problems with low feasible ratio. **Applied Soft Computing**, v. 80, p. 42–56, 2019.

Yang, Y.; Liu, J.; Tan, S.; Liu, Y. A multi-objective differential evolution algorithm based on domination and constraint-handling switching. **Information Sciences**, v. 579, p. 796–813, 2021.

Ye, Q.; Wang, W.; Li, G.; Dai, R. A self-organizing assisted multi-task algorithm for constrained multi-objective optimization problems. **Information Sciences**, v. 664, p. 120339, 2024.

Yu, K.; Liang, J.; Qu, B.; Yue, C. Purpose-directed two-phase multiobjective differential evolution for constrained multiobjective optimization. **Swarm and Evolutionary Computation**, v. 60, p. 100799, 2021.

- Yu, K.; Liang, J.; Qu, B.; Luo, Y.; Yue, C. Dynamic Selection Preference-Assisted Constrained Multiobjective Differential Evolution. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. 52, n. 5, p. 2954–2965, 2022.
- Yu, K.; Wang, L.; Liang, J.; Wang, H.; Qiao, K.; Liang, T. Constraint subsets-based evolutionary multitasking for constrained multiobjective optimization. **Swarm and Evolutionary Computation**, v. 86, p. 101531, 2024.
- Zeng, Y.; Cheng, Y.; Liu, J. A coevolutionary algorithm assisted by two archives for constrained multi-objective optimization problems. **Swarm and Evolutionary Computation**, v. 82, p. 101372, 2023.
- Zhan, Z.-H.; Shi, L.; Tan, K. C.; Zhang, J. A survey on evolutionary computation for complex continuous optimization. **Artificial Intelligence Review**, v. 55, n. 1, p. 59–110, 2022.
- Zhou, J.; Zhang, Y.; Suganthan, P. N. Dual population approximate constrained Pareto front for constrained multiobjective optimization. **Information Sciences**, v. 648, p. 119591, 2023.
- Zhu, Q.; Zhang, Q.; Lin, Q. A Constrained Multiobjective Evolutionary Algorithm With Detect-and-Escape Strategy. **IEEE Transactions on Evolutionary Computation**, v. 24, n. 5, p. 938–947, 2020.
- Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. **IEEE Transactions on Evolutionary Computation**, v. 3, n. 4, p. 257–271, 1999.