



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Ciência da Computação

Adaptação do Algoritmo Genético NSGA-DO à Problemas de Otimização Multiobjetivo Estáticos e Dinâmicos

Jussara Gomes Machado

Feira de Santana

2023



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Ciência da Computação

Jussara Gomes Machado

**Adaptação do Algoritmo Genético NSGA-DO à
Problemas de Otimização Multiobjetivo Estáticos e
Dinâmicos**

Dissertação apresentada à Universidade Estadual de Feira de Santana como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Matheus Giovanni Pires

Feira de Santana

2023

Ficha Catalográfica - Biblioteca Central Julieta Carteado - UEFS

M131

Machado, Jussara Gomes

Adaptação do algoritmo genético NSGA-DO à problemas de otimização multiobjetivo estáticos e dinâmicos / Jussara Gomes Machado. – 2023.

122 f.: il.

Orientador: Matheus Giovanni Pires.

Dissertação (mestrado) – Universidade Estadual de Feira de Santana, Programa de Pós-graduação em Ciência da Computação, Feira de Santana, 2023.

1. Algoritmos. 2. Otimização multiobjetivo. I. Título. II. Pires, Matheus Giovanni, orient. III. Universidade Estadual de Feira de Santana.

CDU 004.722



UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Autorizada pelo Decreto Federal nº 77.496 de 27/04/1976

Reconhecida pela Portaria Ministerial nº 874 de 19/12/1986

Recredenciada pelo Decreto nº 9.271 de 14/12/2004

Recredenciada pelo Decreto nº 17.228 de 25/11/2016

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LISTA DE PRESENÇA DA SESSÃO PÚBLICA DE DISSERTAÇÃO DE MESTRADO EM CIÊNCIA DA COMPUTAÇÃO Nº 18





Mestrando(a): Jussara Gomes Machado

Título da Dissertação: *Adaptação do Algoritmo Genético NSGA-DO à Problemas de Otimização Multiobjetivos Estáticos e Dinâmicos*

Data: 27 de fevereiro de 2023

Horário: 14:30

Local: <https://meet.google.com/ehu-dmqm-dqv>

Nome	Assinatura
Alexandre Cláudio Botazzo Delbem	
Jussara Gomes Machado	
Matheus Giovanni Pires	
Rodrigo Tripodi Calumby	

Abstract

Evolutionary Algorithms (EAs) are useful in solving Multi-Objective Optimization Problems (MOOPs) because they allow finding different solutions with different compensations for the objectives. One class of EAs are Genetic Algorithms (GAs), which use parallel search and optimization techniques based on natural selection and genetic reproduction. A GA commonly applied in the resolution of MOOPs, both artificial and in the real world, is the NSGA-II, which is sometimes used as a basis for the development of other algorithms, such as the NSGA-DO.

The field of Multi-objective Optimization (MOO) is consolidated, we currently have different benchmarks, performance metrics and efficient AEs. However, regarding the latter, what is observed is that the performance of the algorithms is proportional to their complexity, which induces researchers from other fields to continue to prefer the NSGA-II. Furthermore, interest in Multi-objective Dynamic Optimization (DMOO), in which the environment changes over time, has intensified only in recent years and there are many challenges in this emerging field of research.

Regarding the NSGA-DO, it proposes modifications in part of the NSGA-II, and even having shown superior performance in other fields, the algorithm does not present satisfactory results when applied to continuous MOOPs. In this context, recognizing the simplicity and potential of the recent algorithm, as well as the need for advances in the field of DMOO, the objective of this research was the development of improvements to NSGA-DO, as well as the elucidation of important issues related to the field of DMOO.

The methodology adopted here was divided into two phases partially interspersed. In the first phase, classified as a descriptive bibliographical research, review studies published in the field of DMOO were identified, described and analyzed. In the second phase, classified as an explanatory experimental research, the evolutionary strategy of the NSGA-DO was investigated and improvements were applied.

As a result of the analysis of the studies, it can be seen that the main challenges in the field of DMOO revolve around detecting changes and responding to changes. In this process, a DMOA (Dynamic Multi-objective Algorithm) faces difficulties related to the preservation of diversity, convergence considering the new environment and recovery of possible unfeasible solutions. On experimentation, the modifications applied to NSGA-DO resulted in a new GA, Modified NSGA-DO (MNSGA-DO), which

surpasses NSGA-DO and even NSGA-II in problems with different characteristics . Also, a dynamic variant of MNSGA-DO was proposed, the Dynamic MNSGA-DO (D-MNSGA-DO), which achieved satisfactory performance, managing to track and respond to changes in the environment.

With the results obtained, it can be concluded that the present study achieved its objectives by proposing a new GA with a simple strategy and able to solve MOOPS and DMOPs, as well as presenting a compilation of review studies published over the years, these in the field from DMOO.

Keywords: Multi-objective Optimization, Dynamic Multi-Objective Optimization, Genetic Algorithms.

Resumo

Algoritmos Evolutivos (AEs) são úteis na resolução de Problemas de Otimização Multiobjetivo (MOOPs) por possibilitar encontrar distintas soluções com diferentes compensações para os objetivos. Uma classe de AEs são os Algoritmos Genéticos (AGs), que utilizam técnicas de busca e otimização paralela baseadas na seleção natural e reprodução genética. Um AG comumente aplicado na resolução de MOOPs, artificiais e do mundo real, é o NSGA-II, que, por vezes, é utilizado como base no desenvolvimento de outros algoritmos, como o NSGA-DO.

O campo da Otimização Multiobjetivo (MOO) se apresenta consolidado, atualmente temos diferentes *benchmarks*, métricas de desempenho e AEs eficientes. Porém, sobre esse último, o que se observa é que o desempenho dos algoritmos é proporcional a sua complexidade, o que induz pesquisadores de outros campos a continuar preferindo o NSGA-II. Ainda, o interesse pela Otimização Dinâmica Multiobjetivos (DMOO), em que o ambiente se modifica ao longo do tempo, se intensificou somente nos últimos anos e muitos são os desafios desse emergente campo de pesquisa.

Sobre o NSGA-DO, o mesmo propõe modificações em parte do NSGA-II, e mesmo tendo mostrado desempenho superior em outros campos, o algoritmo não apresenta resultados satisfatórios quando aplicado a MOOPs contínuos. Nesse contexto, reconhecendo a simplicidade e potencial do recente algoritmo, assim como a necessidade de avanços no campo da DMOO, o objetivo dessa pesquisa foi o desenvolvimento de melhorias ao NSGA-DO, assim como, a elucidação de questões importantes relacionadas ao campo da DMOO.

A metodologia aqui adotada foi dividida em duas fases parcialmente intercaladas. Na primeira fase, classificada como uma pesquisa bibliográfica descritiva, estudos de revisão publicados no campo da DMOO foram identificados, descritos e analisados. Na segunda fase, classificada como uma pesquisa experimental explicativa, a estratégia evolutiva do NSGA-DO foi investigada e melhorias foram aplicadas.

Como resultado da análise dos estudos pode-se perceber que os principais desafios do campo da DMOO giram em torno da detecção de mudanças e da resposta às mudanças. Nesse processo, um DMOA (Algoritmo Multiobjetivo Dinâmico) enfrenta dificuldades relacionadas à preservação da diversidade, convergência considerando o novo ambiente e recuperação de possíveis soluções inviáveis. Sobre a experimentação, as modificações aplicadas ao NSGA-DO resultaram em um novo AG, o *Modi-*

find NSGA-DO (MNSGA-DO), que supera o NSGA-DO e até mesmo o NSGA-II em problemas com diferentes características. Também, um variante dinâmico do MNSGA-DO foi proposto, o *Dynamic* MNSGA-DO (D-MNSGA-DO), o qual obteve um desempenho satisfatório, conseguindo rastrear e responder às mudanças de ambiente.

Com os resultados obtidos, pode-se concluir que o presente estudo alcançou seus objetivos ao propor um novo AG de estratégia simples e apto a resolver MOOPS e DMOPs, assim como apresentou um compilado dos estudos de revisão publicados ao longo dos anos, estes no campo da DMOO.

Palavras-chave: Otimização Multiobjetivo, Otimização Multiobjetivo Dinâmica, Algoritmos Genéticos.

Prefácio

Esta dissertação de mestrado foi submetida à Universidade Estadual de Feira de Santana (UEFS) como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

A dissertação foi desenvolvida no Programa de Pós-Graduação em Ciência da Computação (PGCC), tendo como orientador o Prof. Dr. **Matheus Giovanni Pires**.

Esta pesquisa foi financiada pela CAPES.

Agradecimentos

Agradeço a todos que me ajudaram, especialmente a minha família, amigos, professores que contribuíram com sugestões e orientador.

“Somos especiais mas somos todos iguais. Temos funções diferentes, hierarquias diferentes e até poderes diferentes, mas somos todos iguais; e quanto mais alto é o cargo que eu ocupo, maior é a minha necessidade de servir as pessoas.”

– Leandro Karnal

Sumário

Abstract	i
Resumo	iii
Prefácio	v
Agradecimentos	vi
Sumário	x
Lista de Publicações	xi
Lista de Tabelas	xii
Lista de Figuras	xiv
Lista de Abreviações	xv
1 Introdução	1
1.1 Objetivos	3
1.2 Contribuições	4
1.3 Organização do Trabalho	5
2 Fundamentação Teórica	6
2.1 Problemas de Otimização Multiobjetivo	7
2.1.1 Definições Básicas	8
2.1.2 Características dos MOOPs	10
2.1.3 Classificação dos DMOPs	13
2.2 <i>Benchmarks</i> e Indicadores de Qualidade	14
2.3 NSGA-II e NSGA-DO	17
2.3.1 NSGA-II (<i>Non-dominated Sorting Genetic Algorithm II</i>) . . .	18
2.3.2 NSGA-DO (<i>Non-dominated Sorting Genetic Algorithm Distance-Oriented</i>)	19

3	Revisão da Literatura	21
3.1	Trabalhos Relacionados à Otimização Multiobjetivo Estática	21
3.2	Trabalhos Relacionados à Otimização Multiobjetivo Dinâmica	22
3.3	Compilado de Alguns dos Tópicos mais Discutidos	41
3.4	Respondendo a Questão de Pesquisa Q1	43
4	Proposta de Adaptação do NSGA-DO	45
4.1	Análise de Desempenho do NSGA-DO para Problemas de Otimização Contínuos	46
4.2	Alterações Realizadas no NSGA-DO	47
4.3	Versão Dinâmica do NSGA-DO Modificado	50
5	Experimentos e Resultados	52
5.1	Experimentos com Problemas de Otimização Estáticos Contínuos	52
5.1.1	Experimentos com Problemas ZDT	53
5.1.2	Experimentos com Problemas WFG	57
5.1.3	Experimentos com Problemas LZ09	60
5.1.4	Experimentos com Problemas DTLZ	62
5.1.5	Comparação Estatística dos Resultados	64
5.2	Problema do Próximo Release (<i>Next Release Problem</i>)	65
5.2.1	Modelagem do NRP	66
5.2.2	Experimento	67
5.3	Avaliação do Desempenho do <i>Dynamic</i> MNSGA-DO (DynamicMNSGA-DO)	69
5.4	Considerações Sobre os Experimentos	74
5.4.1	Evolução do Indicador de Hipervolume	74
5.4.2	Criação de pontos ideais entre descontinuidades da PF	75
5.4.3	Desempenho Diante de Problemas com Muitos Objetivos	76
5.5	Respondendo as Questões de Pesquisa Q2 e Q3	77
6	Conclusões	80
6.1	Possíveis Trabalhos Futuros	80
	Referências	82
A	Classificação da Pesquisa e Metodologia	91
A.1	Classificação da Pesquisa	91
A.2	Metodologia	92
A.2.1	Coleta de Dados	92
A.2.2	Experimentos	92
B	Revisão Sistemática da Literatura	94
B.1	Procedimento Bola de Neve (<i>Snowballing</i>)	95
B.2	Busca em Bibliotecas Digitais	99
B.3	Busca Informal	101

Lista de Publicações

MACHADO, J. G. ; PIRES, M. G. ; BERTONI, F. C. ; PIMENTA, A. H. M. ; CAMARGO, H.A. . A Modified NSGA-DO for Solving Multiobjective Optimization Problems. In: 10th Brazilian Conference on Intelligent Systems (BRACIS), 2021, São Paulo. Conference proceedings BRACIS 2021, Part of the Lecture Notes in Computer Science book series (LNCS, volume 13073), [https : //doi.org/10.1007/978 – 3 – 030 – 91702 – 9](https://doi.org/10.1007/978-3-030-91702-9), 2021. v. 13073. p. 126-139.

Lista de Tabelas

2.1	Classificação dos Indicadores de Qualidade mencionados.	16
5.1	Parâmetros dos operadores de mutação e cruzamento.	53
5.2	Indicadores de qualidade para os problemas da família ZDT.	56
5.3	Média e desvio padrão do tempo de execução dos AGs para o ZDT.	56
5.4	Indicadores de qualidade para os problemas da família WFG.	59
5.5	Indicadores de qualidade para os problemas da família LZ09.	61
5.6	Indicadores de qualidade para os problemas da família DTLZ.	63
5.7	Comparação estatística dos resultados dos indicadores de qualidade.	64
5.8	Características das funções de teste FDA	69
5.9	Resultado Ganha-Perde pos 24 primeiros instantes de tempo	72
5.10	Hipervolume medido para os 24 primeiros instantes de tempo.	72
B.1	Crítérios de Inclusão.	97
B.2	Crítérios de Exclusão.	97
B.3	Crítérios da Avaliação de Qualidade	97
C.1	Custo dos requisitos (r_{11}, \dots, r_{20}) para a empresa	103
C.2	Importância dos requisitos (r_1, \dots, r_{10}) para os clientes	103
C.3	Importância dos requisitos (r_{11}, \dots, r_{20}) para os clientes	104
C.4	Importância dos clientes para o projeto	104

Lista de Figuras

2.1	Conjunto de possíveis soluções para o problema de aquisição de um eletrodoméstico. Adaptado de Deb (2001).	9
2.2	À esquerda temos a ilustração do espaço de decisão (PS^*), enquanto que à direita temos a ilustração do espaço objetivo (PF^*) (MATLAB, 2022).	9
2.3	Em (a), o objetivo multimodal enganoso foi exagerado para maior clareza. Em (b), temos um objetivo multimodal simples. Em (c), observe como os vetores objetivos são mais densos em direção a PF , caracterizando um problema enviesado. Em (d), os vetores objetivos rotulados com a mesma letra ocorrem em uma posição diferente na mesma fronteira, enquanto os vetores objetivos rotulados com o mesmo número ocorrem em fronteiras diferentes na mesma posição. Exemplos retirados de Huband et al. (2006)	11
2.4	PFs plotadas em gráficos bidimensionais, tridimensionais e em gráficos de coordenadas paralelas.	12
2.5	Ilustração dos Indicadores de Qualidade utilizados.	16
2.6	Algoritmo <i>Fast Non-Dominated Sort</i> . Adaptado de Deb et al. (2002).	17
2.7	Na ilustração à esquerda temos o algoritmo <i>Crowding Distance</i> do NSGA-II. Na ilustração à esquerda temos os pontos ideais do NSGA-DO.	18
2.8	À esquerda temos o algoritmo de preservação da diversidade do NSGA-II (adaptado de Deb et al. (2002)) e à direita temos o algoritmo de preservação da diversidade do NSGA-DO (adaptado de Pimenta and de Arruda Camargo (2015)).	20
4.1	PF^* do NSGA-DO ao longo das gerações (Problema ZDT1).	46
4.2	Alteração do NSGA-DO para eleição aleatória dos pontos ideais.	47
4.3	Discretização de um sinal contínuo com diferentes taxas de amostragem. A esquerda temos um sinal distorcido pela amostragem, com n amostras do sinal real, a direita temos um sinal melhor amostrado, com $2n$ amostras do sinal real.	49
4.4	Alternância na seleção das soluções pelos pontos ideais.	49
5.1	PFs^* encontradas pelos AGs para o <i>benchmark</i> ZDT.	54

5.2	Conjunto aproximação dos AGs NSGA-II, MNSGA-DO e NSGA-DO para o <i>benchmark</i> ZDT, com parte da PF^* deslocada e ampliada para facilitar observar a distribuição das soluções.	55
5.3	PFs^* encontradas pelos AGs para o <i>benchmark</i> WFG.	58
5.4	PFs^* encontradas pelos AGs para o <i>benchmark</i> DTLZ.	62
5.5	Dependências dos requisitos e preferências dos clientes. Adaptado de (Xuan et al., 2012).	65
5.6	PFs^* encontradas pelo NSGA-II, MNSGA-DO e NSGA-DO para uma instância de teste NRP de larga escala.	68
5.7	PF^* encontradas pelas versões dinâmicas dos AGs NSGA-II (a), MNSGA-DO (b) e NSGA-DO (c) para as instâncias de teste FDA{1-3}, consideramos uma única execução.	70
5.8	PS^* encontradas pelas versões dinâmicas dos AGs NSGA-II (a), MNSGA-DO (b) e NSGA-DO (c) para as instâncias de teste FDA{1-3}, consideramos uma única execução.	71
5.9	Valores de Hipervolume para os 24 primeiros instantes de tempo . . .	73
5.10	Evolução do Hipervolume para os AGs NSGA-II e MNSGA-DO. . . .	74
5.11	Pontos ideais criados entre as descontinuidades do problema ZDT3. .	75
5.12	Situação em que o algoritmo de preservação da diversidade dos AGs variantes do NSGA-II passa a selecionar todos os indivíduos. Adaptado de Deb et al. (2002).	77
A.1	Fluxograma da fase de coleta de dados.	92
A.2	Fluxograma da fase experimental.	93
B.1	Procedimento <i>Snowballing</i> . Adaptado de Wohlin (2014)	95
B.2	Exemplo de um conjunto inicial de estudos para o Procedimento Bola de Neve (<i>Snowballing</i>) construído a partir da <i>string</i> de busca “Dynamic Multi-objective Optimisation” através do Google Scholar.	96
B.3	Procedimento de busca por estudos em bibliotecas digitais	99
B.4	Formulário de extração dos dados.	102

Lista de Abreviações

Abreviação	Descrição
AE	Algoritmo Evolutivo
AEMO	Algoritmo Evolutivo Multiobjetivo
AG	Algoritmo Genético (<i>Genetic Algorithm</i>)
DMOA	Algoritmo Multiobjetivo Dinâmico
DMOP	Problemas de Otimização Multiobjetivo Dinâmicos (<i>Dynamic MOOP</i>)
FRBS	Sistemas Baseados em Regras Fuzzy
GD	Generation Distance
HV	Hipervolume
IGD	Inverse Generation Distance
IoT	Internet das Coisas (<i>Internet of Things</i>)
MNSGA-DO	Modified NSGA-DO
MOOP	Problemas de Otimização Multiobjetivo (<i>Multi-Objective Optimization Problem</i>)
NSGA	Nondominated Sorting Genetic Algorithm
NSGA-II	Nondominated Sorting Genetic Algorithm II
NSGA-DO	Non-dominated Sorting Genetic Algorithm Distance-Oriented
PS^*	Conjunto Pareto-Ótimo
PF^*	Frente Pareto-Ótima
PS_t^*	Conjunto Pareto-Ótimo Dinâmico
PF_t^*	Frente Pareto-Ótima Dinâmica
SLR	Revisão Sistemática da Literatura (<i>Systematic Literature Review</i>)
SBX	Simulated Binary Crossover
SCF	Sistemas de Classificação Fuzzy
VEGA	Vector Evaluated Genetic algorithm

Capítulo 1

Introdução

“In creating, the only hard thing is to begin: a grass blade’s no easier to make than an oak.”

– James Russell Lowell

A natureza baseada em população dos Algoritmos Evolutivos (AE) se presta bem à resolução de Problemas de Otimização Multiobjetivo (MOOPs) (do inglês, *Multi-Objective Optimization Problem*), caso em que se deseja descobrir uma variedade de soluções que oferecem uma variedade de compensações entre objetivos conflitantes (Huband et al., 2006). Segundo Deb (2001), um MOOP possui um conjunto de funções objetivo que devem ser minimizadas ou maximizadas no processo de otimização. Alguns MOOPs são especialmente projetados para testar diferentes capacidades dos algoritmos, capacidades estas essenciais na otimização de problemas do mundo real, como o *Knapsack Problem* (Martello, 1990) muito semelhante ao problema real NRP (do inglês *Next Release Problem*) do campo de Engenharia de Software (Bagnall et al., 2001).

Uma classe de AEs são os Algoritmos Genéticos (AGs), estes utilizam técnicas de busca e otimização paralela inspiradas na seleção natural e reprodução genética. O NSGA (do inglês, *Nondominated Sorting Genetic Algorithm*), um AG desenvolvido por Srinivas and Deb (1994), marcou a literatura dos Algoritmos Evolutivos Multiobjetivo (AEMOs) como sendo um dos primeiros AGs de ordenação não dominada. Sua característica principal é a capacidade de distribuir a população na região não dominada, mantendo seu potencial reprodutivo estável e uniforme, diferente do que acontecia com o VEGA, o primeiro AE para otimização multiobjetivo (Schaffer, 1985). No entanto, apesar deste marco, muitas foram as críticas associadas ao processo evolucionário do NSGA, as quais incluíam: complexidade $O(N^3)$, falta de elitismo e falta de um mecanismo explícito de preservação da diversidade (Deb et al., 2002).

Em uma tentativa assertiva de melhorar o desempenho do NSGA, Deb et al. (2002) desenvolveram o NSGA-II, um AG que, além de enfatizar soluções dominantes, utiliza o princípio elitista e implementa um algoritmo específico capaz de preservar a diversidade em sua população. O NSGA-II tem sido amplamente utilizado na resolução de diversos tipos de problema (Rangel et al., 2019; Parvizi and Rezvani, 2020; Xu et al., 2020; Zhang et al., 2020; Zhou and Zheng, 2020; Esfe and Tilebon, 2020; Kanwar and Kumar, 2020), e é comumente utilizado como algoritmo base no desenvolvimento de outros AGs (D'Souza et al., 2010; Kwan et al., 2007; Jozefowicz et al., 2005; Akbar and Irohara, 2020; Deb et al., 2007; Azzouz et al., 2015; de Queiroz Lafetá and de Oliveira, 2020).

Segundo Deb (2001), o resultado do processo de otimização de um MOOP é um conjunto de soluções aceitáveis que abrangem os diversos objetivos apresentados. Uma solução considerada aceitável pode ser entendida como uma solução que respeita todas as restrições impostas pelo problema a ser otimizado. Mitchell (1998) fornece um exemplo computacional de MOOP, este considerado um dos problemas mais importantes da Bioengenharia Computacional. Suponha que se deseja utilizar um algoritmo para encontrar determinada proteína (ou sequência de aminoácidos) capaz de combater um vírus específico. Nesse exemplo, a coleção de soluções candidatas (espaço de busca) é o conjunto de todas as sequências possíveis. Cada especificidade da proteína pode ser considerada como um objetivo (funções objetivos). A busca por um conjunto aceitável de sequências de aminoácidos (conjunto final de soluções) deve então considerar as diferentes propriedades da proteína (as restrições do problema). Para esse problema, visto a ampla dimensão do espaço de busca, os AGs surgem como uma estratégia inteligente capaz de escolher o próximo conjunto de sequências a avaliar.

Quando, em um MOOP, as funções objetivo e/ou as restrições mudam ao longo do tempo, estamos lidando com um Problema de Otimização Multiobjetivo Dinâmico (DMOP) (do inglês, *Dynamic Multi-objective Optimization Problem*). No mundo real, a maioria dos problemas de otimização são de natureza dinâmica e com pelo menos dois objetivos conflitantes. No entanto, a maioria das pesquisas se concentram na otimização dinâmica de único objetivo ou na otimização estática de muitos objetivos (Helbig et al., 2016).

Embora a Otimização Multiobjetivo Dinâmica (DMOO) não seja uma área de pesquisa recente (Xie and Steven, 1996; Mehnen et al., 2006; Zhou et al., 2007; Wang and Dang, 2008), o interesse por parte da comunidade científica se intensificou somente nos últimos anos e muitos são os desafios desse emergente campo de pesquisa. Possivelmente, essa migração de comportamento é caracterizada pelos avanços na área de otimização estática e pela natureza dinâmica dos atuais problemas do mundo real, cada vez mais complexos e multidisciplinares (Nnodim et al., 2021). Para otimização estática, temos um amplo conjunto de problemas de teste de referência, como por exemplo, os *benchmarks* ZDT (Zitzler et al., 2000), DTLZ (Deb et al., 2005), WFG (Huband et al., 2005), LZ09 (Li and Zhang, 2008), entre outros. Existem algoritmos eficazes, tais como, SMPSO (Nebro et al., 2009), MOEA/D (Zhang and Li, 2007),

NSGA-II (Deb et al., 2002) e o NSGA-III (Deb and Jain, 2013). Além disso, há indicadores de qualidade, que são medidas utilizadas na avaliação de desempenho destes algoritmos, como a *Distância Geracional* (GD) (Van Veldhuizen and Lamont, 1998), *Distância Geracional Invertida* (IGD) (Van Veldhuizen and Lamont, 1998), *Hipervolume* (HV) (Zitzler and Thiele, 1999) e outros.

Por outro lado, entre os desafios referentes à DMOO, podemos citar:

- Falta de uma abordagem padrão para comparar o desempenho dos Algoritmos Multiobjetivo Dinâmicos (DMOAs), sobretudo por conta da inexistência de *benchmarks* e medidas de desempenho consideradas de referência (Helbig et al., 2016; Azzouz et al., 2017; Helbig and Engelbrecht, 2013c);
- Necessidade de desenvolvimento de algoritmos que possam resolver DMOPs de forma eficiente (Helbig et al., 2012; Helbig and Engelbrecht, 2014);
- Necessidade de revisão do estado da arte sobre as funções de teste e medidas de desempenho (Azzouz et al., 2017).

Para Helbig et al. (2016), uma padronização do processo de avaliação e comparação dos DMOAs é essencial para possibilitar comparações cruzadas. Para Azzouz et al. (2017), é difícil comparar de forma justa os diferentes algoritmos propostos, a menos que se re-implemente todos e re-avale seu desempenho. Para Helbig et al. (2016), a maioria dos DMOAs propostos têm dificuldades de convergência em ambientes de rápidas mudanças e lutam para resolver DMOPs com características específicas (como Fronteiras de Pareto não lineares e dependência entre as variáveis de decisão). Os autores enfatizaram também que a maioria dos algoritmos propostos foram avaliados utilizando funções de teste normalmente fáceis de resolver.

1.1 Objetivos

Conforme já mencionado, existem na literatura propostas de novos algoritmos baseados no NSGA-II, sendo um deles o NSGA-DO (do inglês, *Non-dominated Sorting Genetic Algorithm Distance-Oriented*), desenvolvido por Pimenta and de Arruda Camargo (2015). Este algoritmo altera a forma de seleção dos indivíduos que devem seguir para a próxima geração, sob a justificativa de que o NSGA-II perde soluções ótimas em situações em que o conjunto de soluções não dominadas ultrapassa o limite populacional. No trabalho de Pimenta and de Arruda Camargo (2015), o NSGA-II e o NSGA-DO foram utilizados na geração de bases de conhecimento de Sistemas Baseados em Regras Fuzzy (FRBS), os quais foram utilizados na resolução de problemas de classificação, estes considerados problemas discretos. De acordo com os resultados, o sistema fuzzy resultante do NSGA-DO apresentou melhor acurácia e interpretabilidade.

Apesar do bom desempenho relatado nos trabalhos de Pimenta and de Arruda Camargo (2015) e Santana and Camargo (2019), o NSGA-DO foi testado em um pro-

blema discreto. No entanto, quando há a proposta de um novo AEMO, é interessante que este seja analisado por conjuntos de problemas de referência na literatura, também conhecidos como *benchmarks*, a fim de verificar todo o seu potencial, especialmente na resolução de MOOPs e DMOPs contínuos com diferentes características.

Diante do contexto apresentado, da simplicidade e do potencial do NSGA-DO, o objetivo desta pesquisa é o estudo de sua estratégia evolutiva na resolução de MOOPs e DMOPs contínuos. Além disso, realizar um levantamento do estado da arte no campo dos DMOPs, buscando entender melhor os desafios e como estes vêm sendo abordados pela comunidade científica ao longo dos anos. Os objetivos menores que levam ao cumprimento do objetivo principal incluem:

1. Analisar o desempenho do NSGA-DO na resolução de MOOPs contínuos;
2. Propor alterações ao NSGA-DO e analisar o seu desempenho na resolução de MOOPs e DMOPs contínuos;
3. Entender o cenário atual do campo de pesquisa da DMOO, mais especificamente, as dificuldades enfrentadas pelos algoritmos, assim como suas estratégias de enfrentamento.

Com a conclusão deste estudo esperamos responder às seguintes questões de pesquisa:

- Q1 - Quais dificuldades os atuais Algoritmos Evolutivos enfrentam na otimização dinâmica e quais estratégias são frequentemente adotadas para superá-las?
- Q2 - Qual o desempenho do NSGA-DO na resolução de MOOPs contínuos?
- Q3 - Como adaptar o NSGA-DO para resolução de DMOPs? Como é o seu desempenho?

1.2 Contribuições

As diferentes situações em que se observam problemas de otimização estáticos e dinâmicos, como por exemplo na otimização de rotas de acordo com o tráfego em tempo real (Wahle et al., 2001), planejamento de trajetórias *online* para veículos aéreos não tripulados (Peng and Xu, 2012), navegação robótica (Deb, 2011) e roteamento em redes *ad-hoc* móveis (Constantinou et al., 2011), bem como o crescente interesse em Internet das Coisas (IoT) (do inglês, *Internet of Things*) e Cidades Inteligentes, em que diferentes tecnologias interagem e cooperam em prol do alcance de objetivos em comum (Gomes et al., 2014), motivam ainda mais pesquisas na área, devido ao surgimento de novos problemas neste contexto. Azzouz et al. (2017), além de evidenciar o campo da IoT como um motivador, enfatizam outras contribuições, como a incorporação da preferência dos usuários em diferentes sistemas (preferências essas que podem mudar ao longo do tempo) e o aprimoramento de tecnologias inteligentes voltadas ao bem-estar e maior qualidade de vida, sobretudo a dos idosos.

Portanto, espera-se com os resultados deste trabalho contribuir com a proposta de um novo algoritmo genético multiobjetivo simples, baseado no NSGA-DO, para a resolução de problemas de otimização multiobjetivo estáticos e dinâmicos. Além disso, promover um entendimento sobre como o campo de pesquisa da DMOO vêm sendo conduzido ao longo dos anos.

1.3 Organização do Trabalho

O restante deste documento está dividido da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica, trazendo conceitos importantes para um bom entendimento desta pesquisa. O Capítulo 3 apresenta uma revisão da literatura sobre o campo DMOO, bem como, responde a questão de pesquisa Q1. O Capítulo 4 apresenta as modificações realizadas no NSGA-DO, assim como as justificativas. O Capítulo 5 apresenta e analisa os resultados obtidos da modificação, e por fim, o Capítulo 6 apresenta as conclusões e possíveis trabalhos futuros.

Capítulo 2

Fundamentação Teórica

“Knowing how things work is the basis for appreciation, and is thus a source of civilized delight.”

– William Safire

Pesquisas na área da Otimização Multiobjetivo por meio de Algoritmos Evolutivos foram tão bem sucedidas que, constantemente novas estratégias de evolução são propostas. Esse cenário também resultou em inúmeros estudos relacionados à comparação, isso com o objetivo de demonstrar superioridade de determinado algoritmo sob outros já propostos. Resumidamente, segundo Huband et al. (2006), esse processo de comparação pode ser definido nos seguintes passos: (1) Selecionar os AEs para comparação; (2) Criar ou selecionar um conjunto de problemas de teste; (3) Selecionar um conjunto de medidas para comparar os resultados; (4) Obter os resultados de cada algoritmo em cada problema de teste por meio de execuções ou coleta de dados em estudos já publicados; (5) Gerar medidas para os resultados; e, por fim, (6) Tirar as conclusões.

Atualmente, a comparação de AEs é facilitada pelo grande conjunto de problemas de teste multiobjetivo e pela ampla gama de medidas com as quais os resultados podem ser mensurados.

No entanto, para tirar conclusões precisas, é necessário que tanto os algoritmos quanto os problemas de teste empregados sejam bem compreendidos, assim como as métricas e os métodos estatísticos. Com isso, neste capítulo serão descritos os conceitos básicos relacionados à MOOPs e DMOOPs, os algoritmos envolvidos nos processos de comparação, bem como alguns dos problemas de teste e métricas de desempenho propostos na literatura.

2.1 Problemas de Otimização Multiobjetivo

De acordo com Li et al. (2015), um Problema de Otimização Multiobjetivo (MOOP) pode ser definido pela Equação 2.1.

$$\begin{aligned}
 & \text{Minimizar/Maximizar } F(x) = (f_1(x), \dots, f_m(x))^T \\
 & \text{sujeito a } g_j(x) \geq 0, j = 1, \dots, J \\
 & \quad h_k(x) = 0, k = 1, \dots, K \\
 & \quad x \in \Omega
 \end{aligned} \tag{2.1}$$

Em que g_j representa as j restrições de desigualdade e h_k representa as k restrições de igualdade. $\Omega = \Gamma_{i=1}^n[a_i, b_i] \subseteq \mathbb{R}^n$ é o domínio das variáveis, $x = (x_1, \dots, x_n)^T \in \Omega$ é uma solução candidata. $F : \Omega \rightarrow \mathbb{R}^m$ constitui as m funções objetivo conflitantes e \mathbb{R}^m é chamado espaço objetivo. O conjunto de objetivos factíveis é definido como $\Theta = \{F(x) \mid x \in \Omega, g_j(x) \geq 0, h_k(x) = 0\}$, para $j \in \{1, \dots, J\}$ e $k \in \{1, \dots, K\}$.

Os problemas que foram definidos pela Equação 2.1 não variam no tempo, no entanto, encontra-se no mundo real vários problemas de natureza dinâmica com pelo menos dois objetivos conflitantes, os quais são chamados Problemas de Otimização Multiobjetivo Dinâmicos (DMOP). Neste tipo de problema é introduzido o parâmetro tempo para representar as mudanças do ambiente (Equação 2.2).

$$\begin{aligned}
 & \text{Minimizar/Maximizar } F(x, t) = (f_1(x, t), \dots, f_m(x, t))^T \\
 & \text{sujeito a } g_j(x, t) \geq 0, j = 1, \dots, J \\
 & \quad h_k(x, t) = 0, k = 1, \dots, K \\
 & \quad x \in \Omega
 \end{aligned} \tag{2.2}$$

Novamente, g_j representa as j restrições de desigualdade e h_k representa as k restrições de igualdade. $\Omega = \Gamma_{i=1}^n[a_i, b_i] \subseteq \mathbb{R}^n$ é o domínio das variáveis, $x = (x_1, \dots, x_n)^T \in \Omega$ é uma solução candidata. $F : \Omega \rightarrow \mathbb{R}^m$ constitui as m funções objetivo conflitantes, \mathbb{R}^m é chamado espaço objetivo e t representa o tempo ou a natureza dinâmica do problema. O conjunto de objetivos factíveis é definido como $\Theta = \{F(x, t) \mid x \in \Omega, g_j(x, t) \geq 0, h_k(x, t) = 0\}$, para $j \in \{1, \dots, J\}$ e $k \in \{1, \dots, K\}$.

As funções objetivos (f_1, f_2, \dots, f_m) , tanto dos MOOPs quanto dos DMOPs, são consideradas conflitantes porque, geralmente, não é possível melhorar o resultado de uma sem, conseqüentemente, piorar o resultado da outra. Também, problemas de otimização podem ser escaláveis tanto em termos de parâmetros quanto em termos de objetivos.

2.1.1 Definições Básicas

Apresentamos nesta seção algumas definições importantes para um bom entendimento do assunto desta pesquisa.

Definição 1: *Dominância de Pareto:* O conceito de dominância de Pareto é comumente utilizado para indicar a qualidade de uma solução e auxiliar no processo de seleção. Segundo Deb (2001) e Miettinen (2012), uma solução x_1 domina uma solução x_2 (denotado por $x_1 \prec x_2$), se ambas as seguintes condições são verdadeiras: (1) A solução x_1 não é pior que x_2 em todos os objetivos; e (2) A solução x_1 é estritamente melhor que x_2 em pelo menos um objetivo. Quando tais condições são satisfeitas é dito que x_2 é dominada por x_1 , x_1 é não dominada por x_2 e/ou x_1 é não inferior que x_2 . Caso contrário, é dito que x_1 é incomparável a x_2 (denotado por $x_1 \sim x_2$). Ainda, se, em termos de dominância, x_1 é igual a x_2 , dizemos que as soluções são equivalentes.

Essa relação de dominância não é reflexiva, de modo que uma solução não pode dominar a si mesma; também não é simétrica, se $x_1 \prec x_2$, o contrário ($x_2 \prec x_1$) não pode ser verdadeiro; porém, é transitiva, se $x_1 \prec x_2$ e $x_2 \prec x_3$ então $x_1 \prec x_3$.

Definição 2: *Conjunto Pareto-Ótimo (PS^*):* As soluções não dominadas entre si formam um conjunto denominado Pareto-Ótimo, composto pelas melhores soluções do problema. Segundo Deb (2001), uma propriedade de quaisquer destas soluções é que um ganho em um objetivo implica uma perda em pelo menos um outro objetivo.

Definição 3: *Frente Pareto-Ótima (PF^*):* Quando vistos no espaço objetivo, o Conjunto Pareto-Ótimo forma a Fronteira de Pareto (ou Frente Pareto-Ótima). Estas são consideradas as melhores soluções para o problema.

Para uma melhor compreensão destas definições, considere a seguinte situação ilustrativa. Um cliente deseja comprar um eletrodoméstico qualquer; desconsiderando as exceções, pode-se dizer que quanto maior o custo de um produto melhor será seu desempenho. Dessa forma, se os critérios de busca pelo melhor produto forem menor preço e maior desempenho, não haverá uma solução ótima para todos os objetivos, pois o ganho em um objetivo é a consequente perda no outro, ou seja, o produto de maior custo terá o melhor desempenho e vice-versa.

Através da Figura 2.1 podemos observar o espaço objetivo, em que pode-se notar a relação de “Dominância de Pareto” entre as possíveis soluções. O conjunto de alternativas ótimas desse exemplo é formado pelas soluções 3, 5 e 6, que formam a Fronteira de Pareto.

Definição 4: *Conjunto Pareto-Ótimo Dinâmico (PS_t^*):* O conjunto de soluções não dominadas entre si, presentes no espaço de decisão, para o tempo t é denominado Conjunto Pareto-Ótimo Dinâmico.

Definição 5: *Frente Pareto-Ótima Dinâmica (PF_t^*):* A Frente Pareto-Ótima Dinâmica é o conjunto de soluções não dominadas no tempo t , em relação ao espaço objetivo.

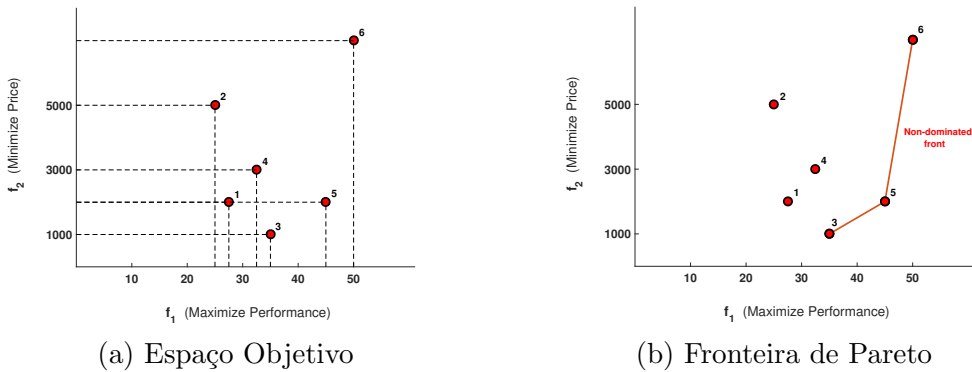


Figura 2.1: Conjunto de possíveis soluções para o problema de aquisição de um eletrodoméstico. Adaptado de Deb (2001).

Definição 6: *Espaço objetivo vs. Espaço de decisão:* Os valores atribuídos ao vetor de decisão x das funções objetivo, compostos pelas variáveis de decisão ($x = x_1, \dots, x_n$), formam o espaço de decisão, espaço das variáveis ou espaço dos parâmetros. Já os valores que as funções objetivo assumem, representam o espaço objetivo.

Para uma melhor compreensão, consideremos o exemplo apresentado na documentação da plataforma MATLAB (MATLAB, 2022): Seja duas funções objetivo (f_1 e f_2) convexas e de único parâmetro x . O gráfico à esquerda da Figura 2.2 apresenta seu espaço de decisão, enquanto que o gráfico à direita, apresenta seu espaço objetivo.

Observe que que entre os mínimos das funções, o valor de x varia entre 0 e 1, caso em que f_1 aumenta e f_2 diminui, assim, as duas funções são conflitantes. Convém então utilizar uma rotina de otimização para encontrar a região de compensação.

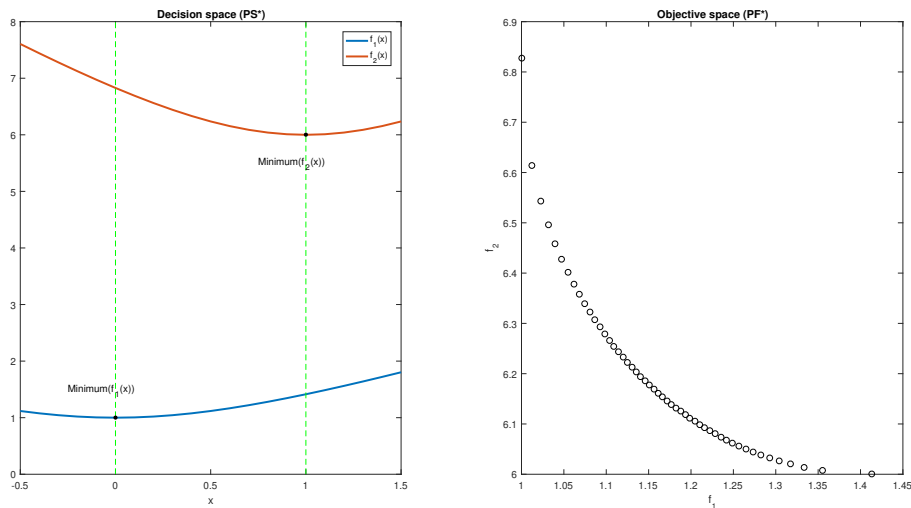


Figura 2.2: À esquerda temos a ilustração do espaço de decisão (PS^*), enquanto que à direita temos a ilustração do espaço objetivo (PF^*) (MATLAB, 2022).

2.1.2 Características dos MOOPs

Tendo compreendido os conceitos básicos, seguimos para o entendimento dos MOOPs, detalhando características do cenário de aptidão e geometria das PFs . Consideramos os trabalhos de Huband et al. (2006), Deb (1999) e Deb et al. (2005).

Cenário de Aptidão (*Fitness Landscape*)

A “natureza do cenário de aptidão” indica os tipos de dificuldades encontradas no espaço de busca, enquanto que a “relação entre o conjunto Pareto-Ótimo (PS) e a Frente Pareto-Ótima (PF)” nos permite julgar se as soluções encontradas são representativas. Em outras palavras, se PF^* encontrada representa o PS nos casos em que é impraticável identificar todo o conjunto.

Um problema de otimização, assim como o cenário de aptidão e o mapeamento entre o PS e a PF , pode ser considerado *um-para-um* ou *muitos-para-um*, sendo o segundo caso o que apresenta mais dificuldades ao otimizador, visto que as escolhas devem ser feitas entre vetores de parâmetros que mapeiam vetores objetivos idênticos.

Quando, em um mapeamento *muitos-para-um*, as pequenas perturbações dos parâmetros não alteram os valores objetivos devido à falta de informações de gradiente, temos os **Problemas com Regiões Planas**, os quais acrescentam dificuldades ao otimizador. Ainda, quando a maior parte do cenário de aptidão é bastante plana, e conseqüentemente, não fornece informações úteis sobre a localização das soluções ótimas, temos os **Problemas com Ótimos Isolados**, muito difíceis de resolver.

Sobre a multimodalidade, um **Problema Multimodal** possui, pelo menos, uma função objetivo com múltiplos ótimos locais (Figura 2.3 (b)), enquanto que em um **Problema Unimodal**, as funções objetivos possuem apenas um ótimo global.

Um **Problema Enganoso** representa um tipo especial de multimodalidade, o qual possui uma função objetivo enganosa. Tais problemas possuem pelo menos um ótimo verdadeiro e um ótimo enganoso, sendo que a maior parte do espaço de busca favorece o ótimo enganoso (Figura 2.3 (a)). Por si só, um problema multimodal já cria a possibilidade de um otimizador ficar preso em um ótimo local, problemas enganosos são mais complexos, pois o ótimo global se localiza em uma região improvável.

Uma característica particular do cenário de aptidão está relacionada à distribuição de vetores de parâmetros no espaço de decisão em relação à distribuição de vetores objetivos no espaço objetivo. Os casos em que ocorre uma variação significativa entre essas distribuições são chamados de *viés* (Figura 2.3 (c)). Em uma explicação prática, o *viés* ocorre quando pequenas mudanças nas variáveis de decisão de algumas soluções ótimas ocasionam mudanças significativas em seus vetores objetivos no espaço objetivo. Um *viés* pode estar relacionado à posição, caso em que as alterações nas variáveis causam mudanças significativas na distribuição das soluções ao longo da PF^* , ou relacionado à distância, caso em que as alterações nas variáveis causam deterioração significativa na proximidade com a PF^* . Quando o *viés* é deliberadamente incorporado a um problema, temos um **Problema Tendencioso**.

Outro aspecto importante se refere às dependências de parâmetros dos objetivos. Considerando o problema de otimizar determinado objetivo variando os parâmetros de sua função, se o seu conjunto de ótimos globais for o mesmo para todos os valores de parâmetros ótimos, dizemos que tal parâmetro é separável em tal objetivo. Caso contrário, o parâmetro é inseparável em tal objetivo. Assim, se todos os objetivos de um problema tiverem somente parâmetros separáveis, dizemos que o **Problema é Separável**, caso contrário, o **Problema é Inseparável**. Encontrar soluções ótimas em um problema separável tende a ser mais fácil, visto que, objetivos separáveis podem ser otimizados considerando cada parâmetro por vez.

Os parâmetros individuais também podem ser categorizados em termos de sua relação com o espaço objetivo, sendo estes **parâmetros de distância**, **parâmetros de posição** ou **parâmetros mistos**. Um parâmetro é categorizado como sendo de distância se a sua modificação alterar os aspectos de convergência, ou seja, podendo tornar o objetivo dominante, equivalente ou dominado. Por sua vez, um parâmetro é categorizado como sendo de posição, se a sua modificação alterar os aspectos de dispersão, ou seja, podendo tornar o objetivo incomparável ou equivalente. Por último, a modificação dos parâmetros mistos permite alterar tanto a posição quanto a distância de um objetivo. Observe a Figura 2.3 (d).

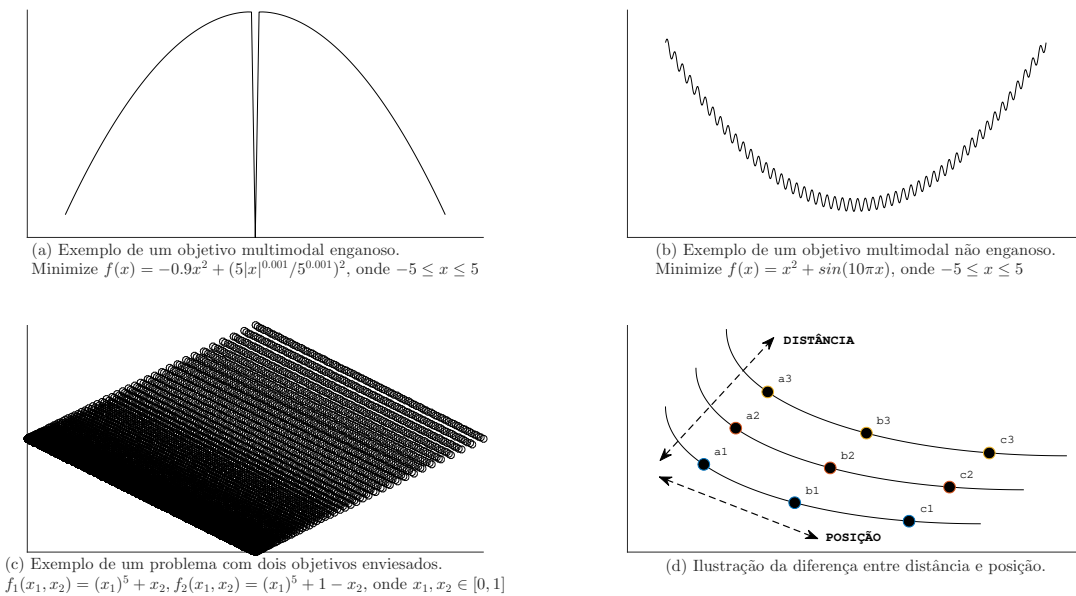


Figura 2.3: Em (a), o objetivo multimodal enganoso foi exagerado para maior clareza. Em (b), temos um objetivo multimodal simples. Em (c), observe como os vetores objetivos são mais densos em direção a PF , caracterizando um problema enviesado. Em (d), os vetores objetivos rotulados com a mesma letra ocorrem em uma posição diferente na mesma fronteira, enquanto os vetores objetivos rotulados com o mesmo número ocorrem em fronteiras diferentes na mesma posição. Exemplos retirados de Huband et al. (2006)

Geometria das Frentes Ótimas de Pareto

As Frentes Ótimas de Pareto (PF) podem ter uma grande variedade de geometrias, isso por retornarem um conjunto de soluções, em que cada uma compensa os objetivos de forma diferente. Entre as possibilidades, estas podem ser convexas, côncavas, lineares ou mistas. Ainda, a PF^* , pode ser degenerada, descontínua, invertida e/ou variante no tempo.

As Figuras de 2.4 ilustram diferentes geometrias e características das PFs , estas em gráficos bidimensionais, tridimensionais e em gráficos de coordenadas paralelas.

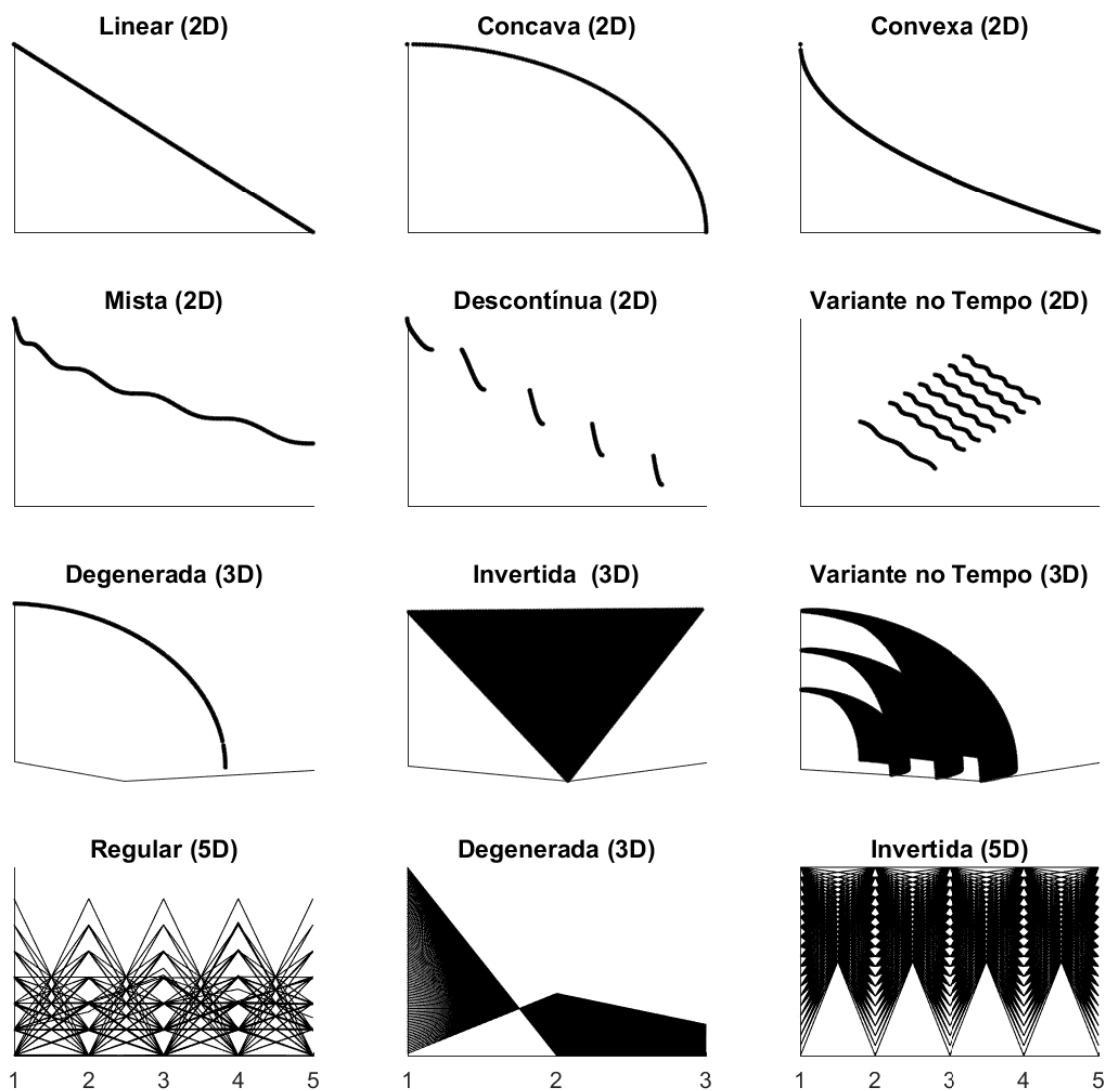


Figura 2.4: PFs plotadas em gráficos bidimensionais, tridimensionais e em gráficos de coordenadas paralelas.

2.1.3 Classificação dos DMOPs

A literatura descreve algumas classificações para DMOPs, vejamos algumas destas.

Classificação Baseada em Efeito (Relação entre o PF e o PS)

Farina et al. (2004) identificaram quatro tipos diferentes de DMOPs baseando-se no comportamento do Conjunto Pareto-Ótimo Dinâmico (*PS*) e da Frente Pareto-Ótima Dinâmica (*PF*):

- Tipo I: O *PS* (variáveis de decisão ótimas) muda, enquanto o *PF* (valores objetivos ideais) permanece invariante;
- Tipo II: Tanto o *PS* quanto o *PF* mudam;
- Tipo III: O *PF* muda, enquanto o *PS* permanece invariante;
- Tipo IV: O *PS* e o *PF* permanecem invariantes.

Para problemas do tipo IV, embora o *PS* e *PF* permaneçam inalterados ao longo do tempo, os autores enfatizam a possibilidade de o espaço de busca variar ao longo do tempo, como, por exemplo, variações dos ótimos locais.

Classificação Baseada na Origem

Tantar et al. (2011) propuseram uma forma de classificação dos DMOPs considerando a origem ou a fonte da dinâmica. Os quatro cenários propostos são descritos a seguir, podendo estes serem sobrepostos.

- Caso 1: As variáveis de decisão mudam ao longo do tempo;
- Caso 2: As funções objetivo mudam ao longo do tempo;
- Caso 3: As variáveis de decisão e/ou funções objetivo dependem de seus valores em ambientes anteriores ;
- Caso 4: As restrições variam ao longo do tempo.

Outras classificações foram propostas na literatura, estas baseadas na frequência, gravidade e previsibilidade das mudanças (Richter, 2013) (Azzouz et al., 2017).

Como o próprio nome supõe, a classificação baseada em **frequência** se refere ao intervalo entre as mudanças, sendo que quanto maior a frequência menos tempo o otimizador terá para se adaptar ao novo ambiente; a classificação baseada na **gravidade** se refere ao grau das mudanças (grandes ou pequenas alterações, por exemplo), quanto menor o grau das mudanças mais próximas são as instâncias do DMOP, podendo o otimizador utilizar informações de ambientes anteriores para acelerar a velocidade de convergência; por fim, a classificação baseada em **previsibilidade**, indica se as mudanças são aleatórias e independente da anterior (mudanças imprevisíveis, acíclicas) ou se as mudanças são previsíveis, caso em que são determinísticas (mudanças periódicas, cíclicas).

2.2 *Benchmarks* e Indicadores de Qualidade

Essa seção descreve alguns importantes conjuntos de problemas, comumente aplicados em estudos comparativos e de avaliação de desempenho de AEs, assim como as métricas utilizadas.

Problemas de Referência (*Benchmarks Problems*)

Zitzler et al. (2000) propuseram um conjunto de seis funções de teste (ZDT1-ZDT6), com cada função possuindo uma característica particular, com diferentes níveis de dificuldade no processo de otimização evolutiva, principalmente na convergência para a Frente Pareto-Ótima. As características desse conjunto de problemas incluem fronteira convexa, não convexa, convexa descontínua, convexa multimodal e convexa enganosa. Os autores argumentam que ao investigar essas diferentes características separadamente, é possível prever os tipos de problemas para os quais uma determinada técnica é ou não adequada.

Deb et al. (2005) propuseram um conjunto com nove funções de teste (DTLZ1-DTLZ9) para avaliar a eficácia dos AEs multiobjetivo no tratamento de problemas com mais de dois objetivos. Entre as principais características do conjunto de teste sugerido estão a simplicidade de construção, escalabilidade para qualquer número de variáveis de decisão e objetivos, conhecimento da forma e da localização da Frente Pareto-Ótima, e introdução de dificuldades controladas associadas à convergência e distribuição das soluções ao longo da fronteira.

Huband et al. (2006) apresentaram um conjunto de ferramentas (WFG *toolkit*) para a criação de problemas de teste multiobjetivo escalonáveis. O WFG *Toolkit* é flexível e permite que características como *viés*, multimodalidade e não separabilidade sejam incorporadas e combinadas em um mesmo problema. Várias geometrias para a Frente Pareto-Ótima também são suportadas, incluindo geometrias convexas, côncavas, mistas convexas/côncavas, lineares, degeneradas e descontínuas. Os autores apresentaram também nove problemas (WFG1 - WFG9) de referência, construídos a partir desse *toolkit*.

Li and Zhang (2008) propuseram um conjunto de nove funções de teste (LZ09 F1 - LZ09 F9), todos bi-objetivos, com exceção do LZ09 F6, que possui três objetivos. O PS^* de cada uma dessas funções é arbitrário, tendo sido projetadas para testar a capacidade dos AEs em lidar com formas PS^* complicadas. As características dessa família de problemas faz com que muitos algoritmos tenham dificuldades em encontrar soluções próximas às soluções ótimas da PF^* .

Muitos outros problemas foram propostos na literatura, tais como, os *benchmarks* CEC2009 (Zhang et al., 2008), GLT (Gu et al., 2012), CDTLZ (Jain and Deb, 2013) e as funções MOP (Liu et al., 2013).

Indicadores de Qualidade

Os indicadores de qualidade, ou medidas de desempenho, permitem quantificar o desempenho dos algoritmos em relação à determinados requisitos, como por exemplo, precisão, que mede a proximidade das soluções encontradas com a verdadeira PF , e diversidade (ou dispersão) das soluções, que indica o quão bem as soluções encontradas é representativa.

Zitzler et al. (2003) define uma medida (métrica ou indicador) da seguinte forma:

- Uma medida de desempenho m -ária P , é uma função $P : \Omega^m \rightarrow \mathbb{R}$, que atribui a cada uma das m soluções aproximadas em PF^* um valor real $P(PF_1^*, PF_2^*, \dots, PF_m^*)$.

Van Veldhuizen and Lamont (1998) propuseram duas métricas de desempenho para medir o grau de proximidade entre o conjunto de soluções ideais e o conjunto de soluções encontradas, denominadas *Distância Geracional (GD)* (do inglês, *Generational Distance*) e *Distância Geracional Invertida (IGD)* (do inglês, *Inverted Generational Distance*). A métrica GD indica a convergência dos algoritmos ao calcular, para cada ponto da PF^* encontrada, a menor distância euclidiana à qualquer ponto da PF ideal. O IGD é considerado tanto uma métrica de convergência quanto uma métrica de diversidade, pois este calcula para cada ponto da PF ideal, a menor distância euclidiana à qualquer ponto da PF^* encontrada.

Zitzler and Thiele (1999) propuseram a conhecida métrica Hipervolume (HV) para calcular a convergência e diversidade no conjunto de soluções encontradas por um algoritmo. Essa métrica mede o tamanho do espaço objetivo dominado pelas soluções em PF^* e limitados por um ponto de referência. O custo computacional do HV é proporcional a quantidade de objetivos.

Deb et al. (2002) propuseram uma métrica para medir a propagação (ou *Spread*) das soluções encontradas por um algoritmo, ou seja, indica a diversidade das soluções na PF^* encontrada. Resumidamente, essa métrica mede a extensão da propagação alcançada entre as soluções obtidas em relação a toda a região Pareto-Ótima.

Zitzler et al. (2003) propuseram um novo indicador, denominado ϵ - *indicator* ou *Epsilon*, para medir a convergência do conjunto de soluções encontrado por um algoritmo. O *Epsilon* fornece o fator pelo qual o conjunto PF encontrado é pior do que o conjunto PF ideal em relação a todos os objetivos.

Ishibuchi et al. (2015) propuseram modificações para as métricas GD e IGD . As novas métricas propostas, denominadas $GD+$ e $IGD+$, levam em consideração a relação de dominância entre uma solução e um ponto de referência quando calcula a sua distância. Se uma solução é dominada por um ponto de referência, a distância euclidiana é usada para o cálculo da distância sem modificação. No entanto, se eles não são dominados entre si, é calculada a distância mínima do ponto de referência à região dominada pela solução.

Muitos outros indicadores foram propostos na literatura, tais como, Máximo *Spread* (*MS*) (Zitzler, 1999), Razão de Hipervolume (*HVR*) (Van Veldhuizen, 1999), Taxa de Erro (*E*) e Número de soluções (*NS*) (Van Veldhuizen, 1999), Métrica de Espalhamento de Schott (*SS*) (Scott, 1995), C-metric (*C*) (Zitzler, 1999), U-measure (*U*) (Leung and Wang, 2003). A “Tabela 2.1” apresenta uma breve classificação destes.

Tabela 2.1: Classificação dos Indicadores de Qualidade mencionados.

Tipo	Descrição	Indicadores
Precisão	Mede a proximidade com a verdadeira PF	GD, IGD, E, GD+, IGD+
Diversidade	Mede a extensão da PF^* ou o quão bem as soluções estão espalhadas ao longo da PF^*	<i>Spread</i> , <i>U-measure</i> , <i>C-metric</i> , <i>MS</i> , <i>NS</i> , <i>SS</i>
Combinadas	Mede tanto a precisão quanto a diversidade.	HV, HVR, <i>e-indicator</i>

Nesta pesquisa foram utilizados quatro diferentes indicadores de qualidade, o GD e IGD, consideradas métricas de precisão (embora alguns autores considerem o IGD também uma métrica de diversidade), o *Spread*, considerado uma métrica de diversidade e o HV, considerado uma métrica combinada, capaz de avaliar tanto a convergência quanto a diversidade.

A Figura 2.5 ilustra o funcionamento destes quatro indicadores.

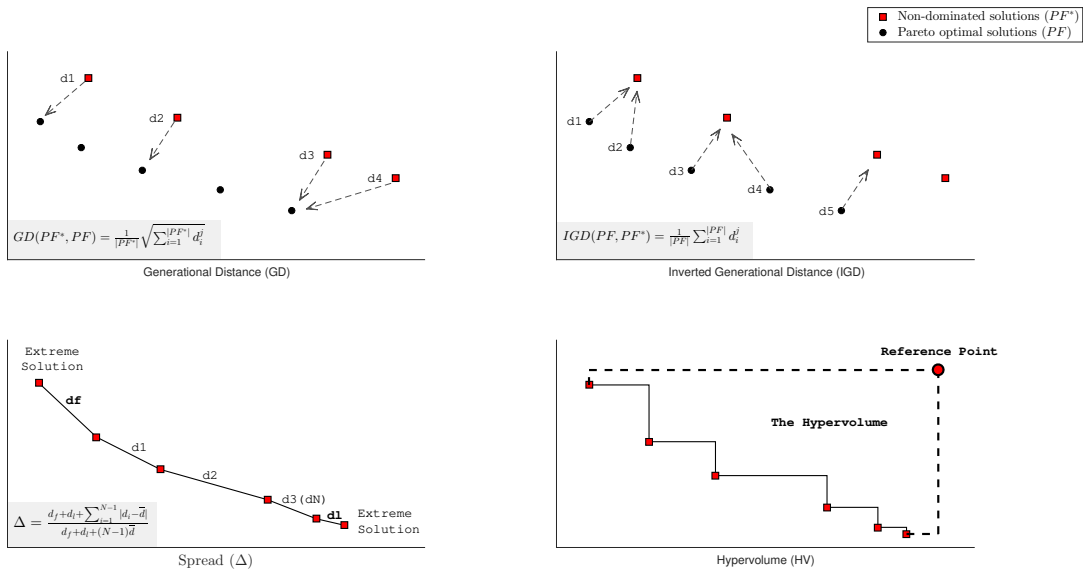


Figura 2.5: Ilustração dos Indicadores de Qualidade utilizados.

2.3 NSGA-II e NSGA-DO

A característica principal do NSGA-II e do NSGA-DO é a utilização do princípio elitista e a aplicação de um mecanismo explícito de preservação da diversidade. A herança deixada pelo NSGA é a priorização de soluções não dominadas através do algoritmo *Fast Non-Dominated Sort* (Deb et al., 2002). Por outro lado, o algoritmo de preservação da diversidade é diferente para os dois algoritmos.

Para ambos os algoritmos, a estratégia inicial de evolução tem o seguinte princípio de funcionamento. Na primeira geração do algoritmo, uma população aleatória é criada (P_t), também conhecida como população da geração atual. Através do processo de seleção, cruzamento e mutação é gerada uma população de descendentes (Q_t). A junção da população atual com a população de descendentes forma uma população com o dobro do tamanho permitido, denominado (R_t). As soluções desse novo conjunto são posteriormente classificadas e separadas em fronteiras de acordo com sua dominação através do algoritmo *Fast Non-Dominated Sort*.

Como a metade da população R_t não pode ser alocada para a nova população (P_{t+1}), a seleção prioriza as soluções das fronteiras dominantes (conforme a definição de “Dominância de Pareto”), F_1 , depois F_2 , até F_n . Quando uma das fronteiras possui mais soluções que o limite restante na nova população, a seleção se baseia em sua ordenação interna, por meio de um algoritmo de preservação da diversidade. A Figura 2.6 demonstra graficamente todo esse processo.

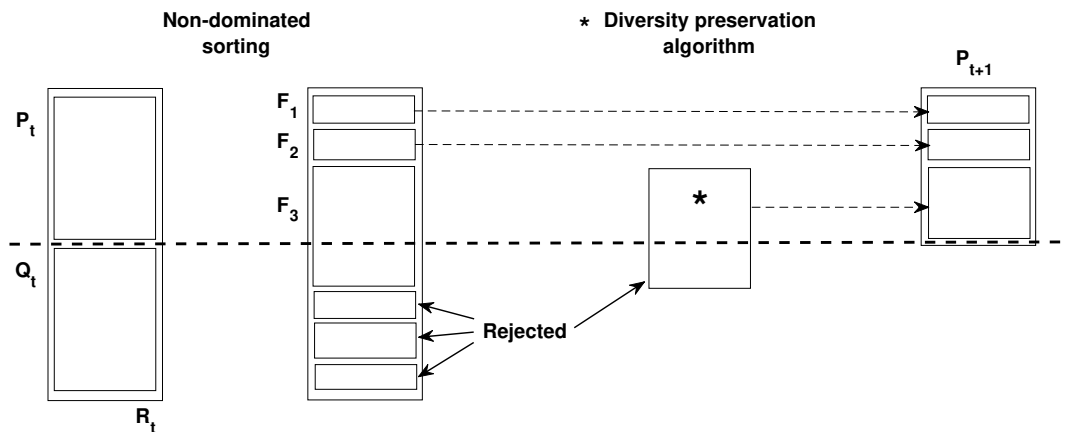


Figura 2.6: Algoritmo *Fast Non-Dominated Sort*. Adaptado de Deb et al. (2002).

A ordenação interna da última fronteira alocada é diferente para cada um dos dois algoritmos, porém, com um mesmo objetivo principal, preservar a diversidade na nova população.

A seguir será detalhada a diferença entre o NSGA-II e o NSGA-DO, que é exatamente o algoritmo de preservação da diversidade.

2.3.1 NSGA-II (*Non-dominated Sorting Genetic Algorithm II*)

O algoritmo de preservação da diversidade utilizado pelo NSGA-II é o *Crowding Distance* (Deb et al., 2002). Basicamente, a estratégia desse algoritmo é priorizar as soluções mais distantes de seus vizinhos, assumindo que estas estão em regiões menos populosas.

De forma mais detalhada, o algoritmo tem o seguinte princípio de funcionamento. Inicialmente, as soluções de uma mesma fronteira F_i são ordenadas em relação a cada objetivo. As soluções pertencentes às extremidades recebem um valor de distância infinito, para garantir que sejam selecionadas, enquanto as soluções intermediárias, por sua vez, recebem um valor proporcional à distância de suas duas soluções adjacentes.

Para cada solução intermediária i é calculada a distância média em relação às soluções vizinhas $i-1$ e $i+1$. As três soluções envolvidas formam um cubóide centralizado na posição da solução i , como mostra a Figura 2.7 (a). A seleção prioriza então as soluções com maiores valores de distância.

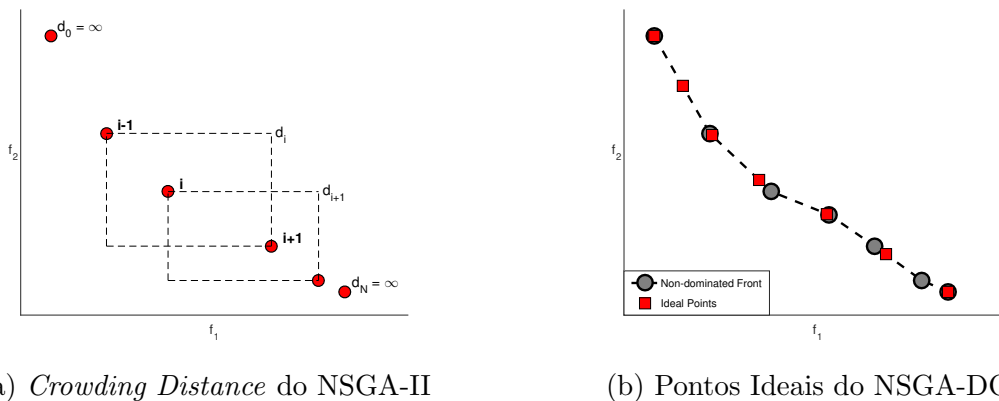


Figura 2.7: Na ilustração à esquerda temos o algoritmo *Crowding Distance* do NSGA-II. Na ilustração à esquerda temos os pontos ideais do NSGA-DO.

Implementações mais recentes do NSGA-II permitem montá-lo considerando diferentes componentes, com isso, é possível construir versões de melhor desempenho mas mantendo sua estratégia de otimização original. Um exemplo, seria a utilização de operadores de variação (operadores de mutação e cruzamento) baseados em decomposição ao invés dos convencionais, a incorporação de paralelismo, diferentes estratégias de criação da população inicial, assim como diferentes formas de encerrar a execução do AG, ou mesmo a substituição do *Fast Non-Dominated Sort* por outro mais eficiente, como o *Merge Non-dominated Sorting* (Moreno et al., 2020). Observe que essas muitas possibilidades podem ser incorporadas também em suas variantes.

2.3.2 NSGA-DO (*Non-dominated Sorting Genetic Algorithm Distance-Oriented*)

A proposta do NSGA-DO é orientar as soluções a convergirem para pontos ideais da fronteira, analisando quais das soluções candidatas são mais próximas a estes pontos e priorizando sua seleção (Pimenta and de Arruda Camargo, 2015). A fronteira considerada na criação destes pontos é o *front* menos dominado (F_1), criado pelo algoritmo *Fast Non-Dominated Sort*, como mostra a Figura 2.7 (b).

De forma simplificada, o NSGA-DO seleciona os indivíduos da seguinte forma. Seja S o conjunto de soluções candidatas, com nS soluções não dominadas oriundas da fronteira F_1 . O comprimento total (L) dessa fronteira é calculado somando-se o valor das nF funções lineares necessárias para conectar cada par de soluções vizinhas em S . Tendo conhecimento do comprimento total da fronteira, o espaçamento ideal (E) entre duas soluções adjacentes é definido pela divisão de L por nF . Considerando o espaçamento ideal encontrado, pontos ideais são criados ao longo de *front* F_1 e as soluções mais próximas a estes pontos são alocadas para a próxima geração.

Pimenta and de Arruda Camargo (2015) apresentam a definição matemática do algoritmo de criação dos pontos ideais. Suponha que cada solução S_i do conjunto S seja denotada por (x_i, y_i) , em que x_i e y_i representam os objetivos a serem otimizados no processo evolutivo. A função linear que intercepta duas soluções vizinhas desse conjunto (por exemplo, S_i e S_j) é definida pela função f^k , (Equação 2.3), em que a_k representa a inclinação da reta definida pela razão $(y_j - y_i) / (x_j - x_i)$ e b_k a interceptação no eixo y do sistema de coordenadas cartesianas.

$$f_{ij}^k(x) = a_k + b_k \quad (2.3)$$

Para calcular o comprimento l_k da função f^k entre os pontos (x_i, y_i) e (x_j, y_j) , é utilizada a equação $l_k = g_k(x_j) - g_k(x_i)$, sendo $j > i$. A função $g_k(x)$ é definida pela Equação 2.4:

$$g_k(x) = x \sqrt{1 + (a_k)^2} \quad (2.4)$$

Dessa forma, o comprimento L da fronteira, formada pelo conjunto S , é definida pela Equação 2.5, sendo $nF = nS - 1$.

$$L = \sum_{k=1}^{nF} l_k \quad (2.5)$$

Tendo conhecimento do comprimento total da fronteira, encontramos o espaçamento ideal entre duas soluções vizinhas a partir da Equação 2.6:

$$E = \frac{L}{nF} \quad (2.6)$$

O cálculo do valor de E , bem como a criação dos pontos ideais ocorrem em cada geração do NSGA-DO, visto que a forma e o comprimento do fronteira dominante se alteram ao longo das iterações.

A Figura 2.8 ilustra a diferença entre os algoritmos de preservação da diversidade do NSGA-II e NSGA-DO, para o caso em que se precisa selecionar oito indivíduos. No exemplo são apresentadas duas fronteiras de pareto (F_1 e F_2), em que as soluções em F_1 dominam as soluções em F_2 e não são dominadas por quaisquer outras soluções. Os círculos (F_1S_1, \dots, F_2S_4) representam as soluções do problema, os quadrados (I_1, \dots, I_5) do gráfico à direita representam os pontos ideais e $function1$ $function2$ são os objetivos conflitantes a serem otimizados.

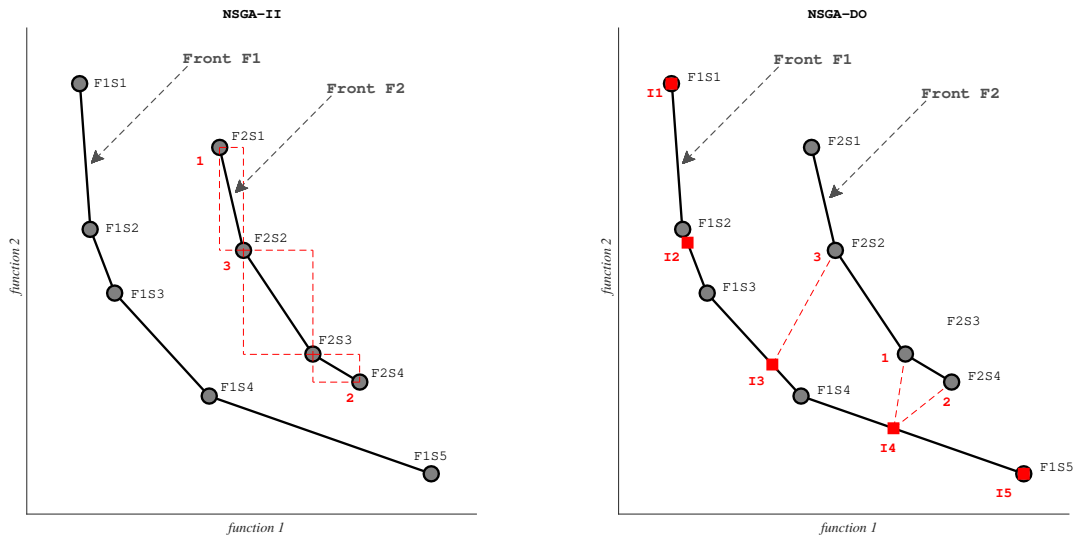


Figura 2.8: À esquerda temos o algoritmo de preservação da diversidade do NSGA-II (adaptado de Deb et al. (2002)) e à direita temos o algoritmo de preservação da diversidade do NSGA-DO (adaptado de Pimenta and de Arruda Camargo (2015)).

Inicialmente, tanto o NSGA-II quanto o NSGA-DO, selecionariam todas as soluções da fronteira F_1 , porque são dominantes e por haver espaço para a alocação de toda a fronteira ($F_{1size} < N$). Como a população é de oito indivíduos e a seleção já alocou na nova população os cinco indivíduos de F_1 , apenas três indivíduos em F_2 serão selecionados para compor a próxima geração. Diferente do NSGA-II, que escolheria as soluções mais distantes dos seus vizinhos (S_1, S_4 e S_2), o NSGA-DO seleciona os indivíduos mais próximos a um ponto ideal, no caso S_3, S_4 e S_2 , nessa ordem. Desse modo, o conjunto final de soluções para os dois algoritmos são:

$$S_{\text{NSGA-II}} = \{F_1S_1, F_1S_2, F_1S_3, F_1S_4, F_1S_5\} + \{F_2S_1, F_2S_4, F_2S_2\}$$

$$S_{\text{NSGA-DO}} = \{F_1S_1, F_1S_2, F_1S_3, F_1S_4, F_1S_5\} + \{F_2S_3, F_2S_4, F_2S_2\}$$

Capítulo 3

Revisão da Literatura

3.1 Trabalhos Relacionados à Otimização Multiobjetivo Estática

O campo da Otimização Multiobjetivo (MOO) estática se apresenta muito consolidado. Atualmente temos um amplo conjunto de EAs, com diferentes estratégias, características e complexidades, bem como um amplo conjunto MOOPs e métricas de desempenho. A apresentação dos novos algoritmos geralmente segue o padrão: aplicar um amplo conjunto de problemas de teste; medir o desempenho do algoritmo proposto em comparação com outros existentes, n vezes; calcular estatísticas básicas sobre o tempo e/ou qualidade da PF^* (média, mediana, moda e outros) e apresentar os resultados em forma de tabela.

Em parte, esse notável avanço, principalmente relacionado a padronização, se justifica pela presença de *frameworks* e repositórios que agrupam diferentes achados desse campo, ainda, utilizando diferentes linguagens, facilitando assim o desenvolvimento de novas pesquisas. Entre os *frameworks* e repositórios, podemos destacar: o jMetal (disponível em Java, Python e C++), que reúne AEs, MOOPs, indicadores de qualidade e testes estatísticos (mais recentemente, com suporte a implementações dinâmicas); o MOEA (disponível em Java), com a mesma proposta do jMetal; o Paradiseo (disponível em C++), um framework de otimização heurística autointitulado como a maior base de código aberto existente; o HeuristicLab que agrupa um amplo conjunto de algoritmos, com diferentes estratégias, bem como algoritmos de *Machine Learning* e problemas de teste; o Opt4J (disponível em Java), uma estrutura de código aberto para Computação Evolutiva.

Outros frameworks ativos de código aberto: EVA2, Watchmaker, ECJ, JCLEC, JGAP, Jenes, EO, Open BEAGLE, PISA, Jenetics, ECF, DEAP.

3.2 Trabalhos Relacionados à Otimização Multiobjetivo Dinâmica

É comum que ao propor novos algoritmos, métricas e problemas dinâmicos os autores façam um levantamento, geralmente simples, de estado da arte. Porém, consideramos aqui somente estudos que tiveram por objetivo principal descrever o cenário atual, à época, da Otimização Multiobjetivo Dinâmica. Contudo, no Anexo A é apresentado um guia sobre como realizar uma ampla e minuciosa Revisão Sistemática da Literatura neste campo de pesquisa.

benchmarks for Dynamic Multi-objective Optimisation (2013)

Helbig and Engelbrecht (2013b) destacam a inexistência de funções de *benchmark* consideradas padrão no campo da DMOO e como contribuição propõem características do que seria um conjunto ideal de funções de *benchmark*. Mais especificamente, o trabalho contribuiu destacando as limitações atuais dos DMOAs e dos DMOPs utilizados e sugerindo um conjunto ideal de funções de *benchmark*.

Sobre as características de um conjunto de funções de *benchmark* MOO ideal, Helbig and Engelbrecht (2013b) mencionam que as funções de teste devem ser fáceis de construir, devem ser escaláveis em termos do número de variáveis de decisão e número de funções objetivo, devem possuir uma *PF* fácil de entender, cuja a forma e localização sejam conhecidas, e deve dificultar a convergência dos algoritmos para a verdadeira *PF*, bem como dificultar uma boa distribuição de soluções. Para produzir dificuldades em relação à convergência e diversidade, o estudo menciona funções de teste multimodal, enganosa, com ótimo isolado ou contendo ruído. Ainda, o estudo menciona algumas propriedades que uma verdadeira *PF* deve ter para causar dificuldades, tais como, convexidade ou não convexidade, descontinuidade e espaçamento não uniforme de soluções.

Sobre as características de um conjunto de funções de *benchmark* para Otimização Dinâmica de Objetivo Único (DSOO) ideal, o estudo menciona que uma função de teste não deve permitir que o ambiente mude completamente de uma etapa de tempo para outra, sem qualquer conexão com o ambiente anterior, pois um simples algoritmo que implementa a estratégia de reinicialização após mudanças teria um bom desempenho. Helbig and Engelbrecht (2013b) destacam ainda as características de um bom gerador de *benchmark* dinâmico de objetivo único, sendo elas, a possibilidade de modificar a complexidade do ambiente e assim desenvolver funções representativas de problemas do mundo real (em termos de sua complexidade); possibilitar especificar características morfológicas (por exemplo, a localização do pico, forma e altura); a possibilidade de especificar o tipo de mudanças do ambiente (por exemplo, se as mudanças são recorrentes ou caóticas e se as mudanças são grandes ou pequenas); e garantir que o ambiente possa ser definido de maneira inequívoca. Ainda, concluindo o assunto, o estudo relata sobre a necessidade de as funções de *benchmark* DSOP terem uma complexidade computacional razoável, serem simples e fáceis de descrever e analisar, com parâmetros ajustáveis, e equilibradas (no sen-

tido de serem complexas o suficiente para representar problemas do mundo real e ao mesmo tempo simples o suficiente para obter uma boa compreensão do desempenho de um algoritmo).

Tendo discutido sobre as características das funções de *benchmark* MOO ideal e DSOP ideal, o estudo chega ao ponto central, abordando sobre as limitações dos DMOPs atuais (à época). Helbig and Engelbrecht (2013b) destacam que nenhum dos problemas que foram propostos possuem *PF* isolado ou enganoso, bem como *PS* diferente para cada variável de decisão ou ainda *PS* complexo, ou seja, o *PS* é definido por curvas não lineares no espaço de decisão. Ainda, os autores mencionam que a maioria dos estudos que investigam o desempenho de DMOAs apenas testaram o algoritmo em alguns DMOPs, além de não testar o algoritmo em problemas do mundo real.

Para DMOPs com uma *PF* isolada (com as funções objetivo contendo regiões planas, em que a falta de informações de gradiente causa dificuldades de convergência), o estudo apresenta uma equação, a mesma aplicada ao *benchmark* WFG, em que as regiões planas são criadas mapeando as variáveis de decisão para novos valores. Como exemplo, os autores aplicaram esse mapeamento ao FDA5.

Para DMOPs com uma *PF* enganosa (em que o problema tem, pelo menos, dois ótimos, com o espaço de busca favorecendo o ótimo enganoso), o estudo propõe uma abordagem que pode ser usada para ajustar DMOPs existentes de forma que estes tenham uma *PF* enganosa. Para tal, foi proposta uma função de transformação (novamente, semelhante à abordagem aplicada ao *benchmark* WFG) e para exemplificação, o FDA5 foi modificado para ter *PFs* enganosas.

Para DMOPs com Conjuntos Pareto Ótimos (*PS*) complicados, o estudo propõe três novos DMOPs tendo como base o *benchmark* LZ09. As funções propostas foram constituídas de tal forma que o número de variáveis de decisão possa ser facilmente escalado, ainda, tais funções adicionam dificuldade ao exigir que DMOAs encontre um *PS* definido por curvas não lineares.

Todos os DMOPs propostos foram aplicados a DMOAs, bem como, avaliados para determinar se realmente são difíceis de resolver.

Tendo contribuído de forma ativa, exemplificando como construir difíceis funções de teste dinâmicas, o estudo retorna a teoria ao apresentar características gerais de funções de *benchmark* MOO (estático ou dinâmico), os quais testam de forma suficiente a capacidade dos algoritmos em várias situações típicas do mundo real. Segundo os autores, o conjunto de funções *benchmark* deve testar (1) questões relacionadas a convergência para a verdadeira Fronteira de Pareto (multimodalidade, com ótimo enganoso e ótimo isolado), (2) a diversidade (diante de *PF* convexa ou não convexa, descontínua e com distribuição não uniforme), (3) aos diferentes tipos ou formas de *PS* (incluindo curvas não lineares) e (4) com variáveis de decisão com dependências ou ligações.

Sobre as características que um conjunto de funções de *benchmark* DMOP ideal deve ter, foram sugeridas: (1) distribuição não uniforme de soluções na PF , de modo que a distribuição de soluções mude ao longo do tempo; (2) mudança na forma da PF ao longo do tempo, de convexa para não convexa ou vice-versa; (3) tendo variáveis de decisão com diferentes taxas de mudança ao longo do tempo; (4) casos em que a PF depende dos valores de PS s ou PF s anteriores; (5) possibilidade de alteração do número de variáveis de decisão ao longo do tempo; (6) possibilidade de alteração do número de funções objetivo ao longo do tempo; e (7) inclusão de DMOPs do mundo real.

O estudo conclui apresentado duas tabelas de sugestões de DMOPs, com a primeira relacionando funções de *benchmark* para cada uma das quatro características identificadas para MOOPs em geral e com a segunda relacionando funções de *benchmark* para cada uma das sete características identificadas para DMOO.

Performance measures for dynamic multi-objective optimisation algorithms (2013)

Helbig and Engelbrecht (2013d) enfatizam a inexistência de medidas de desempenho padrão, fornecendo uma visão abrangente das medidas de desempenho utilizados à época, as quais são, em sua maioria, adaptações de medidas de referência comumente utilizadas na otimização estática. No estudo os autores começam apresentando definições importantes, bem como formulações matemáticas para MOOPs e DMOPs, seguindo para uma discussão abrangente sobre as medidas estáticas e suas adaptações. Na discussão os autores discutiram brevemente sobre formas de avaliar as medidas de desempenho e em seguida fizeram descrições das métricas e as apresentaram matematicamente. Ainda, as avaliam e as classificam em: medidas de precisão, medidas de diversidade, e medidas combinadas. E, especificamente para otimização dinâmica, a categoria "robustez", que refere-se a quão bem o algoritmo se recupera após a ocorrência de uma mudança de ambiente.

Seguindo, os autores levantam cinco questões importantes que devem ser levadas em consideração ao selecionar medidas que quantificam o desempenho dos AEMOs. A primeira e segunda questão levanta a possibilidade de as medidas de desempenho retornarem resultados enganosos nas situações em que os algoritmos perdem o controle da PF durante as mudanças de ambiente, ou quando a PF encontrada possui outliers. A terceira questão se refere à algoritmos que não gerenciam violações de restrições de limite, podendo soluções inviáveis serem adicionadas ao PF^* , possibilitando resultados enganosos relacionados ao desempenho de determinado algoritmo (caso em que as soluções inviáveis são dominantes). A quarta questão aborda a possibilidade de as medidas de precisão serem calculadas tanto em relação ao espaço de decisão (distância entre PS^* e PS é calculada) quanto em relação ao espaço objetivo (proximidade entre as soluções não dominadas de PF^* e a PF). Mesmo que o objetivo principal de um DMOP seja rastrear a mudança da PF , o que justifica a métrica ser aplicada sob o espaço objetivo, em algumas situações é útil determinar o quão próximo o PS está do PS^* . A preocupação, é que, em problemas tendenciosos, pode ocorrer que mesmo que o PS^* do algoritmo esteja muito próximo do PS , o

PF^* esteja bem distante do PF . A quinta questão levantada se refere à comparação de desempenho de vários algoritmos e a influência que a seleção das medidas tem na ordenação destes. Isso porque, a seleção de medidas erradas pode ocasionar uma ordenação incorreta dos algoritmos, principalmente quando a fronteira verdadeira dos problemas aplicados não forem conhecidas.

Por fim, os autores ressaltam a importância de novas pesquisas com relação a medidas de desempenho que não sejam vulneráveis aos problemas discutidos, principalmente sobre métricas que possam ser usadas quando a verdadeira PF é desconhecida.

Challenges of Dynamic Multi-objective Optimisation (2013)

Em Helbig and Engelbrecht (2013c), os autores destacam que, apesar de muitos DMOAs terem sido propostos, esse é um campo de pesquisa muito imaturo em comparação com a MOO e a Otimização Dinâmica de Objetivo Único (DSOO). Iniciando, o estudo faz a introdução de definições importantes e seguindo, os autores abordam sobre os **desafios da otimização dinâmica**, estes relacionados a “avaliação de desempenho de DMOAs”, a “falta de algoritmos de Inteligência Computacional avaliados em DMOPs” e o “problema de selecionar soluções preferidas do conjunto de soluções encontrado”. Concluindo, o estudo aponta novas áreas de pesquisa em Inteligência Computacional e como tais avanços impactam o campo da DMOO.

O estudo menciona o desafio de comparar algoritmos existentes, principalmente por não existir um conjunto padrão de funções de *benchmark* dinâmicos. A questão principal relacionada a este desafio é a dificuldade de avaliar o desempenho de vários algoritmos e obter o entendimento sobre para quais tipos de DMOPs determinados DMOAs tem um desempenho muito bom ou muito ruim. Nisso, é enfatizado a necessidade de desenvolvimento e/ou seleção de *benchmarks* com uma variedade de características e dificuldades, representativas de problemas típicos do mundo real.

Outro desafio mencionado se refere aos problemas com as medidas de desempenho, as quais são, em sua maioria, medidas adaptadas, oriundas da otimização estática. O problema dessas adaptações se refere a possibilidade de resultados enganosos, possíveis em determinadas situações. Outra questão mencionada é a necessidade de pesquisas voltadas ao desenvolvimento de medidas de desempenho que não sejam vulneráveis a tais problemas e que não exijam o conhecimento da verdadeira PF .

Ainda, o estudo discute sobre a dificuldade de as medidas obterem informações sobre a capacidade de rastreamento dos algoritmos durante a resolução de DMOPs. Apesar de algumas medidas indicarem se o algoritmo teve ou não um bom desempenho ao longo do tempo, estas não fornecem informações sobre o quão bem o algoritmo rastreou a PF para ambientes específicos durante a execução. Como solução, os autores discutem uma abordagem de “perdas e ganhos” para analisar os resultados dos DMOAs em vários tipos de DMOPs e vários tipos de ambiente e com isso obter o desempenho geral dos algoritmos comparados.

Outro desafio mencionado se refere a necessidade de avaliar o desempenho dos DMOAs propostos em uma variedade de DMOPs, a fim de obter uma melhor com-

preensão do desempenho de cada algoritmo em ambientes específicos. Principalmente, com a intenção de identificar um ou mais que sirvam como referência na avaliação de outros DMOAs recém-propostos.

Por fim, o último desafio apontado no estudo se refere à seleção de uma solução do conjunto de soluções ótimas encontrado por um algoritmo. É mencionado que no campo da MOO varias abordagens foram propostas, sendo então necessário investigar quais dessas abordagens funcionam bem na DMOO, sobretudo abordagens em que as preferências do tomador de decisão sejam incluídas no processo de otimização.

O estudo segue discutindo **áreas de pesquisa emergentes**, sendo tais: hiper-heurísticas, otimização restrita, otimização de muitos objetivos, algoritmos auto-adaptativos e análise formal de paisagem de aptidão.

Sobre as hiper-heurísticas, que tem por objetivo encontrar, para um problema específico, a melhor heurística ou sequência de heurísticas que produzam soluções de qualidade aceitável, os autores destacam a não aplicação no campo da DMOO. A ênfase está na dificuldade de decidir quando mudar a abordagem em ambientes que estão em constantes mudanças, bem como, a dificuldade em ajustar automaticamente os parâmetros de cada heurística, especialmente com um número limitado de iterações, típico de um ambiente dinâmico.

Sobre a otimização restrita, os autores mencionam a inexistência de pesquisas relacionadas a DMOPs restritos, apesar de a maioria dos problemas de otimização do mundo real serem de natureza dinâmica e com restrições. Com isso, tem-se a necessidade de formular problemas de teste com restrições, tanto estáticas quanto dinâmicas, para então determinar quais das técnicas de manipulação de restrições funcionam bem para quais tipos de ambientes dinâmicos restritos.

Sobre a otimização de muitos objetivos (com mais de três objetivos), também é destacado a inexistência de pesquisas relacionadas.

Também, existe a necessidade de investigação de desempenho de DMOAs auto-adaptativos, os quais adaptam automaticamente os valores de certos parâmetros enquanto resolvem o problema de otimização, ou, em outras palavras, os quais otimizam seus parâmetros em paralelo com o rastreamento da mudança da *PF*.

Por fim, os autores abordam sobre a necessidade de uma análise formal do cenário de aptidão dos DMOPs, isso com o objetivo de promover uma maior compreensão do desempenho de DMOAs em tipos específicos de paisagens.

Key Challenges and Future Directions of Dynamic Multi-objective Optimisation (2016)

Em Helbig et al. (2016) os autores destacam que apesar de muitos problemas do mundo real serem de natureza dinâmica e possuindo vários objetivos conflitantes, a maioria das pesquisas se concentram na DSOO ou na MOO, mesmo com a função dos algoritmos que resolvem DMOPs abrangerem as questões de ambos os campos.

Nesse contexto, o estudo dos autores buscou identificar os principais desafios da DMOO, os quais incluem: avaliar e analisar com eficiência o desempenho dos algoritmos; e construir *benchmarks* capazes de avaliar se determinados algoritmos conseguem superar determinadas dificuldades (tais como, *PF* desconectado, *PS* não linear, um ideal enganoso ou isolado, espaçamento de soluções mudando ao longo do tempo, várias formas de *PF*, etc.).

Mais especificamente, o estudo destaca que, apesar do progresso em relação a alguns dos desafios, ainda são necessárias pesquisas adicionais, sobretudo relacionadas: a construção de *benchmarks* que contenham restrições, estáticas e dinâmicas; à otimização dinâmica de muitos objetivos; a determinação de uma maneira eficiente de incorporar as preferências de um tomador de decisão de forma interativa para orientar o processo de busca, e; ao desenvolvimento de algoritmos que resolvem DMOPs de forma eficiente. É destacado quais aspectos desses desafios foram abordados e quais desafios ainda precisam ser resolvidos, por fim, apresenta direções futuras para o campo.

Sob a "avaliação dos DMOAs", o estudo levanta questões relacionadas aos *benchmarks* e as medidas de desempenho.

Inicialmente, os autores mencionam que questões relacionadas à falta de funções de *benchmark* padrão, a falta de medidas de desempenho padrão, a falta de uma abordagem padrão para comparar o desempenho de DMOAs e problemas com medidas de desempenho foram parcialmente abordados em outros estudos. De modo que seu objetivo é a discussão de questões pendentes sobre *benchmarks* e medidas de desempenho.

O estudo destaca que a maioria das pesquisas em DOO se concentrou em DMOPs irrestritos e que apesar de algumas tentativas, mais pesquisas são necessárias para propor um conjunto de funções de *benchmark* que avaliem adequadamente os DMOAs e determinem se eles podem resolver problemas com diferentes tipos de restrições. Também é destacado que a maioria dos DMOPs do mundo real contém restrições, estáticas e dinâmicas, de modo que os algoritmos tenham sua capacidade testada para lidar com tais restrições, principalmente antes de serem aplicados em problemas do mundo real.

Seguindo, o estudo identifica três categorias de MOOPs restritos, assim como três categorias de DSOP restritos, enfatizando que tais classificações devem ser consideradas no contexto DMOO.

Finalizando o tópico, o estudo lista onze DMOPs do mundo real recentemente propostos para avaliar DMOAs. Isso sob a justificativa de que estes devam ser utilizados como um ponto de partida para determinar os vários tipos de restrições que realmente ocorrem nos problemas do mundo real. Sendo necessário análises para identificar o cenário de aptidão, as características *PS* e *PF*, bem como o impacto das restrições no processo de otimização, garantindo assim que mais de um problema com as mesmas características, ou características semelhantes, não seja adicionado

ao mesmo conjunto de *benchmarks*. Enfatizam ainda que essas características do mundo real sejam correlacionadas com as características dos problemas artificiais.

Outra questão levantada no estudo se refere a necessidade de mais pesquisas voltadas ao desenvolvimento de DMOPs que possam gerenciar as incertezas com eficiência, a justificativa é que a incerteza está presente, tanto nas variáveis de decisão, quanto nos objetivos de alguns problemas do mundo real.

Também foi levantado a questão sobre a necessidade de desenvolver DMOPs escaláveis. Os autores mencionam, que para seus conhecimentos, nenhum estudo sobre Otimização Dinâmica de Muitos Objetivos foi conduzido e os DMOPs escaláveis existentes são limitados.

Finalizando as questões relacionadas aos *benchmarks*, os autores mencionam que utilizar somente problemas do mundo real na avaliação de DMOAs, torna o processo de compreensão dos desempenhos dos algoritmos difícil. Desse modo, é necessário a construção de *benchmarks* de referência, os quais testam o desempenho dos algoritmos para tipos específicos de dificuldade, facilitando assim a análise de desempenho dos algoritmos. A ênfase é no estabelecimento de um conjunto padrão de referências para o campo, estes com características diversas, como, por exemplo, *benchmarks* com intervalos de tempo adaptativos entre as mudanças de ambiente (não somente mudanças em intervalos fixos, a cada n interações) e com desafios relacionados ao cenário de *fitness*.

Sobre as **medidas de desempenho**, o estudo destaca que a maioria das que foram propostos para a MOO apresenta o valor médio, assim como o desvio padrão de um conjunto de n medidas. A questão é que no contexto dinâmico, essa abordagem não indica o quão bem o DMOA foi executado nos vários ambientes de um problema e se ele poderia rastrear com sucesso a mudança de PF e/ou PS ao longo do tempo. Ainda, a maioria dos revisores deseja ver o valor médio e o desvio padrão das medidas de desempenho, o que não condiz, por exemplo, com a abordagem “Ganha-Perde”, que foi sugerida em outro estudo.

Outra questão é a necessidade de padronização de todo o processo de comparação (padronizar um conjunto de DMOOP, medidas de desempenho e a avaliação e análise), isso para facilitar comparações cruzadas de algoritmos, permitindo, por exemplo, que pesquisadores comparem o desempenho de seus algoritmos com os já existentes sem a necessidade de implementar estes outros.

Sobre a **tomada de decisão**, o estudo aponta que ao resolver um DMOP, o tomador de decisão tem a intenção de solucionar uma solução para cada ambiente, ou seja, durante o processo de otimização, de forma interativa. Assim, os autores enfatizam a necessidade de mais pesquisas sobre abordagens para incorporar a tomada de decisão ou as preferências do tomador de decisão no DMOA, sendo que até à época somente um estudo limitado sobre o assunto foi conduzido. Especificamente, os autores sugerem desenvolver abordagens que tornam esse processo rápido, visto que tomadas de decisão frequentes, ou interativas, consomem muito tempo.

Ainda, como contribuição, os autores sugerem abordagens que tornem um DMOA global em um DMOA local por meio da tomada de decisão. Desse modo, a otimização inicia com o objetivo de localizar todo o PS , porém, com base nas informações de preferência, o foco na região ótima de Pareto é ditado pelo tomador de decisão.

Finalizando o tópico sobre o processo de tomada de decisão, outra questão é indicada. Tanto as abordagens *offline* quanto às interativas tornam-se mais difíceis em DMOPs com muitos objetivos, isso devido ao fato de que não é uma tarefa trivial visualizar ou apresentar as soluções para o tomador de decisão. Portanto, a forma como apresentar as várias opções ao tomador de decisão, também é um problema, inclusive no campo da MOO (otimização estática).

Concluindo as questões, o estudo apresenta a necessidade de desenvolvimento de **novos DMOAs**. Antes dos argumentos, ainda apresenta as categorias principais dos algoritmos que já foram desenvolvidos, sendo estes: AEs, algoritmos de PSOs (do inglês *Particle Swarm Optimization*), algoritmos de CI (do inglês *Computational Intelligence*), algoritmos baseados em previsão e algoritmos que convertem um DMOP em DSOP.

Os autores enfatizam que a maioria dos algoritmos propostos lutam para convergir para a PF em ambientes de mudança rápida e lutam para resolver DMOPs com PS não lineares, onde há ligações ou dependências entre as variáveis de decisão e onde a PF é enganosa. Argumentam ainda, que a maioria dos que foram propostos foi avaliado em apenas alguns DMOPs, estes, normalmente, fáceis de resolver. Sem contar, que, para conhecimento dos autores, nenhum algoritmo para otimização dinâmica de muitos objetivos foi publicado. Concluindo as questões, o estudo sugere comparar o desempenho dos DMOAs existentes.

Seguindo, os autores discutem o **curso de ação necessário** para enfrentar os principais desafios do campo, assim como, o impacto potencial dessas ações.

Segundo os autores, o principal desafio a ser abordado é o de padronizar a avaliação dos DMOAs, desafio esse que incorpora sub-desafios, devendo ser enfrentado por uma força-tarefa de pesquisadores. Com isso, espera-se que novos *benchmarks* incluam tanto DMOPs com até três objetivos, quanto DMOPs com mais de três objetivos. Uma vez desenvolvidos, os resultados dos DMOAs avaliados devem estar disponíveis gratuitamente, armazenados em um local central de fácil acesso, para uso de outros pesquisadores.

Ainda, segundo os autores, é necessário aumentar o interesse e a participação da comunidade científica. Isso por meio da organização de sessões especiais, tutoriais e competições nas principais conferências da área.

Sobre o **impacto potencial**, os autores mencionam que o crescente interesse no campo da DMOO levará a mais pesquisadores trabalhando na área. E que, ao focar em DMOPs restritos, os algoritmos podem ser aplicados a problemas do mundo real e, como consequência, haverá uma maior interação entre participantes da indústria

e pesquisadores. Como última sugestão, o estudo menciona que as lições aprendidas na resolução de MOOPs restritos e DSOOs restritos devem ser investigadas e, quando apropriado, aplicadas a algoritmos que resolvem DMOPs restritos. Da mesma forma sugerem que, algoritmos desenvolvidos para otimização estática de muitos objetivos devem ser investigados e as lições aprendidas aplicadas ao desenvolver novos DMOAs.

Dynamic Multi-objective Optimization Using Evolutionary Algorithms: A Survey (2017)

Azzouz et al. (2017) publicaram um amplo estudo sobre DMOO utilizando Algoritmos Evolutivos (AE). Os autores enfatizaram que poucos estudos relacionados à resolução de DMOPs foram conduzidos e que, embora esse tipo de problema seja de significativa importância na prática, a aplicação de AEs para resolvê-los não é muito explorada. Apesar da grande atenção dos pesquisadores, a maioria dos trabalhos existentes são restritos ao caso de objetivo único. Com isso, o estudo se dedica a fazer um levantamento dos AEs que foram propostos na literatura para lidar com DMOPs, bem como, apresenta uma visão geral das funções de teste, medidas de desempenho e testes estatísticos mais comumente usados. Finalizando, os autores discutem desafios reais e direções de pesquisa futura.

Ao lidar com DMOPs, AEs tem a tarefa adicional de rastrear as mudanças de ambiente e reagir a elas, e não apenas a de encontrar um PF diversificado e quase ideal. Segundo os autores, as duas maneiras de reagir às mudanças de ambiente são: (1) considerar que cada mudança indica a chegada de um novo problema e reiniciar todo o processo de otimização; ou (2) utilizar o conhecimento sobre a pesquisa anterior para acelerar a otimização atual, iniciada após a ocorrência de uma mudança. Sendo a primeira sensível ao tempo (o tempo pode não ser suficiente para rastrear uma PF de qualidade) e a segunda sensível a adaptabilidade (a convergência durante a execução pode provocar falha de diversidade).

Nesse contexto, de adaptabilidade e rastreamento da PF , a convergência e diversidade se apresentam como exigências conflitantes. Na literatura, os AEs lidam com tais dificuldades utilizando: abordagens baseadas em introdução de diversidade; abordagens baseadas em previsão de mudança; abordagens baseadas em memória; e abordagens paralelas.

Tendo apresentado essas possibilidades, o estudo se inicia apresentando definições importantes e segue apresentando diferentes classificações dos DMOPs, estas baseadas: na frequência (quando a frequência de mudança aumenta, o tempo dedicado à adaptação torna-se menor), na gravidade (uma taxa de gravidade maior indica grandes mudanças no problema enquanto que uma taxa menor indica pequenas mudanças) na previsibilidade das alterações (indica sua regularidade, podendo ser aleatória ou previsível) e na relação entre o PF e o PS (combinações entre os estados “com mudanças” e “sem mudanças”).

Para garantir a adaptabilidade e permitir que AEs mantenham sua capacidade de

reagir com flexibilidade às mudanças, vários **mecanismos para manter a diversidade** na população foram propostos. Assim, o estudo segue descrevendo cada uma das abordagens.

As abordagens baseadas na diversidade, preservam a diversidade ao longo da execução ou aumentam a diversidade em cada detecção de mudanças. Nesse último caso, ações explícitas são tomadas, como reinicialização ou hipermutação. No estudo, os autores identificaram e descreveram três DMOAs que seguem essa abordagem.

As abordagens baseadas em previsão exploram informações passadas e tentam antecipar a localização do novo *PF* ótimo ou do novo *PS* ótimo. Essa estratégia pode ser aplicada em problemas que seguem uma determinada tendência. No estudo, os autores identificaram e descreveram nove DMOAs que seguem essa abordagem.

As abordagens baseadas em memória utilizam uma memória extra para armazenar informações, consideradas úteis, de gerações passadas para guiar buscas futuras. Essa estratégia é interessante quando soluções ótimas retornam repetidamente aos locais anteriores ou em situações em que o ambiente tem um baixo grau de mudança entre uma etapa de tempo e outra. No estudo, os autores identificaram e descreveram quatro DMOAs que seguem essa abordagem.

Nas abordagens paralelas, os algoritmos utilizam várias subpopulações que evoluem simultaneamente em diferentes processadores e trocam algumas informações através de uma rede estruturada. Os DMOAs que implementam essa estratégia são considerados eficientes por possuírem um tempo de execução muito inferior se comparado a uma implementação sequencial. No estudo, os autores identificaram e descreveram três DMOAs que seguem essa abordagem.

Abordagens que convertem o DMOP em vários MOOPs estáticos são úteis quando as mudanças do ambiente são graduais e contínuas, situações estas que dificultam para um DMOA reagir rapidamente e assim convergir para soluções ótimas. Essa abordagem é uma alternativa interessante quando o problema não permite um bom rastreamento das soluções ótimas continuamente, assim, é preferível obter soluções ótimas em alguns instantes de tempo representativos, ao invés de soluções de baixa qualidade em tempo o todo. No estudo, os autores identificaram e descreveram três DMOAs que seguem essa abordagem.

Sobre as abordagens, o estudo conclui apresentando cinco DMOEA que não se enquadram em nenhuma das classificações anteriores, possuindo assim uma abordagem única.

Para cada algoritmo descrito, os autores apresentaram, em forma de tabela, os *benchmarks* em que foram testados, bem como as métricas de desempenho utilizados e outros DMOAs com os quais foram comparados.

Concluindo o assunto acerca das diferentes abordagens utilizadas por DMOAs na DMOO, o estudo segue apresentando vários DMOPs artificiais. Seguindo, o estudo apresenta **aplicações do mundo real**, as agrupando e classificando em: problemas de

controle; problemas de escalonamento; problemas de gestão de recursos; problemas de roteamento; e problemas de projeto mecânico. Para cada uma das categorias é apresentado pelo menos dois exemplos. O estudo finaliza o tópico apresentando, em forma de tabela, os estudos em que foram aplicados cada um dos DMOPs.

Em continuidade, o estudo discute sobre a importância de se aplicar **métricas de desempenho** no processo de comparação e classificação da eficácia dos DMOAs em relação a diferentes requisitos. Enfatizando que a escolha apropriada das medidas, bem como a escolha apropriada dos testes estatísticos, é importante para produzir uma comparação justa.

O estudo segue analisando um amplo conjunto de métricas de desempenho aplicadas na avaliação de diferentes DMOAs. As métricas mencionadas foram descritas matematicamente e agrupadas por categoria: Medidas de Desempenho de Precisão; Medidas de Desempenho de Diversidade; Medidas de Desempenho Combinadas; e Medidas de Desempenho de Robustez.

Seguindo, o estudo menciona a importância de aplicar **testes estatísticos** para determinar se a diferença entre a população de valores, das métricas de desempenho, de fato, não é obtida por acaso. Finalizando a discussão sobre avaliação de desempenho de DMOAs, o estudo apresenta duas tabelas com informações sobre estudos que utilizaram das métricas mencionadas e estudos que aplicaram testes estatísticos paramétricos e não paramétricos.

Chegando ao fim, o estudo destaca que apesar das várias abordagens propostas para resolver DMOPs, muitos desafios não são considerados, principalmente os relacionados à padronização. Os autores destacam que não há funções de *benchmark* dinâmico multiobjetivo padrão, assim como não há métricas de desempenho consideradas padrão. Com isso, o processo de comparação dos diferentes algoritmos propostos torna-se uma tarefa complicada, visto que uma comparação justa exigiria que estes fossem reimplementados e assim reavaliados da mesma forma. Os autores destacam ainda que os testes estatísticos não são muito utilizados, apesar de sua importância e utilidade.

Também, segundo os autores, é muito necessário a condução de estudos que apresentem um **estado da arte** abrangente das funções de *benchmark* existentes e das medidas de desempenho existentes, os quais podem ser aplicados na DMOO. Além de um **estudo estatístico comparativo** de trabalhos representativos de diferentes abordagens, isso usando funções de teste padrão e métricas de desempenho padrão.

Tendo discutido os desafios gerais do campo DMOO, os autores discutem três **oportunidades de pesquisa** deste campo em que se observa uma falta de estudos, sendo estas: otimização dinâmica com restrições; abordagens paralelas dinâmicas; e tomada de decisão automática.

Sobre a otimização dinâmica com restrições, o estudo menciona que apesar das várias técnicas de manipulação de restrições propostas e incorporadas aos AEs, a maioria está restrita à otimização estática. No contexto dinâmico, essas técnicas precisam

considerar as mudanças de ambiente e assim transformar soluções inviáveis em viáveis sempre que as restrições mudarem. Com isso, é importante o desenvolvimento de *benchmarks* que considerem tanto a dinamicidade das funções objetivo quanto às restrições.

Sobre as abordagens paralelas dinâmicas, o estudo menciona que ao lidar com DMOPs, uma restrição de tempo é imposta, visto que os DMOAs devem convergir rapidamente para o verdadeiro *PF* antes de uma mudança. Nesse contexto, algoritmos paralelos surgem como uma alternativa eficiente, apesar de a literatura mostrar que esta representa uma direção de pesquisa menos focada.

Sobre a tomada de decisão, o estudo menciona que os DMOAs precisam considerar que as preferências do usuário mudam ao longo do tempo e com isso, a técnica de manipulação de preferências deve permitir que as preferências sejam adaptadas interativamente ou aprendidas automaticamente durante o processo de otimização.

Finalmente, o estudo conclui sugerindo duas **direções principais de pesquisa futura**: 1. A exploração da elegibilidade de EAs dinâmicos para resolver problemas revelados por tecnologias de casas inteligentes e cidades inteligentes; e 2. Devido ao fato de que os problemas de casas e cidades inteligentes dependem fortemente das preferências do usuário, tais AEs devem considerar as preferências do tomador de decisão ao longo do tempo, bem como considerar a mudança dessas preferências.

Evolutionary Dynamic Multi-objective Optimisation: A Survey (2022)

No recente estudo de Jiang et al. (2022), os autores destacam que o campo da DMOO é relativamente jovem mas em rápido crescimento. Sendo que após quase duas décadas de esforço comunitário, houve avanços significativos em vários tópicos.

Os autores iniciam definindo DMOPs como uma sequência de MOOPs estacionários, apresentando definições comuns para esses dois campos (estático e dinâmico), continuando, apresenta definições específicas para o contexto de DMOO. Sobre as **definições específicas**, os autores descrevem o conceito de frequência das mudanças”, medida em número de gerações ou avaliações de aptidão, e a gravidade da mudança que representa a magnitude da mudança ambiental ou a quantidade de perturbação no sistema atual estacionário. Sobre esse último conceito, os autores mencionam brevemente uma forma de classificação da severidade de mudança dos DMOPs, sendo: severidade mínima, severidade máxima e severidade média.

Seguindo, o estudo destaca as dificuldades e desafios únicos do campo da DMOO, os quais incluem: (1) dificuldade em se construir *benchmark* dinâmicos representativos do mundo real, visto que conjuntos de teste são geralmente de tamanho limitado, não sendo possível combinar em um mesmo conjunto de testes os diferentes recursos dinâmicos com as propriedades dos problemas estáticos; (2) dificuldades dos DMOAs em garantir a diversidade tanto no espaço de decisão quanto no espaço objetivo; (3) dificuldades em relação às métricas de desempenho, isso em relação a dificuldade de quantificar a qualidade de uma sequência de conjuntos de soluções, o que pode ser

computacionalmente caro; (4) complexidade da comparação de DMOAs, especificamente devido a natureza multidimensional do conjunto de soluções ótimas obtidas a cada instante de tempo (PF_t^*); e (5) dificuldade de visualização dos conjuntos de soluções encontradas ao fim de cada ambiente, especialmente para DMOPs com um alto número de objetivos.

Os autores destacam que alguns dos resultados do DSOO parecem ser aplicáveis ao DMOO, no entanto, exigindo adaptações significativas. Assim, com a existência da multiobjetividade, o conjunto de problemas, o design dos algoritmos e sua avaliação de desempenho enfrentam desafios únicos, sendo altamente necessário uma pesquisa separada.

O estudo segue apresentando brevemente vários estudos que contribuam em vários tópicos para o campo da DMOO, assim como várias teses de doutorado. Também apresenta uma “local chave” de promoção à pesquisa relacionada a Computação Evolutiva em Ambientes Dinâmicos e Incertos (ECiDUE, do inglês *Evolutionary Computation in Dynamic and Uncertain Environments*), bem como menciona congressos e simpósios que incluem competições, tutoriais e sessões especiais no campo.

Com isso, o estudo apresenta as seguintes contribuições: Cobertura, ao fornecer uma imagem atualizada sobre os vários tópicos de pesquisa do campo DMOO; Taxonomia, com a categorização dos estudos e detalhamento das várias linhas de pesquisa (as quais incluem regras de classificação dos problemas, métodos de detecção de mudança e novos tipos de mecanismos de resposta às mudanças) e com o fornecimento de tabelas resumos, definições matemáticas e gráficos ilustrativos; União da teoria com a Prática, apresentando como problemas práticos foram resolvidos por DMOAs, demonstrando sua utilidade em aplicações do mundo real; e Oportunidades de Pesquisa, identificando questões em aberto nesse campo, as quais permanecem sem soluções, bem como, destacando uma série de oportunidades para pesquisas futuras.

Para descrever o cenário atual do campo da DMOO, o estudo buscou por artigos relevantes através da condução de uma Revisão Sistemática da Literatura (RSL) guiada por palavras-chave combinadas e aplicadas em diferentes bases de dados.

Iniciando a discussão dos resultados obtidos a partir da análise dos estudos, os autores identificaram cinco diferentes direções de pesquisa: (1) compreensão das características dinâmicas; (2) construção de DMOPs de referência; (3) projeto de DMOAs eficazes; (4) desenvolvimento de métricas de desempenho (5) aplicação de DMOAs na otimização dinâmica de muitos objetivos, na otimização dinâmica multiobjetiva restrita e em aplicações dinâmicas do mundo real.

A partir das direções identificadas, o estudo sugere uma nova taxonomia composta por seis componentes principais, algumas das quais apresentam sub taxonomias, sendo estas: (1) Classificação dos DMOPs; (2) *Benchmarks* dinâmicos; (3) Detecção de mudança; (4) Resposta a mudança; (5) Métricas de desempenho; e (6) Vinculação da teoria com a prática.

Sobre a classificação dos problemas, os autores mencionam duas abordagens de classificação de problemas com base nas características da dinâmica. Ambas as abordagens criam um conjunto de regras de classificação, enquanto a primeira foca no efeito da dinâmica sob o *PS* e/ou *PF* verdadeiros, a segunda estuda a fonte da dinâmica que perturba os problemas de otimização.

A abordagem de classificação baseada em efeito considera a dinâmica do verdadeiro *PS* e/ou *PF*, agrupando os DMOPs em quatro distintos tipos: Tipo I – o *PS* muda ao longo do tempo enquanto o *PF* permanece estacionário; Tipo II – tanto o *PF* quanto o *PS* mudam ao longo do tempo; Tipo III – o *PF* muda ao longo do tempo enquanto o *PS* permanece estacionário; e Tipo IV – tanto o *PF* quanto o *PS* permanecem estacionários, embora as funções/restrições objetivas possam mudar ao longo do tempo.

A abordagem de classificação baseada em causa foca a origem ou a fonte da dinâmica que torna os problemas de otimização variantes no tempo, agrupando os DMOPs em quatro distintos casos: Caso 1 – as variáveis de decisão mudam ao longo do tempo; Caso 2 – as funções objetivo mudam ao longo do tempo; Caso 3 – a função e/ou os parâmetros de decisão dependem de seus valores em ambientes anteriores; Caso 4 – os ambientes evoluem com o tempo devido a restrições dinâmicas. Uma observação é que essa classificação também se aplica a DSOPs, além de os casos não serem mutuamente exclusivos, havendo possibilidade de sobreposição entre eles.

As duas abordagens descritas fornecem algumas diretrizes para a construção de *benchmarks* de referência, sugerindo onde e como introduzir dinâmica em MOOPs estacionários para torná-los dinâmicos.

Sobre os *benchmarks*, os autores destacam que nossa compreensão em relação aos DMOPs é bem limitada, apesar da atenção crescente relacionada a este tópico. Considerando os trabalhos publicados no campo, existem basicamente três grupos de *benchmarks* dinâmicos: DMOPs irrestritos contínuos, DMOPs restritos e DMOPs discretos. O estudo apresenta um resumo desses problemas em forma de tabela, e em material complementar, a descrição matemática detalhada de algumas suítes de teste importantes.

Ainda, sobre o tópico, os autores mencionam que a maioria dos DMOPs são contínuos e irrestritos, provavelmente por se basearem em estruturas de *benchmarks* contínuas existentes na MOO (como ZDT, DTLZ e WFG). Dessa forma, os recursos dinâmicos são adicionados a estruturas estáticas, incluindo *PF/PS* variável no tempo, paisagens de aptidão dinâmicas e ligações dinâmicas entre as variáveis de decisão. Os autores mencionam também a existência de outras abordagens para criar DMOPs irrestritos, com recursos estáticos e dinâmicos. Concluindo, os autores destacam que poucos problemas irrestritos e discretos foram propostos no contexto DMOO.

Sobre a **Detecção de Mudanças**, uma mudança ambiental pode provocar dois cenários em relação às soluções não dominadas do ambiente anterior, (1) ou algumas

destas soluções são dominadas pelas soluções ótimas do ambiente atual (2) ou algumas destas soluções dominam as soluções ótimas do ambiente atual. Para o primeiro cenário, se as soluções ótimas anteriores forem parcialmente ou não dominadas por quaisquer soluções geradas para o ambiente atual, as soluções anteriores precisam ser reavaliadas pois podem não mais serem soluções viáveis. Para o segundo cenário, essas soluções anteriores tornam-se soluções inviáveis, estando fora do espaço objetivo alcançável para o ambiente atual. Independente do cenário, um AEMO deve perceber a ocorrência de mudanças ambientais, embora seja possível que tais algoritmos sejam indiferentes às mudanças ao custo computacional de reavaliar toda a população em cada geração. Com isso, os autores mencionam a existência de três abordagens principais para detecção de mudanças: reavaliação aleatória, detecção baseada em sensor e detecção baseada na população.

Na reavaliação aleatória, a cada geração, os detectores de alteração (uma proporção da população) é selecionada aleatoriamente e reavaliada. Havendo mudanças é sinalizado que houve uma alteração ambiental. Na detecção baseada em sensores, os detectores de alteração são selecionados de forma específica, ou seja, posicionados em regiões estratégicas do espaço de busca, para então serem reavaliados e sinalizar ou não alguma mudança. Uma observação interessante é que estes sensores têm um limite de cobertura de monitoramento, não conseguindo detectar alterações "fora do seu campo de visão". A detecção baseada em população é uma abordagem que combina esquemas baseados em sensores e abordagens estatísticas para mudanças ambientais menos detectáveis. Assim, um teste estatístico é realizado para verificar a existência de alguma diferença significativa entre a população anterior e a população atual. Não havendo uma diferença significativa, um novo teste baseado em sensores é aplicado.

Sobre a **Resposta à Mudança**, que se refere às medidas tomadas pelos AEMOs quando detectado uma alteração ambiental, alguns mecanismos foram propostos na literatura: aumento/manutenção da diversidade, abordagens multipopulacionais, abordagens baseadas em previsão, abordagens baseadas em memórias, pesquisa local, abordagens baseadas em dinâmica, abordagens baseadas em pontos especiais e abordagens baseadas em robustez.

A abordagem de aumento/manutenção da diversidade, introduz alguma diversidade extra na ocorrência de mudanças ou mantém uma alta diversidade na população ao longo do tempo, independente das mudanças. Para aumentar a diversidade, existem duas implementações populares, introduzir soluções aleatórias na nova população ou hipermutar algumas soluções existentes.

As abordagens populacionais criam múltiplas subpopulações com intercomunicação, estas espalhadas no espaço de busca. Assim, a diversidade populacional é mantida, uma vez que não é provável que se perca uma grande quantidade de diversidade no caso de mudanças ambientais, a menos que as várias regiões de busca apresentem mudanças ambientais.

As abordagens baseadas em previsão, utilizam modelos preditivos que podem prever

o novo PS/PF após uma mudança com base em aproximações PS/PF anteriores. Essas abordagens são úteis em ambientes que apresentem dinâmicas previsíveis.

Em abordagens baseadas em memória, soluções específicas de ambientes anteriores são guardadas para posteriormente, quando necessário, serem recuperadas e mescladas na população. Esse tipo de abordagem é útil quando um novo ambiente é semelhante ao ambiente anterior.

Na pesquisa local, a vizinhança das soluções atuais é percorrida para identificar soluções promissoras e direcionar para a PF os novos ambientes. Esse tipo de abordagem é eficiente quando o novo PS está próximo da PS atual.

As abordagens baseadas em dinâmica, tentam lidar com mudanças ambientais avaliando seus efeitos nos DMOAs. Dois tipos dessa abordagem foram investigados: o primeiro responde às mudanças de acordo com sua gravidade (sendo a gravidade alta, o algoritmo realiza um grande ajuste, sendo baixa realiza um ajuste sutil); o segundo assume que nem todas as variáveis de decisão das soluções são afetadas por mudanças (o algoritmo deve possuir diferentes mecanismos de resposta para diferentes tipos de variáveis).

As abordagens baseadas em pontos especiais carregam características importantes da PF , por exemplo a localização dos pontos de joelho (ponto em que uma curva se dobra visivelmente), em que se um objetivo melhorar um pouco, pelo menos um outro objetivo será acompanhado por um sério declínio. Ou a incorporação de pontos especiais de interesse para os tomadores de decisão.

Por fim, as abordagens baseadas em robustez empregam uma otimização robusta ao longo do tempo, com o objetivo de encontrar soluções aceitáveis em diferentes ambientes, evitando a necessidade de rastrear novas soluções em cada mudança ambiental.

Sobre as **Medidas de Desempenho**, estas podem ser categorizadas em: medida de convergência, medida de diversidade, medida de dinâmica e medidas híbridas (que quantificam vários aspectos de desempenho). Os autores disponibilizam em material complementar as definições de diferentes medidas de desempenho.

Medidas de Convergência, em um contexto DMOO, indicam o quão bem os DMOAs convergem para a verdadeira PF_t antes de uma mudança ambiental. Os autores destacam que várias medidas de convergência dinâmica foram propostas, estas baseadas em suas contrapartes estáticas, e que uma estratégia comumente adotada é aplicar uma medida estática ao final de cada período de tempo, apresentando como resultado a média aritmética dos valores.

Medidas de Diversidade quantificam até que ponto as soluções aproximadas são diversificadas no espaço variável ou no espaço objetivo. No contexto DMOO, a média aritmética também pode ser aplicada para adaptar as medidas de diversidade estática.

Medidas Dinâmicas analisam a capacidade dos DMOAs em lidar com a dinâmica do ambiente, como por exemplo, indicar o tempo de reação necessário para o algoritmo se adaptar às mudanças, a estabilidade do algoritmo, a taxa de sucesso de uma detecção de mudança, bem como o custo associado e a pontualidade.

Medidas Híbridas quantificam vários aspectos do desempenho dos algoritmos, novamente, a média aritmética pode ser adotada para medir o desempenho de DMOAs.

Sobre a **Vinculação da teoria com a prática**, os autores identificam aplicações do mundo real, concentrando-se estas em cinco áreas: programação/planejamento, controle/design, fabricação/produção, alocação/gerenciamento de recursos e ciência de dados, sendo a primeira a mais popular. Sobre o tema, os autores identificam estudos relacionados a aplicações práticas do mundo real, no contexto DMOO. Apresentando, em forma de tabela, a tarefa de otimização resolvida, o número e a dinamicidade dos objetivos, a existência de restrições, bem como a sua natureza (estática ou dinâmica), o tipo de variável de decisão e as abordagens utilizadas na detecção de mudanças.

Concluindo, os autores discutem **lições aprendidas** com o desenvolvimento de DMOAs, recomendações e diretrizes de design algorítmico, implicações práticas e oportunidades de pesquisa.

Discutindo as lições aprendidas com o desenvolvimento de DMOAs, os autores focam em três tópicos principais: problemas de referência, desenvolvimento de DMOAs e medidas de desempenho.

Sobre os problemas de referência, os autores destacam uma evolução significativa dos *benchmarks* nas últimas duas décadas, sendo que recursos dinâmicos simples são facilmente adicionados em MOOPs estáticos, estes limitados em termos de impacto. No entanto, nos últimos anos, recursos dinâmicos foram construídos com um nível mais alto de complexidade e diversidade, recursos estes provavelmente encontrados em cenários do mundo real. Como consequência, uma ampla gama de técnicas de manipulação dinâmica foram propostas, baseados na justificativa de que é improvável que uma única técnica tenha um desempenho eficiente em relação aos diversos recursos dinâmicos. Ainda sobre o assunto, os autores destacam que os DMOPs devem possibilitar o entendimento da dinâmica e seus impactos nos processos de otimização, isso porque as propriedades dos *benchmarks* estáticos podem esconder recursos dinâmicos.

Sobre o desenvolvimento de DMOAs, os autores destacam algumas lições aprendidas: (1) misturar abordagens de resposta às mudanças funciona melhor, (2) A previsão da população é útil, visto que a previsão multidirecional tem um desempenho melhor do que a previsão unidirecional, (3) modelos de aprendizagem de máquina melhoram a resposta à mudança, e (4) abordagens orientadas por preferências/referências simplificam a busca de soluções para DMOPs, aliviando a complexidade da resposta à mudança.

Sobre as medidas de desempenho, os autores destacam que a maioria das que existem no campo da DMOO são medidas adaptadas e pertencentes ao campo da MOO estática. Sendo estas limitadas por não serem capazes de medir a resposta e adaptação dos DMOAs às mudanças ambientais. Os autores destacam que medidas recentemente propostas se concentram na dinâmica dos algoritmos em resposta às mudanças, complementando as medidas convencionais de desempenho emprestadas da MOO estática. Porém, o tópico ainda se encontra em aberto, existindo a necessidade de pesquisas voltadas ao desenvolvimento de medidas baseadas em dinâmica para facilitar a avaliação de DMOAs.

Sobre o **design algorítmico**, os autores mencionam que este é o tópico mais ativamente investigado no campo da DMOO. A literatura indica que várias pesquisas diferentes foram conduzidas para este tópico e embora que os algoritmos propostos sejam diferentes, estes compartilham algumas características e etapas de projeto comuns. Com isso, os autores apresentaram as seguintes recomendações e diretrizes para projetar DMOAs: (1) verificar a necessidade de detecção de mudanças, bem como escolher a abordagem e determinar como os algoritmos devem reagir; (2) iniciar utilizando abordagens simples de resposta à mudança para entender se o desafio dos DMOPs considerados é a perda da diversidade; (3) prosseguir adicionando progressivamente outros tipos de abordagem de resposta à mudança até que o algoritmo atenda aos requisitos de desempenho; (4) para diminuir a complexidade e o custo computacional do algoritmo, reduza as abordagens de resposta à mudança; (5) escolha uma estrutura de otimização que se adeque perfeitamente as decisões tomadas nas etapas anteriores para produzir soluções de alta qualidade para DMOPs; e (6) realize melhorias iterativas no algoritmo, avaliando-o sistematicamente em diversos ambientes de teste.

Sobre as **implicações práticas**, os autores mencionam que a DMOO surge frequentemente de aplicações do mundo real, portanto, a investigação deste campo tem profundas implicações práticas. Sendo as pesquisas conduzidas particularmente benéficas para os profissionais que enfrentam e procuram soluções para DMOPs em aplicações do mundo real, podendo estes profissionais utilizarem DMOAs sem necessariamente ter muito conhecimento sobre o seu funcionamento.

Concluindo, sobre as **oportunidades de pesquisa**, os autores destacam que apesar da crescente comunidade no campo da DMOO e do progresso em relação a alguns tópicos, mais esforços precisam ser investidos a fim de resolver diferentes desafios. Com isso, os autores destacam futuras oportunidades de pesquisa, estas relacionadas a: avaliação comparativa de DMOAs, medidas de desempenho, Softwares, manipulação de mudança, dinâmica assíncrona e de ligação temporal, escalabilidade, preferência, robustez e aprendizado de máquina.

Sobre as oportunidades relacionadas a avaliação comparativa, os autores mencionam que apesar dos avanços significativos em relação a DMOPs contínuos, poucos estudos foram conduzidos em relação a outros tipos de DMOPs, como os restritos e discretos. Também destacam a necessidade de *benchmarks* dinâmicos que abordam pelo

menos três objetivos, capazes de equilibrar a dinâmica e a multiobjetividade. Outra oportunidade está relacionada à necessidade de construir um conjunto apropriado de recursos dinâmicos que de fato desafiem os DMOAs. Assim como a construção de DMOPs representativos de cenários do mundo real.

Sobre as oportunidades relacionadas às medidas de desempenho, os autores mencionam que as métricas emprestadas da MOO estática, com algumas poucas exceções, negligenciam a avaliação da resposta dos algoritmos às mudanças. Assim, é necessário o desenvolvimento de diversas medidas de desempenho baseadas em dinâmica, permitindo uma análise sistemática do desempenho dos DMOAs, possibilitando ainda avanços em abordagens de otimização guiadas por indicadores. Os autores mencionam também oportunidades de pesquisa relacionadas a DMOP de dados multidimensionais, os que podem ser comparados utilizando uma técnica de escalarização ou teste estatístico de alta dimensão. Por fim, outra oportunidade relacionada ao tópico se refere à falta de uma representação gráfica eficaz na inspeção de desempenho dos DMOAs, especialmente quando o número de *PFs* a serem exibidos é alto.

Sobre as oportunidades relacionadas aos *softwares*, os autores mencionam que, embora existam estruturas para alguns tópicos dinâmicos, se observa uma falta de plataformas abertas voltadas à implementação, execução, comparação e análise de DMOAs de uso geral. A inexistência de tais plataformas, tem implicações significativas nas pesquisas visto que muitos algoritmos são implementados em diferentes ambientes, bem como são executados em diferentes configurações, o que dificulta comparações justa entre os DMOAs.

Sobre as oportunidades relacionadas ao manuseio das mudanças, os autores destacam que a maioria dos DMOAs utilizam o método de reavaliação para detectar mudanças, porém, ainda não é evidente o quão eficaz é esse método e quais podem ser as consequências se houver uma falha na detecção. Da mesma forma, é necessário investigar o desempenho das abordagens de resposta às mudanças. Nesse sentido, os autores sugerem um conjunto de testes com várias mudanças imprevisíveis, aleatórias ou semi-aleatórias.

Sobre as oportunidades relacionadas à dinâmica assíncrona e de ligação temporal, os autores mencionam que mudanças assíncronas influenciam significativamente o desempenho dos DMOAs. Nesse sentido, seria interessante o desenvolvimento de pesquisas incluindo mudanças assíncronas em restrições e/ou parâmetros ambientais.

Sobre as oportunidades relacionadas à escalabilidade, os autores destacam que apenas alguns poucos estudos investigam DMOPs com mais de três objetivos, sendo que, para o conhecimento dos autores, a escalabilidade dos DMOPs em termos de números de variáveis ainda não é explorada.

Sobre as oportunidades relacionadas à preferência, os autores destacam que essa linha de pesquisa pode ser mais investigada para criar uma diretriz concreta sobre quando e como os pontos especiais devem ser aplicados. Para conhecimento dos au-

tores, apenas alguns estudos investigaram a incorporação de preferência em DMOO para facilitar os desafios de tomada de decisão.

Sobre as oportunidades relacionadas à robustez, que no contexto DMOO pode ser entendida como a capacidade de manter bons resultados em inúmeras mudanças ambientais, os autores mencionam o tópico como uma direção de pesquisa promissora. Assim, evitando que os DMOAs sejam frequentemente solicitados a se adaptar.

Por fim, sobre as oportunidades relacionadas à aprendizagem de máquina, os autores mencionam que o uso de aprendizagem de máquina no aprimoramento das estratégias de busca dos DMOAs tem sido uma nova tendência. Desse modo, os autores sugerem uma investigação mais profunda de modelos avançados de aprendizagem de máquina, sobretudo para lidar com mudanças ambientais complexas e na manipulação de grandes volumes de dados.

3.3 Compilado de Alguns dos Tópicos mais Discutidos

Nesta seção faz um compilado sobre alguns dos tópicos mais discutidos nos estudos de revisão aqui descritos.

Sobre **benchmarks dinâmicos**, Helbig and Engelbrecht (2013b) evidenciaram a falta de DMOPs desafiadores, destacando que nenhum dos problemas possuía características específicas que atribuíam dificuldades aos otimizadores, por exemplo, com *PF* isolado ou enganoso e *PS* complexo ou diferente para cada variável de decisão. Helbig et al. (2016), além de destacar a necessidade da construção de DMOPs de referência (capazes de testar o desempenho dos DMOAs para tipos específicos de dificuldade), destaca a necessidade de desenvolver *benchmark* com restrições tanto estáticas quanto dinâmicas, estes baseados nos vários tipos de restrições que realmente ocorrem nos problemas do mundo real. Os autores também mencionam a necessidade de desenvolvimento de DMOPs escaláveis em termos de objetivos e variáveis de decisão, assim como DMOPs que gerenciam incertezas. Jiang et al. (2022), destacaram a necessidade de pesquisas voltadas ao desenvolvimento de DMOPs restritos e discretos, bem como o desenvolvimento de DMOPs escaláveis, especialmente em relação ao número de variáveis de decisão. Também destacam que novos DMOPs devem ser desafiadores e representativos de cenários do mundo real. Os autores sugerem ainda o desenvolvimento de um conjunto de testes com várias mudanças imprevisíveis, aleatórias ou semi-aleatórias, para investigar o desempenho das abordagens de resposta às mudanças.

Sobre as **medidas de desempenho**, Helbig and Engelbrecht (2013d) ressaltaram a importância de novas pesquisas relacionadas ao desenvolvimento de medidas de desempenho que não sejam vulneráveis a problemas específicos da otimização dinâmica. Além disso, existe a necessidade do desenvolvimento de medidas capazes de avaliar e ordenar de forma eficiente os algoritmos, principalmente quando a *PF* ótima não é conhecida. Helbig and Engelbrecht (2013c) mencionaram, novamente em outro

estudo, a possibilidade de resultados enganosos encontrados por medidas adaptadas da MOO estática, sendo necessário a condução de pesquisas voltadas ao desenvolvimento de medidas que não sejam vulneráveis a determinadas situações, assim como o desenvolvimento de medidas que não exijam o conhecimento da verdadeira PF . Também foi discutido a necessidade de desenvolvimento de métricas capazes de rastrear o desempenho dos algoritmos durante a resolução dos diferentes cenários de otimização dos DMOPs. Jiang et al. (2022), destacam que as métricas emprestadas da MOO estática, com algumas poucas exceções, negligenciam a avaliação da resposta dos algoritmos às mudanças.

Sobre a **comparação dos DMOAs**, Helbig and Engelbrecht (2013c) discute o desafio de comparar os DMOAs propostos, especificamente por não existir um conjunto padrão de *benchmarks* dinâmicos, estes com uma variedade de características e dificuldades representativas de problemas típicos do mundo real. Helbig et al. (2016), destaca a necessidade de padronizar todo o processo de comparação, permitindo assim comparações cruzadas. Azzouz et al. (2017), destacam que a falta de uma padronização das funções de *benchmark* assim como das métricas de desempenho dificultam o processo de comparação dos diferentes DMOAs, visto que uma comparação justa exigiria a reimplementação e reavaliação de todos.

Sobre as **preferências do tomador** de decisão, Helbig and Engelbrecht (2013c) discute o desafio de selecionar uma solução de cada $PF*_t$, sendo necessário investigar quais das abordagens propostas para o campo MOO estático podem ser adaptadas para um contexto dinâmico. Helbig et al. (2016) mencionam a necessidade de incorporar as preferências do tomador de decisão de forma interativa, com o objetivo de tornar esse processo mais rápido e menos custoso, além da necessidade de mais pesquisas relacionadas à forma de apresentar as várias opções ao tomador de decisão, sobretudo em DMOPs com muitos objetivos. Jiang et al. (2022) mencionam oportunidades de pesquisa relacionadas à investigação da incorporação de preferência em DMOO para facilitar os desafios de tomada de decisão.

Sobre **direções futuras**, Helbig and Engelbrecht (2013c) indicam oportunidades de pesquisa relacionadas a hiper-heurísticas, otimização restrita, otimização de muitos objetivos, algoritmos auto-adaptativos e análise formal de paisagem de aptidão. Azzouz et al. (2017) destaca como direções futuras a DMOO com restrições, cujas as técnicas precisam considerar as mudanças do ambiente e quando necessário transformar soluções inviáveis em soluções viáveis, assim como, a construção de *benchmarks* que considerem tanto a dinamicidade das funções objetivo quanto a dinamicidade das restrições. O estudo também destaca como oportunidade de pesquisa as abordagens paralelas, que reduzem o tempo de execução, visto que na DMO, o tempo desempenha um papel decisivo. Também destaca oportunidades relacionadas à tomada de decisão, com as técnicas permitindo que as preferências sejam adaptadas interativamente ou aprendidas automaticamente durante o processo de otimização. Jiang et al. (2022), identificam como direções de pesquisa futura a compreensão das características dinâmicas, a construção de DMOPs de referência, o projeto de DMOAs eficazes, o desenvolvimento de métricas de desempenho e a aplicação de

DMOAs na otimização dinâmica de muitos objetivos, na otimização dinâmica multiobjetiva restrita e em aplicações dinâmicas do mundo real.

Sobre o **desenvolvimento de DMOAs**, Helbig et al. (2016) enfatizam a necessidade de desenvolver DMOAs que convergem em ambientes de rápidas mudanças e também capazes de resolver DMOPs com PS não lineares, com ligações ou dependências entre as variáveis de decisão e onde a PF é enganosa, além do desenvolvimento de DMOAs capazes de resolver DMOPs com muitos objetivos. Jiang et al. (2022) propõe que os DMOAs incorporem abordagens robustas, capazes de manter bons resultados em inúmeras mudanças ambientais, como também utilizar o aprendizado de máquina para aprimorar as estratégias de busca.

Sobre **aplicações práticas**, Azzouz et al. (2017) destacam que a resolução de problemas revelados por tecnologias de casas inteligentes e cidades inteligentes, considerando as preferências do tomador de decisão ao longo do tempo. Helbig et al. (2016) mencionaram os seguintes DMOPs do mundo real descritos na literatura: Planejamento de processamento mineral sob capacidade de equipamentos variados; Controle dinâmico de sobretensão sustentada durante um processo de restauração do sistema de energia; Otimização do fornecimento de energia para fabricação de grãos de magnésia; Alocação de energia no downlink em sistema de satélite distribuído; A produção de polisilício de alta qualidade em um reator químico de deposição de vapor; Otimização de processos químicos; Sistema de apoio à decisão para a indústria de transformação; Agendamento dinâmico de *job shop*; Terraplenagens que envolvem o nivelamento ou modelagem de uma área alvo através do movimento ou processamento da superfície do solo, ou seja, otimização de uma linha de produção; Roteamento dinâmico de veículos e o problema dinâmico do caixeiro viajante (TSP); Otimização da qualidade de serviço para sistemas de aplicativos dinâmicos orientados a dados baseados em nuvem; Planejamento de trajetória online para veículos aéreos não tripulados; Otimização da Gestão de Inventário. Jiang et al. (2022), identificaram aplicações práticas concentradas nas áreas de programação/planejamento, controle/design, fabricação/produção, alocação/gerenciamento de recursos e ciência de dados, sendo a primeira a mais popular.

3.4 Respondendo a Questão de Pesquisa Q1

Q1 - Quais dificuldades os atuais Algoritmos Evolutivos enfrentam na otimização dinâmica e quais estratégias são frequentemente adotadas para superá-las?

Basicamente, os DMOAs enfrentam dois desafios, a detecção de mudanças e a resposta às mudanças. Conforme visto nos estudos descritos, muitas estratégias e abordagens de enfrentamento foram propostas. Sobre a "detecção de mudanças", utilizando estratégias como a reavaliação aleatória, a detecção baseada em sensores e/ou a detecção baseada em população. Sobre a "resposta à mudança" utilizando abordagens de aumento/manutenção da diversidade, abordagens populacionais, abordagens baseadas em previsão, abordagens baseadas em memória, pesquisa

local, abordagens baseadas em dinâmica, abordagens baseadas em pontos especiais e abordagens baseadas em robustez.

Capítulo 4

Proposta de Adaptação do NSGA-DO

O trabalho desenvolvido por Pimenta and de Arruda Camargo (2015) teve como objetivo propor um AG multiobjetivo capaz de orientar as soluções de forma a convergirem para os pontos ideais da Fronteira de Pareto. O algoritmo, chamado pelos autores de *Non-Dominated Sorting Genetic Algorithm Distance Oriented* (NSGA-DO), substitui a distância de multidão do NSGA-II pela distância entre uma solução real encontrada e o ponto ideal mais próximo desta solução. Dessa forma, o algoritmo busca evitar que o processo de seleção do NSGA-II perca algumas soluções ótimas em situações em que o conjunto de soluções não dominadas (*front F1*) ultrapassa o limite populacional. Assim, uma vez encontrada a Fronteira de Pareto, o NSGA-DO utiliza a localização destas soluções para melhorar sua distribuição.

Quando apresentado, o NSGA-DO foi testado para a geração de bases de conhecimento de Sistemas Baseados em Regras Fuzzy (FRBS), os quais foram utilizados na resolução de problemas de classificação, estes considerados problemas discretos, sendo que os resultados obtidos pelo novo AG foram superiores em relação ao NSGA-II, de acordo com Pimenta and de Arruda Camargo (2015); Santana and Camargo (2019). No entanto, o NSGA-DO não havia sido avaliado em outros tipos de problema, com isso, parte desta pesquisa se dedicou a analisar sua estratégia evolutiva na resolução de MOOPs e DMOPs contínuos com diferentes características, assim como, se possível, propor melhorias ao algoritmo de modo que este apresente melhores resultados em relação a convergência e a diversidade.

Portanto, este capítulo inicialmente apresentará os resultados do estudo do desempenho do NSGA-DO na resolução de MOOPs contínuos, enfatizando sua principal deficiência diante de problemas com espaço de busca complexos. Em seguida, propostas de modificação ao AG serão apresentadas, de modo que este obtenha resultados satisfatórios na resolução, tanto de MOOPs como também de DMOPs.

4.1 Análise de Desempenho do NSGA-DO para Problemas de Otimização Contínuos

Ao aplicar o NSGA-DO em MOOPs contínuos, com espaços de busca complexos, o desempenho do AG é insatisfatório, como demonstraremos no Capítulo 5. Nessas situações, ao possibilitar que alguns pontos selecionem mais indivíduos que outros, determinadas regiões ótimas vão sendo privilegiadas, e como consequência, estas regiões concentram cada vez mais indivíduos. Observe na Figura 4.1 o desempenho do algoritmo ao ser aplicado a instância de teste ZDT 1.

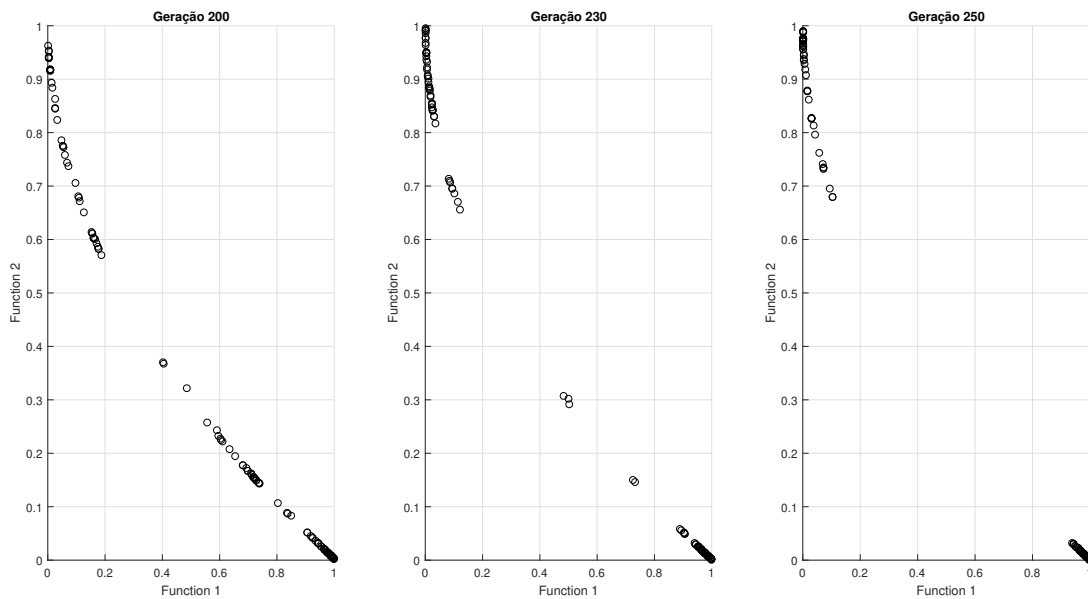


Figura 4.1: PF^* do NSGA-DO ao longo das gerações (Problema ZDT1).

Ao privilegiar determinadas regiões, *gaps* são criados ao longo da PF^* , e os operadores de mutação e cruzamento, cada vez mais, vão perdendo a capacidade de gerar indivíduos em regiões não povoadas. Somente para evidenciar esse problema, o NSGA-DO foi modificado de modo a não possibilitar o privilégio dos pontos mais próximos às melhores soluções, com isso, obtemos uma significativa melhora em relação diversidade. Nessa pequena modificação, baseada na estratégia do NSGA-III, os pontos ideais são selecionados de forma aleatória sendo a solução mais próxima ao ponto eleito adicionado na nova população. A Figura 4.2 apresenta a PF^* do NSGA-DO com essa pequena modificação para o *benchmark* ZDT após 500 gerações.

Podemos concordar que o desempenho do AG foi significativamente melhorado, mesmo que ainda insatisfatório. Com isso, a investigação foi continuada e melhorias assertivas foram propostas, conforme veremos a seguir.

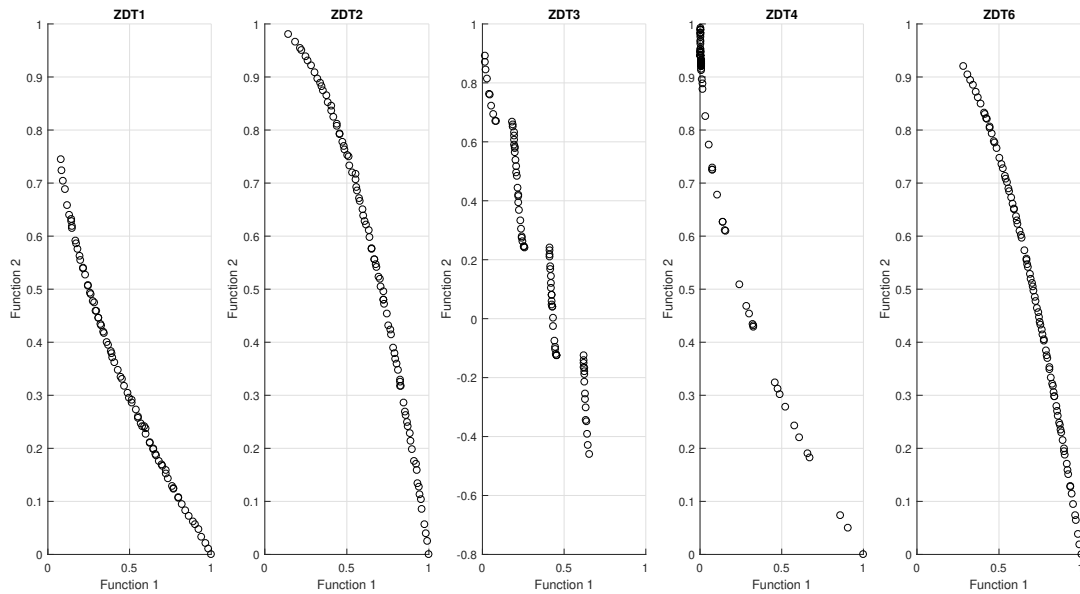


Figura 4.2: Alteração do NSGA-DO para eleição aleatória dos pontos ideais.

4.2 Alterações Realizadas no NSGA-DO

Considerando o estudo realizado na Seção 4.1, o principal fator que resulta em uma má distribuição das soluções na Fronteira de Pareto pelo NSGA-DO é a concentração das soluções em alguns pontos ideais. Se retornarmos à Figura 2.8 (b), podemos observar que o ponto ideal I_4 é o responsável pela seleção de duas soluções, o ponto I_3 é o responsável por selecionar uma, enquanto os pontos I_1 , I_2 e I_5 não selecionam nenhuma. Esse comportamento, portanto, gera a concentração de soluções em certas regiões do espaço objetivo. Conforme demonstrado, a aplicação dos operadores de cruzamento e mutação não conseguem evitar que o NSGA-DO gere fronteiras com *gaps*, estes que ao longo das gerações se tornam piores.

Explicando de uma outra forma, uma vez que um determinado ponto selecione bem mais soluções que outros, a região desse ponto torna-se aglomerada, assim, muito provavelmente, em gerações futuras, pontos serão criados nessa mesma região e todo o processo será repetido.

Outro fator identificado que dificulta a convergência e a preservação da diversidade no NSGA-DO é a quantidade de pontos ideais. Essa quantidade é proporcional ao tamanho do *front* dominante, não existindo um equilíbrio entre quantidade de pontos ideais e quantidade de soluções que é preciso selecionar a partir desses pontos.

Como solução aos dois problemas citados, propomos um novo método de seleção. Inicialmente, os pontos ideais são criados em proporção à duas vezes a quantidade de soluções a serem selecionadas. Posteriormente, o conjunto de pontos ideais é

percorrido, dos extremos ao centro, sendo adicionado na nova população a solução mais próxima ao ponto atual. O Algoritmo 1 define esse novo método de seleção.

Algorithm 1: *Modified* NSGA-DO Selection Operator

Input : Pareto Front (F_1), front in which the solutions are selected (F) and the amount of solutions to be found (S)

Output: Selected solutions set (*SelectedSolutions*)

```

1 IdealPoints  $\leftarrow$  GenerateIdealPoints( $F_1, 2S$ );
2 SelectedSolutions  $\leftarrow$   $\emptyset$  ;
3 alternate  $\leftarrow$  true;
4 Solution  $\leftarrow$  SearchNearestSolution(IdealPoints[0],  $F$ );
5 SelectedSolutions  $\leftarrow$  SelectedSolutions  $\cup$  Solution;
6 Solution  $\leftarrow$  SearchNearestSolution(IdealPoints[|IdealPoints| - 1],  $F$ );
7 SelectedSolutions  $\leftarrow$  SelectedSolutions  $\cup$  Solution;
8 for  $i \leftarrow 2$  to |IdealPoints| do
9   | if alternate = true then
10  |   | index  $\leftarrow$   $i$  ;
11  | else
12  |   | index  $\leftarrow$  |IdealPoints| -  $i$  ;
13  | end
14  | alternate  $\leftarrow$  not alternate;
15  | Solution  $\leftarrow$  SearchNearestSolution(IdealPoints[index],  $F$ );
16  | SelectedSolutions  $\leftarrow$  SelectedSolutions  $\cup$  Solution;
17  | if |SelectedSolutions| =  $S$  then
18  |   | break;
19  | end
20 end

```

Semelhante ao processo de “conversão analógico digital”, em que a taxa de amostragem determina a precisão do sinal resultante, a quantidade de pontos ideais indica sua capacidade de representar a fronteira dominante. A Figura 4.3 ilustra um exemplo de representação de um sinal, com diferentes taxas de amostragem. A imagem à esquerda apresenta um sinal distorcido pela amostragem, com n amostras do sinal real. A imagem à direita apresenta um sinal melhor amostrado, com $2n$ amostras do sinal real. Partindo desse pressuposto, diminuimos o espaçamento ideal do NSGA-DO para forçar a criação de mais pontos, de modo que estes representem melhor a fronteira dominante. Nesse sentido, alteramos a Equação 2.6 para a Equação 4.1, em que S é a quantidade de soluções a serem encontradas.

$$E = \frac{L}{2 * S} \quad (4.1)$$

Ressaltamos aqui, que essa alteração é meramente para fins de precisão, visto que não permitir que um determinado ponto selecione mais soluções que outro, por si só

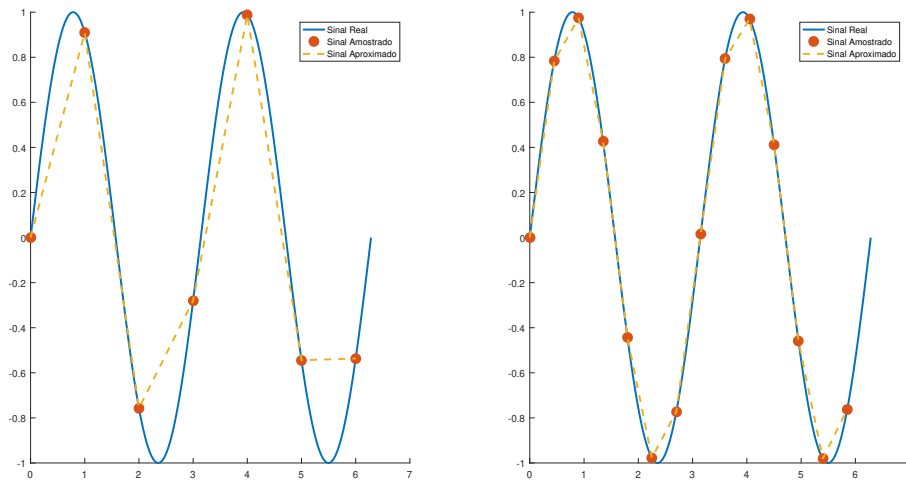


Figura 4.3: Discretização de um sinal contínuo com diferentes taxas de amostragem. A esquerda temos um sinal distorcido pela amostragem, com n amostras do sinal real, a direita temos um sinal melhor amostrado, com $2n$ amostras do sinal real.

já resolve o problema dos *gaps*. Por exemplo, se na Equação 4.1 tivéssemos $3 * S$, a precisão seria ainda maior, obviamente, a um custo computacional também maior.

Por conta da alteração realizada na Equação 2.6, metade dos pontos ideais não selecionam nenhuma solução. O resultado da aplicação do Algoritmo 1 produz, portanto, uma alternância de pontos ideais com e sem o poder de seleção, conforme ilustrado na Figura 4.4. Os pontos marcados com um círculo são os pontos que selecionaram uma solução.

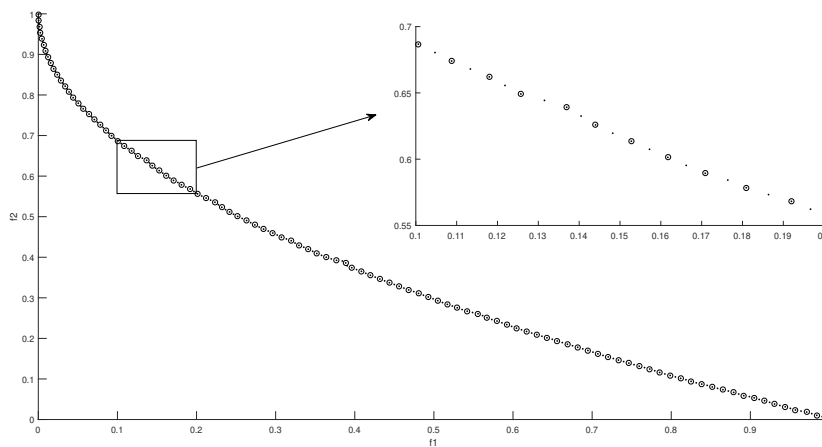


Figura 4.4: Alternância na seleção das soluções pelos pontos ideais.

É interessante destacar também, que diferente do que acontece com o NSGA-DO, o Algoritmo 1 não permite que um mesmo ponto ideal selecione mais de uma solução, evitando assim a concentração de indivíduos em determinadas regiões da Fronteira de Pareto. A modificação realizada resultou no *Modified NSGA-DO* (MNSGA-DO).

Quando observado que o maior problema do NSGA-DO era permitir que determinados pontos selecionem bem mais soluções que outros, foi implementado uma estratégia baseada no AG NSGA-III, semelhante a alteração apresentada no subcapítulo 4.1. Nessa estratégia, pontos ideais eram escolhidos aleatoriamente, sendo a solução mais próxima a este ponto adicionada na nova população. Em seguida o mesmo era removido da lista de pontos disponíveis, isso para que novas soluções não fossem selecionadas a partir dele. Como resultado, a PF^* do NSGA-DO não mais apresentava *gaps*, no entanto, a distribuição das soluções não era uniforme como a distribuição dos pontos ideais.

A estratégia testada, a de selecionar os pontos ideais de forma aleatória, não conseguiu manter uma distribuição uniforme das soluções. Experimentalmente, foi constatado que uma seleção organizada resulta em uma distribuição de melhor qualidade, considerando as análises visuais e a aplicação de diferentes indicadores de qualidade.

É interessante destacar também que as adaptações realizadas foram, em parte, baseadas na observação do processo evolutivo do NSGA-DO, geração-a-geração, por meio de animações reais utilizando o *software* MATLAB e o Framework jMetal (Durillo and Nebro, 2011). Versões recentes do Framework jMetal implementam ferramentas visuais semelhantes.

4.3 Versão Dinâmica do NSGA-DO Modificado

Em Deb et al. (2007), entre outras contribuições, o procedimento do NSGA-II foi modificado para rastrear uma nova Fronteira de Pareto assim que houver uma mudança no problema. Duas variações do algoritmo foram sugeridas para preservar a diversidade após ocorrer uma mudança, sendo elas: substituição aleatória de algumas soluções (DNSGA-II-A) e mutação aleatória de algumas soluções (DNSGA-II-B). As abordagens apresentadas foram testadas e comparadas em problemas de teste artificiais e em um problema de otimização do mundo real.

Nos dois algoritmos variantes foi introduzido um teste que, a cada geração, verifica se houve uma mudança no problema. Esse teste consistiu em escolher aleatoriamente algumas soluções da população de pais e reavaliá-las, havendo mudanças em qualquer um dos objetivos e/ou funções de restrição, é estabelecido que houve uma mudança no problema. Ocorrendo uma mudança, todas as soluções da população de pais são reavaliados usando os objetivos e restrições alterados, isso antes de mesclar com a população de filhos.

Na primeira versão dinâmica do NSGA-II (DNSGA-II-A), sempre que ocorre uma mudança, $\zeta\%$ da nova população é substituída por soluções criadas aleatoriamente.

Na segunda versão dinâmica do NSGA-II (DNSGA-II-B), sempre que ocorre uma mudança, $\zeta\%$ da nova população é substituída por soluções mutantes a partir de soluções existentes escolhidas aleatoriamente. Segundo os autores e conforme os resultados experimentais apresentados no estudo, enquanto a primeira abordagem funciona melhor em problemas que sofrem uma grande mudança nos objetivos e restrições, a segunda funciona bem em problemas que sofrem pequenas mudanças.

A estratégia utilizada considera o problema estacionário por algum período de tempo, de modo que, esse é o período dado para um DMOA encontrar o conjunto de soluções ótimas. Ao fim desse tempo, um novo problema é constituído com base no cenário atual e uma nova otimização é realizada em um novo período de tempo.

Ao detectar uma mudança, os dois DMOAs variantes do NSGA-II, aumentam a diversidade ao introduzir soluções aleatórias ou mutantes, desta forma, um bom nível de diversidade é mantido em todo o processo de busca. Essa é uma estratégia simples, de fácil implementação e com resultados satisfatórios, apesar de ser sensível à frequência de mudança (τT) e à escolha da razão populacional ($\zeta\%$). Conforme constatado pelos autores, existe um limite inferior para τT , no qual, apesar de uma pequena mudança no problema, o número de iterações não é suficiente para um algoritmo rastrear adequadamente as novas soluções ótimas. Os autores constataram também que com a introdução de mais soluções aleatórias o desempenho do algoritmo se deteriora significativamente, enquanto que com a adição de mais soluções mutantes o desempenho se deteriora ligeiramente.

Considerando o resultado deste estudo e devido a fácil implementação da estratégia, o MNSGA-DO foi modificado para resolver DMOPs considerando o DNSGA-II-A. Assim, o algoritmo inicia sua execução como se tivesse a tarefa de resolver um MOOP estático e após a detecção de uma mudança, substitui aleatoriamente $\zeta\%$ soluções da população atual por soluções aleatórias. O que difere é a forma com a qual o algoritmo detecta uma mudança, a qual é baseada em indicadores de qualidade. Nessa estratégia, a fronteira atual é comparada com a fronteira anterior considerando determinado indicador de qualidade, e havendo uma diferença positiva ou negativa em relação a um valor pré-estabelecido, é indicado que houve uma mudança. Para fins de teste, foram implementados também versões dinâmicas do NSGA-DO e NSGA-II.

Capítulo 5

Experimentos e Resultados

Este capítulo inicialmente apresenta um estudo comparativo entre os algoritmos NSGA-DO, MNSGA-DO e NSGA-II na resolução de diferentes MOOPs e DMOPs. O MNSGA-DO é a proposta deste trabalho, a qual modifica o algoritmo NSGA-DO, que por sua vez é uma alteração do conhecido algoritmo NSGA-II.

Os experimentos estão divididos em duas etapas. A primeira está relacionada aos problemas estáticos, mais especificamente, os AGs foram testados nos *benchmarks* contínuos: ZDT, WFG, LZ09 e DTLZ, além de um problema real e discreto do campo de Engenharia de Software, o NRP (do inglês *Next Release Problem*). A segunda etapa de experimentos engloba os problemas dinâmicos, considerando os problemas bi-objetivos do *benchmark* FDA.

A partir dos experimentos apresentados, o capítulo segue apresentando algumas considerações sobre o comportamento dos AGs, especificamente sobre as limitações dos algoritmos estudados e pontos de possíveis melhorias. Por fim, as questões de pesquisa Q2 e Q3 serão respondidas.

5.1 Experimentos com Problemas de Otimização Estáticos Contínuos

Para realização dos experimentos foi utilizado o framework JMetal (Durillo and Nebro, 2011), versão 5.11, o qual implementa os indicadores de qualidade e as instâncias de testes aqui considerados. Os operadores genéticos foram o SBX (do inglês *Simulated Binary Crossover*) (Deb et al., 1995) e a mutação polinomial (Deb and Agrawal, 1999). Esses operadores são comumente utilizados em estudos comparativos de AEs, como por exemplo, nos trabalhos de Deb and Jain (2013); Li et al. (2014); Seada and Deb (2015); Ishibuchi et al. (2016); de Souza Ribeiro (2016); Figueiredo (2017).

O experimento consistiu na aplicação dos problemas ZDT, WFG e LZ09 com dois objetivos e o DTLZ com três objetivos. Por falta de acesso à *PF* ótima das instâncias de teste ZDT5 e LZ09F6, estes testes foram desconsiderados.

Para medir a convergência e a diversidade foram aplicados os indicadores de qualidade Distância Geracional (GD), Distância Geracional Invertida (IGD), *Spread* e Hipervolume (HV). Além disso, o tempo de execução dos AGs também foi examinado. Para analisar a significância das diferenças entre os resultados foi utilizado o teste estatístico de Wilcoxon (Wilcoxon, 1992), com nível de confiança de 95%.

Os parâmetros dos operadores genéticos, de mutação e cruzamento, são apresentados na Tabela 5.1, os quais foram definidos considerando diferentes estudos, conforme explicado nas próximas seções.

Tabela 5.1: Parâmetros dos operadores de mutação e cruzamento.
valores

Operador de Variação	Parâmetro	ZDT	WFG	LZ09	DTLZ
Mutação Polinomial	Probabilidade de mutação (p_m)	$1/N_{var}$			
	Índice de distribuição para mutação (n_m)	20			
Cruzamento Binário Simulado (SBX)	Probabilidade de cruzamento (p_c)	0.9			1.0
	Índice de distribuição para cruzamento (n_c)	30	20	30	

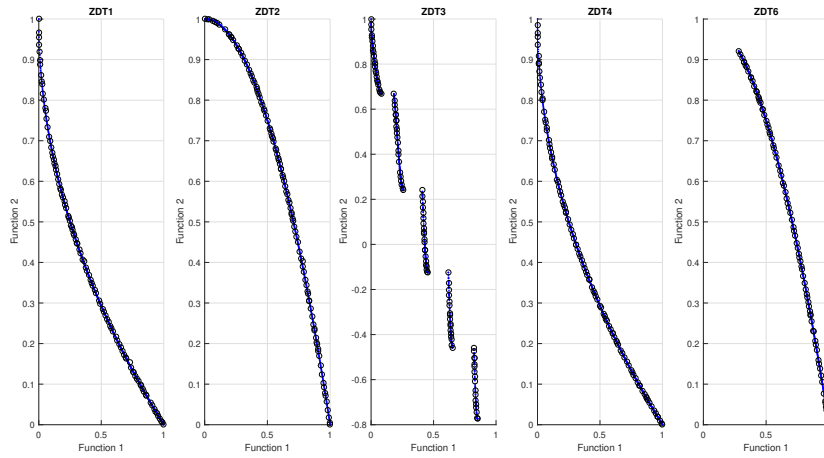
Os resultados serão apresentados em duas formas, através de gráficos das PFs^* dos três AGs, em cada instância de teste, e por meio de tabelas com o valor médio dos indicadores e seu desvio padrão. Para facilitar a análise das tabelas, algumas células possuem um fundo cinza em cada linha. Sendo dois níveis de cinza, com o mais escuro indicando o algoritmo que obteve o melhor desempenho, e o mais claro indicando o algoritmo que obteve o segundo melhor desempenho. Também, o melhor, no caso, o menor desvio padrão foi sublinhado.

5.1.1 Experimentos com Problemas ZDT

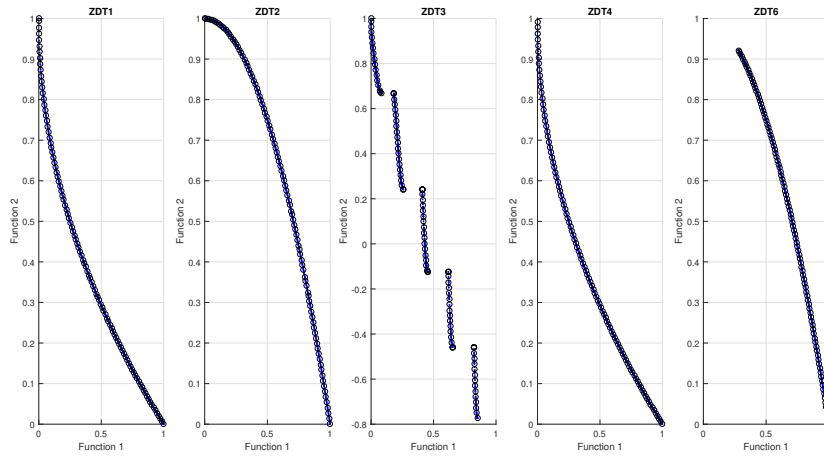
O conjunto de problemas ZDT é amplamente utilizado na literatura de AEMOs. Cinco destes problemas são de valor real, sendo o ZDT5 codificado em binário e não utilizado nessa pesquisa por falta de acesso a sua PF ótima. Os desafios desse *benchmark* incluem multimodalidade (ZDT4), multimodalidade com viés (ZDT6) e descontinuidade, tanto no espaço objetivo quanto no espaço de decisão (ZDT3).

Para os problemas dessa família foram utilizadas 30 variáveis de decisão para as instâncias ZDT{1-3} e 10 para as instâncias ZDT{4 e 6}, como aplicado no estudo original de Zitzler et al. (2000). O critério de parada foi o alcance de 500 gerações. Esta quantidade de gerações é suficiente para o NSGA-II alcançar bons resultados de hipervolume, segundo o estudo de Seada and Deb (2014). Para cada algoritmo, cada instância de teste foi executada 30 vezes.

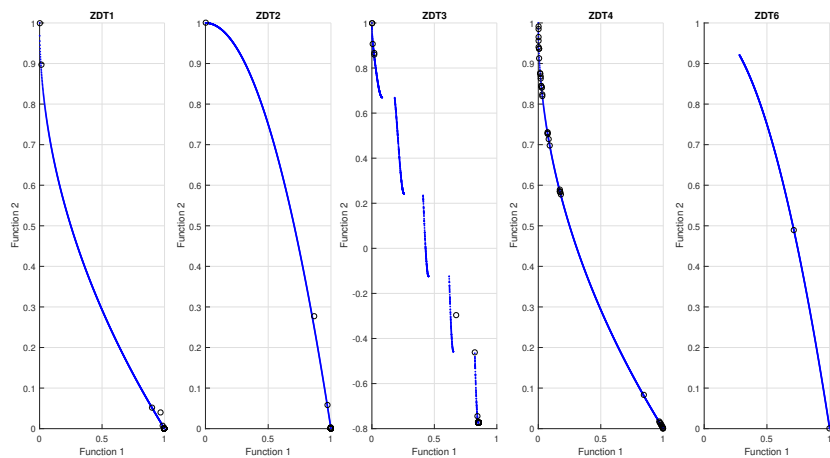
A Figura 5.1 ilustra as melhores fronteiras encontradas por cada um dos três AGs, segundo o indicador de hipervolume (HV), juntamente com a Fronteira Pareto-Ótima.



(a) Algoritmo NSGA-II



(b) Algoritmo MNSGA-DO.



(c) Algoritmo NSGA-DO

Figura 5.1: PFs^* encontradas pelos AGs para o *benchmark* ZDT.

De modo geral, podemos observar que tanto o NSGA-II quanto o MNSGA-DO apresentam um conjunto diversificado de soluções, no entanto, as soluções do MNSGA-DO são melhor distribuídas. O NSGA-DO, por sua vez, concentra as soluções em algumas regiões do espaço objetivo, consequência da sua priorização da convergência em detrimento da diversidade.

Para uma melhor observação da distribuição das soluções ao longo do espaço objetivo, a Figura 5.2 apresenta em um mesmo gráfico a PF^* de cada um dos três AGs. Como as soluções se sobrepõem, parte da PF^* de cada algoritmo foi ampliada e deslocada. Com isso, podemos observar que as soluções do MNSGA-DO realmente possuem uma distribuição mais uniforme.

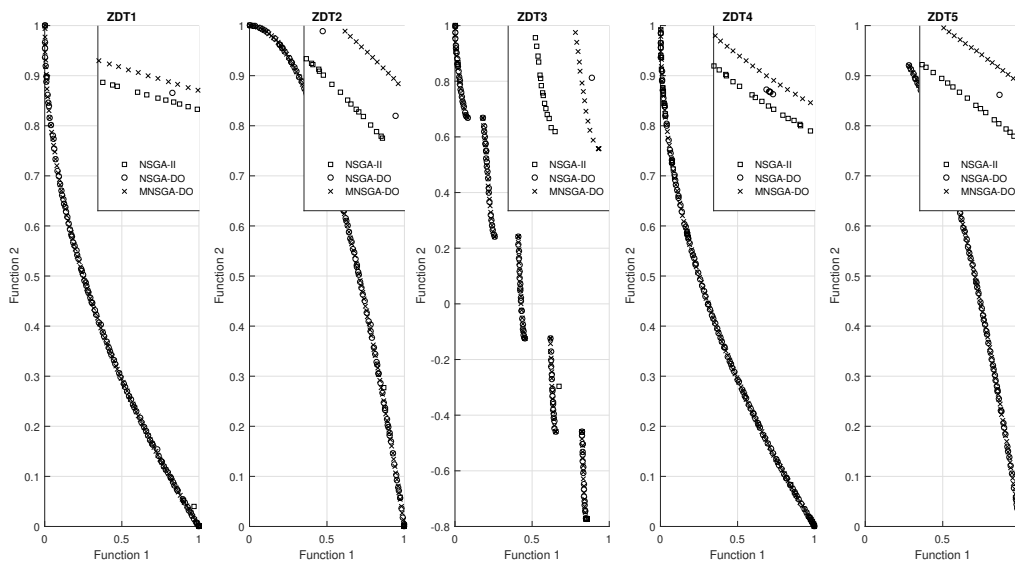


Figura 5.2: Conjunto aproximação dos AGs NSGA-II, MNSGA-DO e NSGA-DO para o *benchmark* ZDT, com parte da PF^* deslocada e ampliada para facilitar observar a distribuição das soluções.

A Tabela 5.2 apresenta os resultados dos indicadores de qualidade. O indicador GD, que mede a convergência, mostrou bons resultados para o NSGA-DO. Isso acontece porque, mesmo com as soluções desse algoritmo concentradas em regiões específicas, essas regiões ainda fazem parte da PF ótima. Porém, ao analisar os valores de desvio padrão, observamos que as maiores variações estão associadas ao NSGA-DO, o que pode indicar uma instabilidade no algoritmo. Para os indicadores IGD, *Spread* e HV, os melhores resultados foram obtidos pelo MNSGA-DO, seguido pelo NSGA-II.

Consolidando as observações, o NSGA-DO aparenta obter bons resultados em relação a convergência, apesar de as soluções convergirem para algumas poucas regiões do espaço objetivo, esse comportamento já foi relatado no trabalho de Figueiredo (2017). Considerando os outros indicadores, que medem também a diversidade, os melhores resultados estão associados ao MNSGA-DO.

Tabela 5.2: Indicadores de qualidade para os problemas da família ZDT.

<i>Generation Distance (GD)</i>						
	NSGA-DO		MNSGA-DO		NSGA-II	
ZDT1	$2.28e-05$	$5.5e-05$	$1.58e-04$	$1.5e-05$	$2.21e-04$	$5.6e-05$
ZDT2	$2.26e-05$	$6.0e-05$	$7.32e-05$	$4.3e-05$	$1.37e-04$	$3.2e-05$
ZDT3	$5.43e-04$	$1.4e-03$	$1.32e-04$	$6.8e-06$	$1.18e-04$	$1.0e-05$
ZDT4	$1.41e-04$	$8.8e-05$	$1.59e-04$	$6.0e-05$	$1.49e-04$	$5.0e-05$
ZDT6	$6.64e-02$	$2.5e-01$	$5.81e-05$	$7.4e-06$	$6.17e-05$	$1.0e-05$

<i>Inverse Generation Distance (IGD)</i>						
	NSGA-DO		MNSGA-DO		NSGA-II	
ZDT1	$1.37e-02$	$6.8e-04$	$1.45e-04$	$4.2e-06$	$1.90e-04$	$8.7e-06$
ZDT2	$1.26e-02$	$9.3e-04$	$1.43e-04$	$4.5e-06$	$1.93e-04$	$1.1e-05$
ZDT3	$1.17e-02$	$1.2e-03$	$1.34e-04$	$2.0e-06$	$2.96e-04$	$6.1e-04$
ZDT4	$1.15e-02$	$2.2e-03$	$1.45e-04$	$9.1e-06$	$1.89e-04$	$8.7e-06$
ZDT6	$2.26e-02$	$5.1e-03$	$1.34e-04$	$2.6e-07$	$2.23e-04$	$1.4e-05$

<i>Spread (SP)</i>						
	NSGA-DO		MNSGA-DO		NSGA-II	
ZDT1	$1.97e+00$	$1.8e-02$	$1.19e-01$	$3.1e-02$	$3.72e-01$	$2.9e-02$
ZDT2	$1.96e+00$	$3.2e-02$	$1.14e-01$	$4.3e-02$	$3.70e-01$	$2.6e-02$
ZDT3	$1.96e+00$	$3.5e-02$	$7.04e-01$	$4.1e-03$	$7.46e-01$	$1.6e-02$
ZDT4	$1.86e+00$	$1.2e-01$	$3.90e-02$	$1.3e-02$	$3.78e-01$	$3.3e-02$
ZDT6	$8.27e-01$	$3.4e-01$	$4.38e-02$	$2.5e-03$	$4.01e-01$	$2.3e-02$

<i>Hipervolume (HV)</i>						
	NSGA-DO		MNSGA-DO		NSGA-II	
ZDT1	$1.83e-02$	$4.5e-02$	$6.62e-01$	$1.5e-04$	$6.60e-01$	$2.9e-04$
ZDT2	$1.39e-02$	$3.0e-02$	$3.28e-01$	$1.8e-04$	$3.27e-01$	$3.2e-04$
ZDT3	$2.31e-02$	$5.1e-02$	$5.15e-01$	$4.6e-05$	$5.15e-01$	$8.6e-04$
ZDT4	$1.62e-01$	$1.3e-01$	$6.60e-01$	$1.0e-03$	$6.59e-01$	$8.1e-04$
ZDT6	$1.32e-02$	$4.6e-02$	$4.01e-01$	$1.8e-04$	$3.98e-01$	$3.6e-04$

A Tabela 5.3 apresenta o tempo de execução dos algoritmos. Como podemos observar, houve uma redução no custo computacional do MNSGA-DO em relação ao NSGA-DO, embora este demande mais tempo em relação ao NSGA-II.

Tabela 5.3: Média e desvio padrão do tempo de execução dos AGs para o ZDT.

	NSGA-DO		MNSGA-DO		NSGA-II	
ZDT1	$3.79e+03$	$6.7e+02$	$2.83e+03$	$5.4e+02$	$2.48e+03$	$5.8e+02$
ZDT2	$3.76e+03$	$5.8e+02$	$2.72e+03$	$4.5e+02$	$2.36e+03$	$5.0e+02$
ZDT3	$3.78e+03$	$6.3e+02$	$2.78e+03$	$5.3e+02$	$2.39e+03$	$5.2e+02$
ZDT4	$3.15e+03$	$4.1e+02$	$2.14e+03$	$3.2e+02$	$1.93e+03$	$4.9e+02$
ZDT6	$2.99e+03$	$3.7e+02$	$1.99e+03$	$3.2e+02$	$1.74e+03$	$2.9e+02$

5.1.2 Experimentos com Problemas WFG

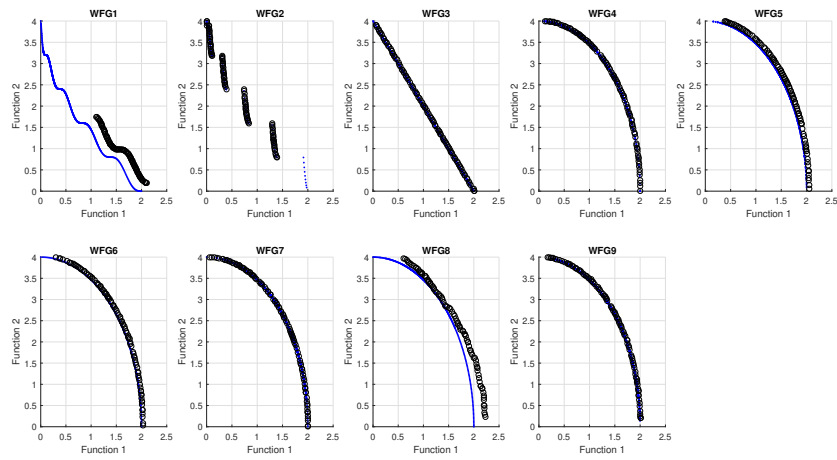
Para os problemas da família WFG foram utilizadas 24 variáveis de decisão em todas as instâncias, com $k = 4$ (parâmetro de posição), $L = 20$ (parâmetro de distância) e 500 gerações. Esses valores para os parâmetros de posição e distâncias, que controlam a complexidade dos problemas, foram os mesmos adotados nos estudos de Huband et al. (2006); Bradstreet et al. (2007); Nebro and Durillo (2009). Nestes trabalhos foi utilizado o algoritmo NSGA-II. Os parâmetros para os operadores de mutação e cruzamento foram baseados nos estudos de Nebro et al. (2008); Gong et al. (2016); Nebro and Durillo (2009), que também utilizaram o NSGA-II. A quantidade de gerações variou entre 200 e 2500.

A Figura 5.3 ilustra as melhores fronteiras encontradas por cada um dos três AGs segundo o indicador de hipervolume (HV). Novamente, observamos que o NSGA-DO, apesar de conduzir as soluções para regiões ótimas do espaço objetivo, apresenta dificuldades em preservar a diversidade nos conjuntos de soluções encontrados, nota-se a existência de *gaps* e aglomeração de soluções. Entre os algoritmos NSGA-II e MNSGA-DO, observamos que ambos apresentam, além da boa convergência, uma boa diversidade. Uma análise mais atenciosa evidencia ainda que o MNSGA-DO apresenta uma melhor distribuição das soluções, apesar de esta não ser tão evidente quanto em relação aos problemas da família ZDT, possivelmente por conta da pouca quantidade de gerações. Prova dessa melhor distribuição são os resultados encontrados para as instâncias de teste WFG1 e WFG2, em que, notoriamente, o MNSGA-DO povoa uma região maior da fronteira.

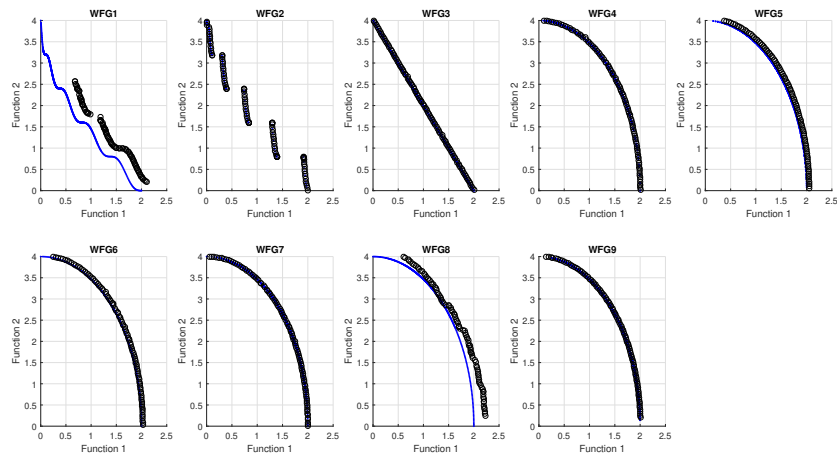
A Tabela 5.4 apresenta os resultados para os indicadores de qualidade. Sobre o indicador GD, não se observa um consenso sobre qual AG obtém os melhores resultados. Em relação ao desvio padrão, o NSGA-DO novamente apresentou os piores resultados, com as maiores variações. Entre os outros dois AGs, o MNSGA-DO apresentou menor variação para a maioria das instâncias de teste.

Segundo os indicadores IGD e HV, o NSGA-II conseguiu superar o MNSGA-DO somente para as instâncias de teste WFG1, WFG2 e WFG8. Enquanto que o indicador SP indica que o NSGA-II foi superado em todas as instâncias de teste. Como esperado, o NSGA-DO apresentou os piores resultados. Sobre o desvio padrão, para a maioria das instâncias de teste, as menores variações estão associadas ao MNSGA-DO seguido pelo NSGA-II.

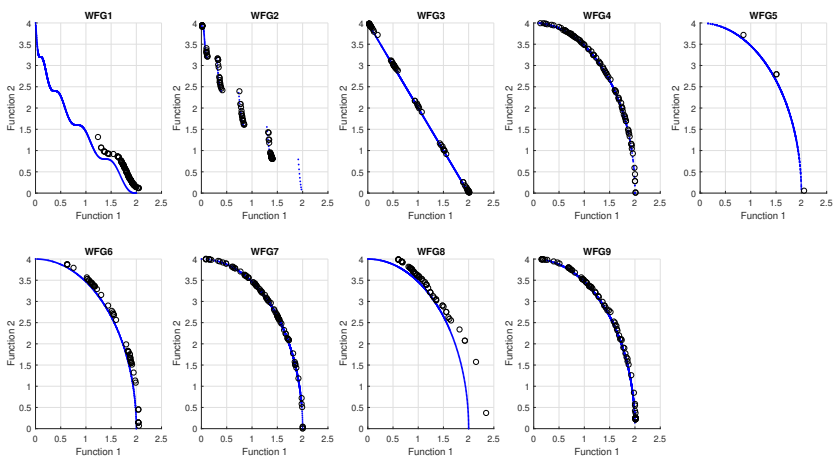
Sobre esse *benchmark*, devido a grande dificuldade apresentada por algumas de suas instâncias de teste, alguns experimentos não convencionais foram apresentados à literatura. Nos estudos de Nebro et al. (2008) e Nebro and Durillo (2009), AEs foram comparados avaliando a quantidade de gerações necessárias para alcançar determinado índice de qualidade. Também, no estudo de Huband et al. (2006), AEs foram comparados variando a quantidade de gerações, isso para avaliar se os mesmos conseguem superar os desafios das instâncias de teste WFG1 e WFG8. Experimentos como estes são interessantes pois avaliam toda a capacidade dos algoritmos ao não restringir a execução a uma pequena quantidade de gerações.



(a) Algoritmo NSGA-II.



(b) Algoritmo MNSGA-DO.



(c) Algoritmo NSGA-DO.

Figura 5.3: PFs^* encontradas pelos AGs para o benchmark WFG.

Tabela 5.4: Indicadores de qualidade para os problemas da família WFG.

Generation Distance (GD)

	NSGA-DO		MNSGA-DO		NSGA-II	
WFG1	1.01e-02	7.5e-03	4.14e-03	2.1e-03	3.33e-03	2.0e-03
WFG2	6.36e-04	5.2e-04	5.20e-04	3.7e-05	5.08e-04	4.1e-05
WFG3	3.49e-04	1.1e-04	4.06e-04	6.9e-05	5.32e-04	7.3e-05
WFG4	1.51e-03	3.1e-04	6.85e-04	3.3e-05	6.98e-04	4.7e-05
WFG5	1.54e-02	5.7e-03	2.72e-03	9.0e-06	2.70e-03	2.8e-05
WFG6	1.52e-03	2.8e-04	2.21e-03	4.3e-04	2.28e-03	3.4e-04
WFG7	2.17e-03	3.0e-04	3.21e-04	1.5e-05	3.77e-04	4.0e-05
WFG8	5.25e-03	7.7e-04	6.44e-03	8.5e-05	6.10e-03	1.6e-04
WFG9	1.66e-02	1.7e-02	3.37e-03	2.0e-03	3.90e-03	1.9e-03

Inverse Generation Distance (IGD)

	NSGA-DO		MNSGA-DO		NSGA-II	
WFG1	1.56e-02	1.4e-03	8.18e-03	3.6e-03	4.39e-03	7.5e-04
WFG2	1.62e-02	6.9e-03	8.03e-03	2.0e-03	7.42e-03	1.3e-03
WFG3	7.14e-03	4.7e-03	1.98e-04	1.9e-05	2.64e-04	2.0e-05
WFG4	2.06e-03	2.3e-03	1.73e-04	7.0e-06	1.91e-04	1.5e-05
WFG5	2.68e-02	4.9e-03	9.53e-04	2.9e-06	9.75e-04	6.2e-06
WFG6	1.70e-02	5.6e-03	1.11e-03	2.1e-04	1.16e-03	1.7e-04
WFG7	4.61e-03	3.1e-03	1.00e-04	3.0e-06	1.27e-04	8.3e-06
WFG8	1.18e-02	5.9e-03	1.99e-03	8.0e-05	1.93e-03	6.0e-05
WFG9	7.91e-03	5.0e-03	6.07e-04	4.7e-04	7.42e-04	4.5e-04

Spread (SP)

	NSGA-DO		MNSGA-DO		NSGA-II	
WFG1	1.25e+00	1.3e-01	5.74e-01	1.0e-01	5.80e-01	3.0e-02
WFG2	1.56e+00	1.9e-01	7.63e-01	9.3e-03	7.81e-01	1.1e-02
WFG3	1.83e+00	1.4e-01	1.50e-01	2.3e-02	3.43e-01	3.2e-02
WFG4	1.35e+00	2.6e-01	2.82e-01	1.8e-02	3.68e-01	2.9e-02
WFG5	1.03e+00	1.1e-01	2.53e-01	9.2e-03	3.92e-01	2.8e-02
WFG6	1.80e+00	9.1e-02	2.65e-01	1.5e-02	3.81e-01	3.0e-02
WFG7	1.79e+00	1.2e-01	2.47e-01	1.8e-02	3.57e-01	2.8e-02
WFG8	1.74e+00	6.9e-02	3.68e-01	2.3e-02	4.19e-01	2.2e-02
WFG9	1.32e+00	5.5e-01	3.02e-01	2.5e-02	4.20e-01	3.2e-02

Hypervolume (HV)

	NSGA-DO		MNSGA-DO		NSGA-II	
WFG1	2.63e-01	3.9e-02	4.65e-01	7.0e-02	5.41e-01	3.0e-02
WFG2	3.90e-01	1.3e-01	5.56e-01	1.2e-02	5.59e-01	7.2e-04
WFG3	2.69e-01	1.5e-01	4.90e-01	1.1e-03	4.88e-01	1.2e-03
WFG4	1.66e-01	4.3e-02	2.16e-01	4.2e-04	2.15e-01	5.3e-04
WFG5	5.58e-03	2.1e-02	1.96e-01	1.1e-04	1.94e-01	3.4e-04
WFG6	2.71e-02	3.2e-02	1.82e-01	5.2e-03	1.81e-01	4.1e-03
WFG7	9.70e-02	5.0e-02	2.09e-01	2.4e-04	2.08e-01	4.0e-04
WFG8	2.88e-02	2.8e-02	1.62e-01	1.9e-03	1.63e-01	1.5e-03
WFG9	6.09e-02	7.0e-02	2.05e-01	3.0e-02	1.96e-01	2.9e-02

5.1.3 Experimentos com Problemas LZ09

Para os problemas da família LZ09 foram utilizadas 10 variáveis de decisão para as instâncias F7 e F8, 30 para as demais e 1500 gerações. Os parâmetros para os operadores de mutação e cruzamento foram baseados nos estudos de Nebro et al. (2013); Navas and Urbaneja (2013); Chen et al. (2016), os quais usaram o NSGA-II.

Uma característica desse conjunto de problemas são seus espaços de decisão complexos, enquanto que os respectivos espaços objetivos se apresentam todos da mesma forma, uma PF côncava, com exceção da instância de teste LZ09 que possui uma PF com forma convexa. Nesse sentido, ainda que os algoritmos rastreiem de forma satisfatória a PF ótima, o que realmente importa é a distribuição dos vetores de decisão, ou em outras palavras, a distribuição dos parâmetros das soluções ótimas. Deve-se observar, que estudos envolvendo esse *benchmark* e utilizando o NSGA-II, o desempenho do NSGA-II é insatisfatório, mesmo dispondo de muitíssimas gerações (> 10000 gerações). Com isso, muitos dos estudos utilizaram o NSGA-II baseado em decomposição, melhorando assim os resultados.

Para as instâncias de teste F1 - F5, o conjunto de pareto é constituído por várias curvas não lineares no espaço de decisão. F7 e F8 são instâncias semelhantes a F1, porém, estas possuem muitas soluções em ótimos locais, sendo o F8 considerado mais difícil que o F7. Para F9, a fronteira de pareto é côncava e seu conjunto de pareto é o mesmo que o de F2.

Como mencionado, o objetivo do conjunto de testes LZ09 é rastrear uma distribuição dos parâmetros no espaço de decisão. Com relação a isso, e conforme o esperado, nenhum dos AGs aqui estudados demonstraram resultados minimamente satisfatórios, ainda que visivelmente o NSGA-II e o MNSGA-DO, considerando o espaço objetivo, tenham convergido e apresentado boa distribuição das soluções.

As imagens das PF^* encontradas pelos três AGs não são apresentadas porque, como mencionado, o objetivo desse conjunto de testes é rastrear o PS ótimo. Também, os conjuntos de parâmetros ótimos de cada algoritmo (PS^*) não serão apresentados por estes se distanciar consideravelmente do esperado, além de não dispormos dos PSs ótimos, o que dificulta uma comparação visual.

A Tabela 5.5 apresenta os resultados dos indicadores de qualidade. Segundo o indicador GD, os melhores resultados estão associados ao MNSGA-DO, com exceção do LZ09F9. Os indicadores IGD e HV concordam que o NSGA-II obtém os melhores resultados, com exceção dos problemas LZ09{7-8}. O SP atribuiu melhores resultados ao NSGA-II em metade dos testes, enquanto que para a outra metade considerou o MNSGA-DO com melhor desempenho.

Tabela 5.5: Indicadores de qualidade para os problemas da família LZ09.

Generation Distance (GD)

	NSGA-DO		MNSGA-DO		NSGA-II	
LZ09F1	6.17e-03	3.3e-03	1.08e-03	<u>2.3e-04</u>	1.74e-03	9.8e-04
LZ09F2	6.32e-03	2.8e-03	1.08e-03	<u>1.9e-04</u>	1.61e-03	7.8e-04
LZ09F3	5.90e-03	3.3e-03	1.08e-03	<u>3.3e-04</u>	1.75e-03	1.5e-03
LZ09F4	6.15e-03	3.5e-03	1.19e-03	<u>4.1e-04</u>	2.01e-03	2.3e-03
LZ09F5	6.52e-03	3.4e-03	1.59e-03	1.2e-03	1.70e-03	<u>9.3e-04</u>
LZ09F7	1.46e-02	6.4e-03	9.81e-04	<u>1.2e-03</u>	1.98e-03	3.0e-03
LZ09F8	1.67e-02	8.7e-03	1.36e-03	<u>1.7e-03</u>	1.91e-03	2.1e-03
LZ09F9	3.14e-02	6.1e-03	2.49e-02	7.2e-04	2.46e-02	4.9e-04

Inverse Generation Distance (IGD)

	NSGA-DO		MNSGA-DO		NSGA-II	
LZ09F1	1.18e-02	4.9e-03	1.57e-03	7.2e-04	5.50e-04	<u>3.8e-05</u>
LZ09F2	1.29e-02	6.3e-03	1.54e-03	6.0e-04	5.50e-04	<u>6.6e-05</u>
LZ09F3	1.43e-02	5.7e-03	1.44e-03	6.9e-04	5.50e-04	<u>5.7e-05</u>
LZ09F4	1.23e-02	5.0e-03	1.57e-03	6.3e-04	5.47e-04	<u>4.5e-05</u>
LZ09F5	1.29e-02	6.4e-03	1.41e-03	4.5e-04	5.45e-04	<u>3.6e-05</u>
LZ09F7	2.84e-02	1.1e-02	3.52e-04	3.3e-05	3.96e-04	<u>1.5e-05</u>
LZ09F8	2.56e-02	1.1e-02	3.55e-04	3.7e-05	3.97e-04	<u>1.9e-05</u>
LZ09F9	1.36e-02	3.8e-03	1.08e-02	7.0e-05	1.07e-02	3.0e-05

Spread (SP)

	NSGA-DO		MNSGA-DO		NSGA-II	
LZ09F1	1.40e+00	1.2e-01	4.34e-01	<u>4.0e-02</u>	4.35e-01	5.1e-02
LZ09F2	1.40e+00	1.4e-01	4.33e-01	4.4e-02	4.29e-01	3.9e-02
LZ09F3	1.45e+00	1.5e-01	4.30e-01	<u>5.7e-02</u>	4.41e-01	6.3e-02
LZ09F4	1.44e+00	1.3e-01	4.40e-01	<u>6.5e-02</u>	4.38e-01	1.0e-01
LZ09F5	1.39e+00	1.2e-01	4.73e-01	1.1e-01	4.39e-01	<u>5.5e-02</u>
LZ09F7	1.58e+00	1.2e-01	3.47e-01	<u>1.1e-01</u>	4.35e-01	1.2e-01
LZ09F8	1.52e+00	1.2e-01	3.81e-01	1.4e-01	4.40e-01	<u>1.1e-01</u>
LZ09F9	1.43e+00	1.2e-01	4.43e-01	8.0e-02	4.22e-01	4.2e-02

Hipervolume (HV)

	NSGA-DO		MNSGA-DO		NSGA-II	
LZ09F1	4.65e-01	4.9e-02	6.44e-01	3.8e-03	6.48e-01	8.0e-04
LZ09F2	4.44e-01	8.8e-02	6.44e-01	3.3e-03	6.48e-01	8.4e-04
LZ09F3	4.12e-01	1.2e-01	6.45e-01	3.3e-03	6.48e-01	7.4e-04
LZ09F4	4.57e-01	4.8e-02	6.44e-01	2.9e-03	6.48e-01	8.8e-04
LZ09F5	4.48e-01	1.0e-01	6.44e-01	2.7e-03	6.48e-01	8.8e-04
LZ09F7	2.01e-01	1.9e-01	6.55e-01	1.1e-03	6.53e-01	<u>5.2e-04</u>
LZ09F8	2.67e-01	2.0e-01	6.55e-01	1.1e-03	6.53e-01	6.6e-04
LZ09F9	4.43e-01	6.4e-02	6.45e-01	2.9e-03	6.48e-01	6.1e-04

5.1.4 Experimentos com Problemas DTLZ

Para os testes com três objetivos foram aplicados os problemas da família DTLZ. Considerando os estudos Deb and Jain (2013); Li et al. (2014); Seada and Deb (2014), o tamanho populacional foi definido em 92 e a quantidade de gerações 1000 (máximo de gerações utilizado nos estudos considerando as instâncias de teste com três objetivos).

Esse *benchmark* contém nove problemas de otimização, sendo quatro destes aqui considerados, DTLZ{1-4}. O número total de variáveis de decisão para este conjunto de problemas é $n = M + k - 1$, em que M define a quantidade de objetivos e k estipula a dificuldade de resolução do problema. Como nos trabalhos de Deb et al. (2005); Deb and Jain (2013); de Souza Ribeiro (2016), especificamos $k = 5$ para DTLZ1 e $k = 10$ para DTLZ{2-4}.

A instância DTLZ1 possui uma fronteira linear com vários ótimos locais ($11^k - 1$); DTLZ{2-4} possuem fronteiras convexas, sendo o DTLZ3 também multimodal e o DTLZ4 tendencioso.

A Figura 5.4 ilustra as PFs^* encontradas por cada um dos três algoritmos, considerando o indicador HV. Com a análise visual, constatamos que nenhum dos algoritmos consegue aproximar de forma satisfatória seu conjunto de soluções do conjunto de soluções considerado ótimo. No entanto, uma simples análise visual nos permite constatar que o MNSGA-DO supera o NSGA-DO, embora o mesmo seja superado pelo NSGA-II, que, de modo geral, apresenta uma melhor distribuição das soluções.

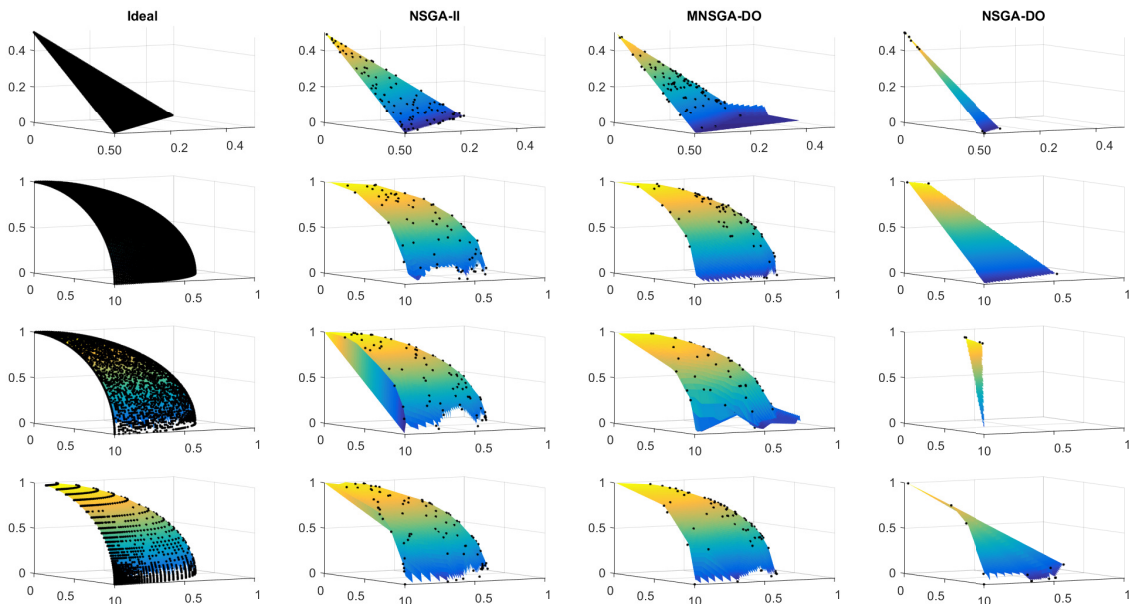


Figura 5.4: PFs^* encontradas pelos AGs para o *benchmark* DTLZ.

A Tabela 5.6 apresenta os resultados dos indicadores de qualidade utilizados. Para o IGD e HV, os melhores resultados estão associados ao NSGA-II. De modo geral, os melhores resultados estão associados ao NSGA-II, embora não se note um consenso nos indicadores GD e *Spread*, que medem somente a convergência e somente a diversidade, respectivamente. Desconsiderando as exceções, os melhores valores de desvio padrão também estão associados ao NSGA-II.

Tabela 5.6: Indicadores de qualidade para os problemas da família DTLZ.

<i>Generation Distance (GD)</i>						
	NSGA-DO		MNSGA-DO		NSGA-II	
DTLZ1	4.65e-02	1.9e-01	3.29e-02	6.8e-02	2.42e-02	9.3e-02
DTLZ2	1.54e-03	1.5e-04	9.84e-04	1.6e-04	1.31e-03	1.6e-04
DTLZ3	1.90e-02	9.5e-02	1.24e-02	4.6e-02	1.44e-03	5.4e-04
DTLZ4	5.31e-03	5.6e-03	5.44e-03	1.2e-03	4.84e-03	1.3e-03

<i>Inverse Generation Distance (IGD)</i>						
	NSGA-DO		MNSGA-DO		NSGA-II	
DTLZ1	5.73e-03	1.3e-03	1.49e-03	4.8e-04	6.08e-04	5.6e-05
DTLZ2	6.41e-03	7.4e-04	1.24e-03	3.2e-04	7.60e-04	3.9e-05
DTLZ3	1.12e-02	7.1e-04	2.16e-03	7.1e-04	1.22e-03	5.7e-05
DTLZ4	7.15e-03	1.6e-03	2.88e-03	1.8e-03	1.77e-03	2.1e-03

<i>Spread (SP)</i>						
	NSGA-DO		MNSGA-DO		NSGA-II	
DTLZ1	1.65e+00	2.9e-01	8.61e-01	2.5e-01	8.55e-01	1.6e-01
DTLZ2	1.91e+00	9.3e-02	6.48e-01	5.8e-02	6.90e-01	4.9e-02
DTLZ3	1.90e+00	1.5e-01	7.35e-01	1.6e-01	7.20e-01	5.3e-02
DTLZ4	1.62e+00	2.6e-01	5.76e-01	1.3e-01	6.81e-01	9.3e-02

<i>Hipervolume (HV)</i>						
	NSGA-DO		MNSGA-DO		NSGA-II	
DTLZ1	1.01e-01	1.1e-01	6.50e-01	4.9e-02	7.61e-01	5.7e-03
DTLZ2	6.45e-03	1.8e-02	3.49e-01	2.4e-02	3.75e-01	6.6e-03
DTLZ3	5.37e-03	1.8e-02	3.30e-01	3.3e-02	3.72e-01	8.8e-03
DTLZ4	3.48e-02	6.7e-02	3.24e-01	6.7e-02	3.51e-01	9.6e-02

Aqui, percebemos que a estratégia do MNSGA-DO, embora muito satisfatória para otimização bi-objetivo, é mais sensível à escalabilidade de objetivos. Em parte, essa sensibilidade está relacionada à sua natureza, que é baseada em dominância de pareto. Mesmo com isso, um estudo recente sugere que AEs desse tipo degradam severamente para três tipos de instâncias de teste de muitos objetivos, estes semelhantes aos problemas DTLZ{1-4} (Ishibuchi et al., 2020). Nos experimentos do citado estudo, o NSGA-II supera diferentes algoritmos baseados em decomposição, inclusive em MOOPs com oito e dez objetivos.

5.1.5 Comparação Estatística dos Resultados

Para verificar a existência de diferença estatística significativa entre os resultados, foi aplicado o teste *Wilcoxon rank-sum*. Os resultados estão descritos na Tabela 5.7. Um símbolo \blacktriangle ou ∇ implica um *p-value* inferior a 0.05, indicando que a hipótese nula é rejeitada. Caso contrário, um símbolo - é utilizado. A ocorrência do \blacktriangle indica que o algoritmo da linha foi significativamente melhor do que o algoritmo da coluna. Por sua vez, o ∇ indica o oposto. Cada símbolo -, \blacktriangle ou ∇ se refere a uma função de teste, ou seja, considerando a família de problemas ZDT, o primeiro símbolo se refere à função ZDT1, o segundo símbolo se refere à função ZDT2, e assim por diante, até o último símbolo que se refere a função ZDT6.

Tabela 5.7: Comparação estatística dos resultados dos indicadores de qualidade.

		ZDT		WFG		LZ09	
		MNSGA-DO	NSGA-II	MNSGA-DO	NSGA-II	MNSGA-DO	NSGA-II
GD	NSGA-DO	$\blacktriangle\blacktriangle\blacktriangle - \nabla$	$\blacktriangle\blacktriangle\blacktriangle - \nabla$	$\nabla - \blacktriangle\nabla\blacktriangle\nabla\blacktriangle\nabla$	$\nabla - \blacktriangle\nabla\blacktriangle\nabla\blacktriangle\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla$
	MNSGA-DO		$\blacktriangle\blacktriangle\nabla - -$		$- - \blacktriangle - \nabla - \blacktriangle\nabla -$		$\blacktriangle\blacktriangle\blacktriangle - \blacktriangle\blacktriangle\nabla$
IGD	NSGA-DO	$\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla$
	MNSGA-DO		$\blacktriangle\blacktriangle - \blacktriangle\blacktriangle$		$\nabla - \blacktriangle\blacktriangle - \blacktriangle\nabla\blacktriangle$		$\nabla\nabla\nabla\nabla\nabla\blacktriangle\nabla$
SP	NSGA-DO	$\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla$
	MNSGA-DO		$\blacktriangle\blacktriangle\blacktriangle\blacktriangle$		$- \blacktriangle\blacktriangle\blacktriangle\blacktriangle$		$- - - - - \blacktriangle -$
HV	NSGA-DO	$\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla\nabla$	$\nabla\nabla\nabla\nabla\nabla\nabla$
	MNSGA-DO		$\blacktriangle\blacktriangle\blacktriangle\blacktriangle$		$\nabla - \blacktriangle\blacktriangle\blacktriangle - \blacktriangle - \blacktriangle$		$\nabla\nabla\nabla\nabla\nabla\blacktriangle\nabla$

O teste estatístico mostrou que para os problemas da família ZDT, o NSGA-DO apresentou, de modo geral, os melhores resultados para o indicador GD, que mede apenas a convergência. Porém, esse comportamento não se mantém diante dos outros conjuntos de problemas. Inclusive, para a família LZ09 observamos exatamente o oposto, os piores resultados estão associados ao NSGA-DO. Quando avaliamos os indicadores IGD, SP e HV, capazes de avaliar a diversidade, observamos que para todas as instâncias de teste o NSGA-DO obtêm os piores resultados.

Por outro lado, o MNSGA-DO apresenta resultados bastante satisfatórios. Em uma comparação com o NSGA-II, das 22 instâncias de teste, considerando o indicador GD, o AG foi significativamente melhor em 10 ($\approx 45,45\%$), não apresentou diferença significativa em 8 ($\approx 36,36\%$) e foi significativamente pior somente em 4 ($\approx 18,18\%$). Para o indicador IGD, o AG foi significativamente melhor em 11 ($\approx 50\%$), não apresentou diferença significativa em 3 ($\approx 13,63\%$) e foi significativamente pior em 8 ($\approx 36,36\%$). Para o indicador SP, o AG foi significativamente melhor em 15 ($\approx 68,18\%$), não apresentou diferença significativa em 7 ($\approx 31,81\%$) e não apresentou resultado pior em nenhuma instância de teste (0%). Por fim, para o indicador HV, o MNSGA-DO foi significativamente melhor em 12 ($\approx 54,54\%$), não apresentou diferença significativa em 3 ($\approx 13,63\%$) e foi significativamente pior em 6 ($\approx 27,27\%$).

5.2 Problema do Próximo Release (*Next Release Problem*)

No modelo de desenvolvimento de software iterativo e incremental, versões funcionais do sistema (ou *releases*) são entregues aos clientes ao longo de ciclos ou iterações. Nessa forma de desenvolvimento, os projetistas devem definir o conjunto de requisitos que devem ser implementados no próximo ciclo, considerando geralmente as preferências dos clientes e as restrições do projeto. É desejável que o conjunto de requisitos a serem implementados agregue satisfação aos clientes, a um custo de implementação relativamente aceitável.

A tarefa de selecionar quais funcionalidades implementar até uma próxima entrega pode-se tornar demasiadamente complexa, pois além de considerar as opiniões dos clientes e as restrições do projeto (estas geralmente orçamentárias), os requisitos tendem apresentar ordens de precedência, dependência de outros requisitos e até exigir implementação simultânea. Nesse cenário, o Problema do Próximo Release (NRP), do inglês *Next Release Problem*, consiste em selecionar requisitos em meio a objetivos concorrentes e potencialmente conflitantes, considerando um conjunto de restrições pré-definidas.

A Figura 5.5 ilustra uma instância do NRP com sete clientes ($C1, \dots, C7$) e oito requisitos ($r1, \dots, r8$) e seus respectivos custos de implementação ($cost1, \dots, cost8$). Nesse exemplo, as setas indicam as dependências dos requisitos enquanto as linhas tracejadas indicam quais requisitos são de interesse de quais clientes. Por exemplo, o cliente $C5$ solicita a implementação dos requisitos $r5$ e $r8$, no entanto, a implementação do requisito $r5$ é dependente da implementação do requisito $r4$ que, por sua vez, é dependente da implementação do requisito $r3$.

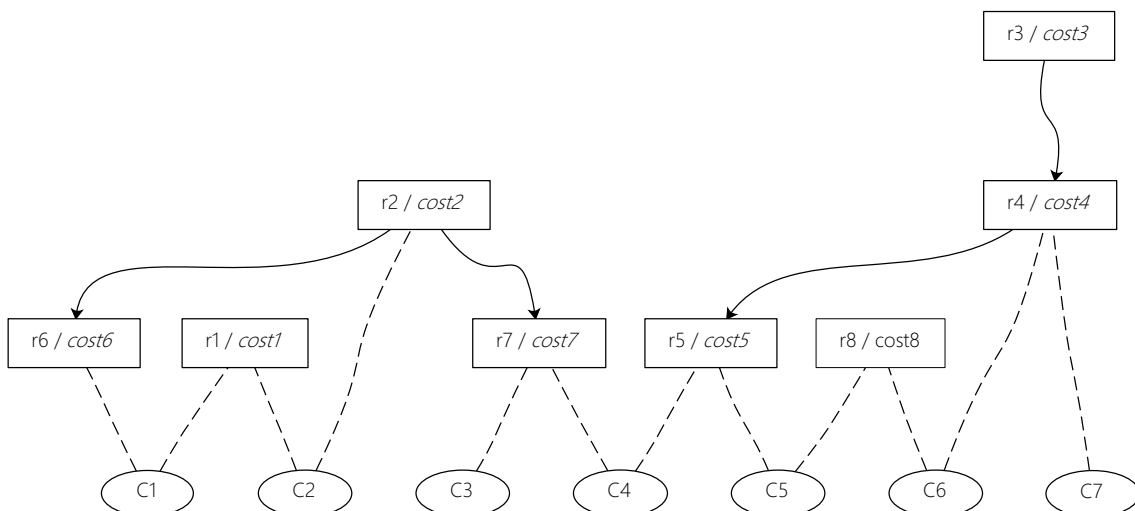


Figura 5.5: Dependências dos requisitos e preferências dos clientes. Adaptado de (Xuan et al., 2012).

Uma das estratégias de resolução de NRPs é a aplicação de algoritmos genéticos capazes de combinar requisitos considerando, por exemplo, maximizar a satisfação dos clientes e minimizar os custos, além de respeitar as dependências.

5.2.1 Modelagem do NRP

Suponha que uma empresa ou organização desenvolvedora de determinado software atenda a um conjunto de clientes denotados por:

$$C = \{c_1, c_2, \dots, c_m\} \quad (5.1)$$

cujos requisitos devem ser considerados em uma próxima entrega.

O conjunto com todos os requisitos a serem implementados é denotado por:

$$R = \{r_1, r_2, \dots, r_n\} \quad (5.2)$$

sendo cada elemento desse conjunto independente.

Cada requisito $r_i (1 \leq i \leq n)$ a ser implementado possui um custo de implementação que representa os recursos a serem alocados em seu desenvolvimento. Esses custos são representados em forma de um vetor denotado por:

$$Cost = \{cost_1, cost_2, \dots, cost_n\} \quad (5.3)$$

Cada cliente $c_j (1 \leq j \leq m)$ possui um grau de importância para a empresa (ou organização). Essa importância pode ser representada em forma de um vetor denotado por:

$$Weight = \{w_1, w_2, \dots, w_m\} \quad (5.4)$$

onde $w_j \in [0, 1]$ e $\sum_{j=1}^m w_j = 1$.

O nível de satisfação de um determinado cliente depende dos requisitos de seu interesse que serão implementados em uma próxima versão do software. O peso que um cliente c_j atribui a um requisito r_i é denotado por:

$$value(r_i, c_j) \quad (5.5)$$

sendo $value(r_i, c_j) > 0$ se o cliente j tem interesse pelo requisito i e 0 caso contrário.

Com base na descrição acima, cada requisito r_i recebe uma pontuação de acordo com a equação:

$$score_i = \sum_{j=1}^m w_j * value(r_i, c_j) \quad (5.6)$$

O vetor de decisão $\vec{x} = \{x_1, x_2, \dots, x_n\}$ determina os requisitos que devem ser implementados na próxima versão do software, onde x_i é 1 se o requisito r_i for selecionado e 0 caso contrário.

Tratamos como objetivos conflitantes o “custo de implementação dos requisitos” e o “grau de satisfação dos clientes”. Assim, com o processo de otimização, o tomador de decisão terá à disposição uma variedade de arranjos de requisitos com diferentes graus de compensação aos objetivos conflitantes considerados. A função de maximização, que representa o grau de satisfação dos clientes ou agregação de valor para a empresa é definida por:

$$\text{Maximize } \sum_{i=1}^n \text{score}_i * x_i \quad (5.7)$$

A função de minimização, que representa os custos associados aos requisitos a serem implementados é definida por:

$$\text{Minimize } \sum_{i=1}^n \text{cost}_i * x_i \quad (5.8)$$

Para que o segundo objetivo seja convertido em um problema de maximização, o somatório dos custos é multiplicado por -1. As funções objetivos f_1 e f_2 são então representadas da seguinte forma:

$$\begin{cases} \text{Maximize } f_1(\vec{x}) = \sum_{i=1}^n \text{score}_i * x_i \\ \text{Maximize } f_2(\vec{x}) = - \sum_{i=1}^n \text{cost}_i * x_i \end{cases} \quad (5.9)$$

5.2.2 Experimento

Como operadores de cruzamento e mutação, foram utilizados o *Single Point Crossover* e o *Bit Flip Mutation*, com probabilidade de cruzamento $p_c = 0.1$ e probabilidade de mutação $p_m = 1/n$, sendo n o número de requisitos. O operador de seleção foi o *Binary Tournament Selection*. O tamanho da população foi definido para 100 e a quantidade de gerações 250. O experimento considerou uma única execução independente para cada algoritmo estudado. O problema aplicado consiste em uma instância de teste NRP de larga escala, com 536 requisitos sem dependências e 3502 clientes.

Neste experimento, cada solução é codificada como uma string binária x , de comprimento n , que representa o número máximo de requisitos, em que $X_i = 1$ indica que o requisito i está incluído na próxima versão do software e $X_i = 0$ indica que não está.

Considerando o NRP como um MOOP, sua *PF* ótima deve cobrir o máximo número de situações diferentes, além de fornecer uma boa distribuição de soluções, o que no mundo real significa fornecer ao Engenheiro de Software um conjunto de opções ótimas contendo um grande número de configurações diferentes (Durillo et al., 2009).

A Figura 5.6 apresenta os resultados obtidos por cada um dos AGs, em que a escala x representa a “Satisfação dos Clientes”, que deve ser maximizada, e a escala y , representa os “Custos” associados a cada valor de satisfação, que deve ser minimizado.

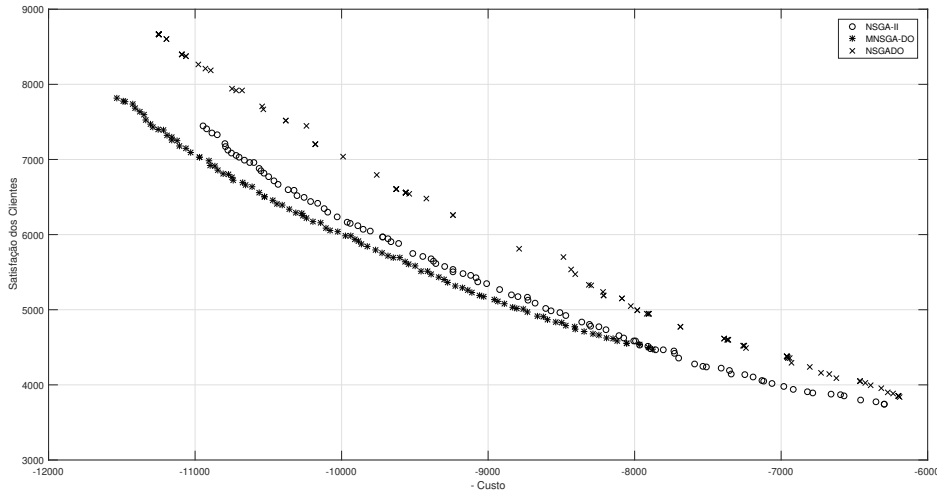


Figura 5.6: PFs^* encontradas pelo NSGA-II, MNSGA-DO e NSGA-DO para uma instância de teste NRP de larga escala.

A análise visual dos resultados nos permite concluir que o NSGA-DO original obteve um melhor desempenho, sendo suas soluções de menor custo e maior satisfação para os clientes. No entanto, nota-se que o algoritmo apresenta menos combinações de requisitos, e sua PF^* apresenta pequenos *gaps* e aglomera algumas soluções. No que se refere ao MNSGA-DO e ao NSGA-II, o primeiro apresenta mais soluções ótimas em relação ao objetivo “satisfação dos clientes”, enquanto o segundo apresenta mais soluções ótimas em relação ao objetivo “reduzir os custos de implementação”, com a PF^* do NSGA-II cobrindo uma maior região do espaço objetivo.

O resultado deste experimento favorece muito o NSGA-DO e, por o NRP ser um MOOP discreto, vai de encontro aos achados obtidos em estudos publicados pelo autor. No entanto, deve-se observar que o experimento considerou apenas uma única execução e realizou apenas uma análise visual das Fronteiras de Pareto, não medindo sua qualidade por meio de indicadores de qualidade, não avaliando a quantidade de soluções ótimas obtidas e nem realizando análises estatísticas.

5.3 Avaliação do Desempenho do *Dynamic* MNSGA-DO (DynamicMNSGA-DO)

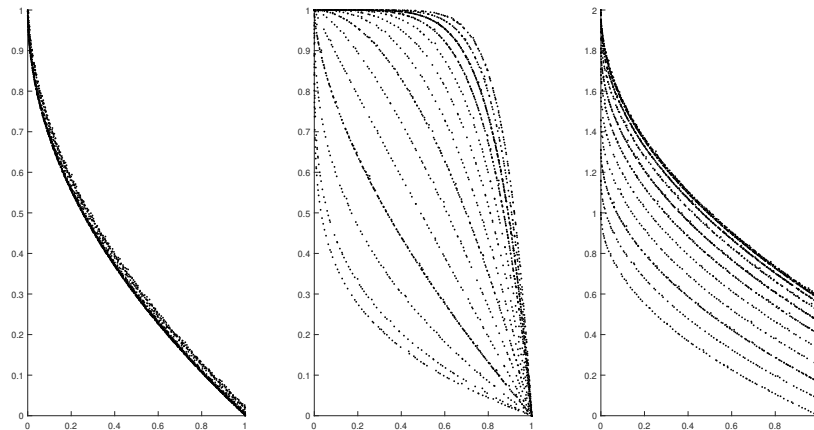
Para avaliar o desempenho da versão dinâmica do MNSGA-DO, intitulado DynamicMNSGA-DO, foi utilizado os problemas bi-objetivos dinâmicos do *benchmark* FDA, proposto em Farina et al. (2004), o qual contém cinco DMOPs (FDA1-5) que são uma extensão direta dos *benchmarks* ZDT e DTLZ. Segundo os autores, esse conjunto de testes permite investigar diferentes obstáculos presentes em problemas do mundo real, tais como: não convexidade, descontinuidade, problemas multimodais e enganosos, entre outros. A Tabela 5.8 descreve brevemente as características de cada teste desse conjunto.

Tabela 5.8: Características das funções de teste FDA

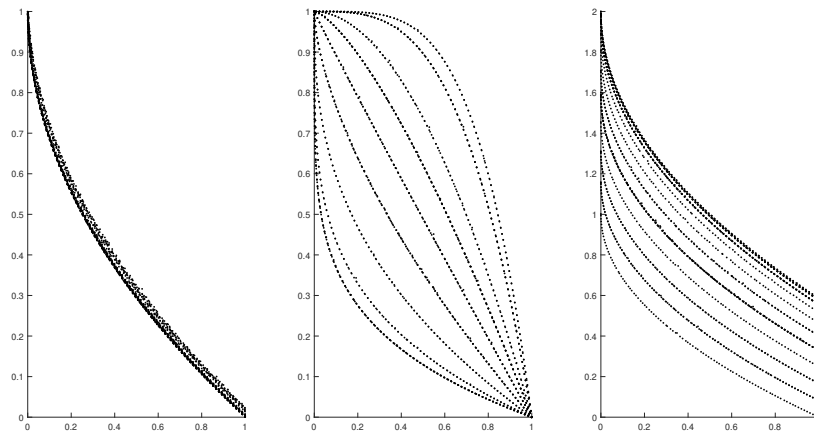
Nome	Tipo	POF	Qtd. Objetivos
FDA1	I	POF Convexa	2
FDA2	III	POFs convexas a não convexas	2
FDA3	II	POFs convexas	2
FDA4	I	POFs não convexas	3
FDA5	II	POFs não convexas	3

Em FDA1, a tarefa de um MOEA dinâmico é encontrar a mesma Fronteira Pareto-Ótima toda vez que houver uma mudança no espaço de decisão. Em FDA2, a PF_t^* oscila de uma forma convexa para não convexa devido a mudança em uma das funções objetivo, enquanto o valor das variáveis de decisão das soluções Pareto-Ótimas permanece inalterado. Em FDA3, a tarefa do MOEA é a de encontrar um conjunto amplamente distribuído de soluções toda vez que houver uma mudança no espaço de decisão. Em FDA 4, a tarefa de um DMOA é a de encontrar a mesma superfície esférica, de raio 1, sempre que houver uma mudança no espaço de decisão. Por fim, em FDA5, a tarefa é encontrar uma boa distribuição na PF_t^* toda vez que houver uma mudança na densidade das suas soluções, devido a alterações feitas no espaço de decisão.

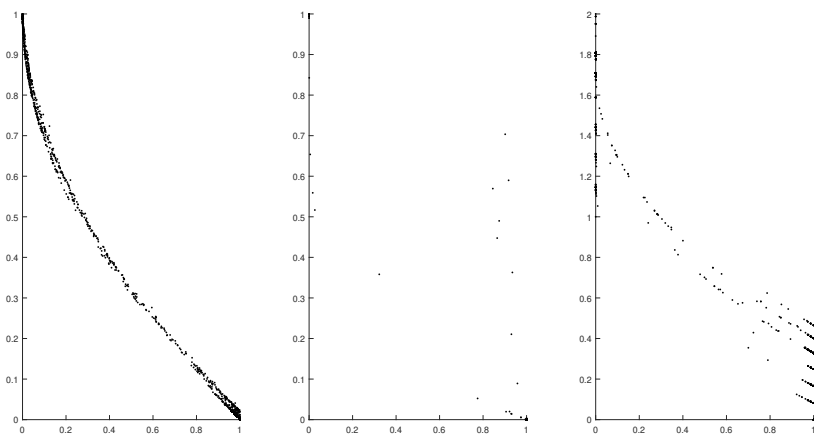
Para a experimentação, novamente utilizamos o framework jMetal que em suas últimas atualizações implementa uma forma de simular ambientes dinâmicos. Conforme o exemplo disponível, após os DMOAs completarem um ciclo de iterações, definido em 250 gerações, ocorre uma mudança após τT novas gerações. Definimos a frequência de mudança em cinco ($\tau T = 5$) e a severidade da mudança em 10 ($nT = 10$), tal como sugere o estudo de Farina et al. (2004). O indicador de qualidade utilizado na detecção de mudanças foi o IGD com uma sensibilidade de 0.055. A Figura 5.7 apresenta as fronteiras encontradas pelos AGs dinâmicos NSGA-DO, NSGA-II e MNSGA-DO para os 24 primeiros instantes de tempo. Na Figura 5.8 podemos observar os vetores de decisão referente às soluções ótimas encontradas.



(a) Algoritmo *Dynamic NSGA-II*

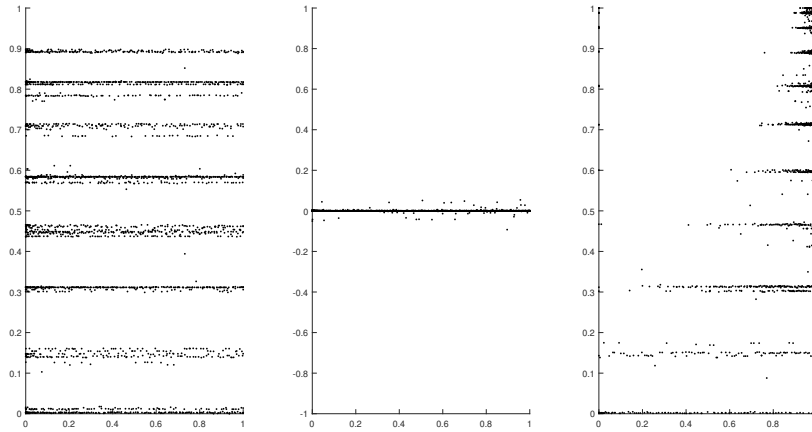


(b) Algoritmo *Dynamic MNSGA-DO*.

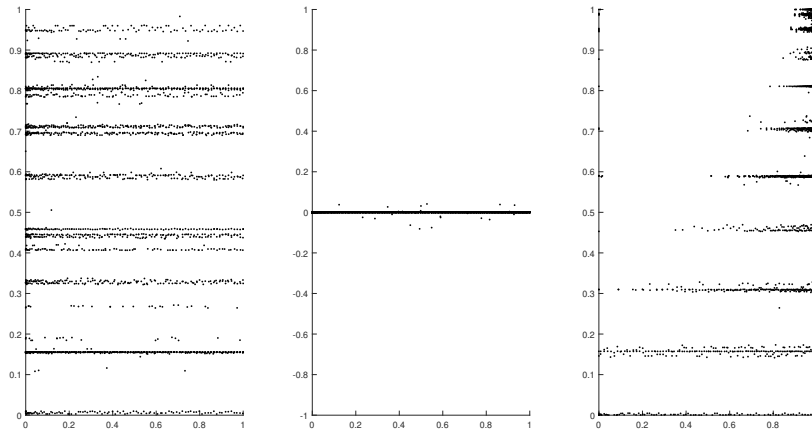


(c) Algoritmo *Dynamic NSGA-DO*

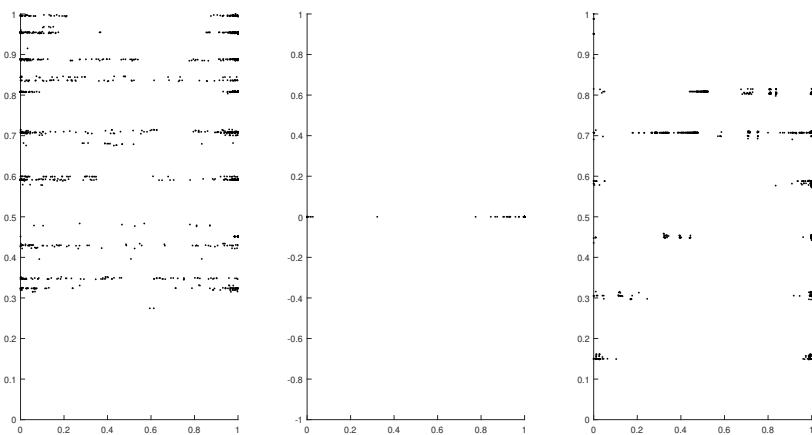
Figura 5.7: PF^* encontradas pelas versões dinâmicas dos AGs NSGA-II (a), MNSGA-DO (b) e NSGA-DO (c) para as instâncias de teste FDA{1-3}, consideramos uma única execução.



(a) Algoritmo *Dynamic NSGA-II*



(b) Algoritmo *Dynamic MNSGA-DO*.



(c) Algoritmo *Dynamic NSGA-DO*

Figura 5.8: PS^* encontradas pelas versões dinâmicas dos AGs NSGA-II (a), MNSGA-DO (b) e NSGA-DO (c) para as instâncias de teste FDA{1-3}, consideramos uma única execução.

Conforme observamos nas imagens, o MNSGA-DO e o NSGA-II conseguiram reagir de forma satisfatória às mudanças de ambiente, enquanto que o NSGA-DO apresentou um comportamento semelhante ao observado na otimização estática. A cada mudança de cenário, o Hipervolume foi calculado sobre a PF_t^* resultante, com isso pôde-se aplicar o teste Ganha-Perde descrito em Helbig and Engelbrecht (2013a). Esse teste consiste em contar a quantidade de vezes que determinado algoritmo obteve um melhor desempenho em comparação a outro, considerando n instantes de tempo. Os resultados desse teste estão apresentados na Tabela 5.9, em que mostra a quantidade de vezes em que o AG da linha ganhou em relação ao AG da coluna.

Tabela 5.9: Resultado Ganha-Perde pos 24 primeiros instantes de tempo

	FDA1		FDA2		FDA3	
	MNSGA-DO	NSGA-II	MNSGA-DO	NSGA-II	MNSGA-DO	NSGA-II
NSGA-DO	0	0	0	1	0	2
MNSGA-DO		14		24		24

A Tabela 5.10 apresenta os mesmos resultados de forma detalhada, mostrando os valores de hipervolume obtidos por cada algoritmo em cada instante de tempo de cada DMOP. Como podemos observar, o *Dynamic* MNSGA-DO obteve um melhor desempenho em todos os ambientes dos problemas FDA 2 e FDA 3, em que a Fronteira Pareto-Ótima se modifica ao longo do tempo. Ainda, para a maioria dos cenários do FDA1, o *Dynamic* MNSGA-DO supera os outros dois algoritmos. Observamos também, que em algumas raras exceções do problema FDA 1, o *Dynamic* NSGA-DO supera o *Dynamic* NSGA-II.

Tabela 5.10: Hipervolume medido para os 24 primeiros instantes de tempo.

	NSGA-II			MNSGA-DO			NSGA-DO		
	FDA1	FDA2	FDA3	FDA1	FDA2	FDA3	FDA1	FDA2	FDA3
t1	6,52E-01	8,29E-01	6,60E-01	6,53E-01	8,30E-01	6,62E-01	4,87E-01	1,02E-02	1,79E-04
t2	6,58E-01	7,96E-01	5,53E-01	6,30E-01	8,30E-01	5,55E-01	6,20E-01	6,18E-03	6,97E-02
t3	6,59E-01	7,12E-01	4,54E-01	6,57E-01	8,30E-01	5,56E-01	4,07E-01	5,87E-01	2,12E-01
t4	6,31E-01	6,06E-01	4,54E-01	6,60E-01	8,30E-01	4,55E-01	6,22E-01	4,37E-03	1,62E-01
t5	6,57E-01	6,06E-01	3,64E-01	6,61E-01	7,97E-01	4,56E-01	1,60E-02	1,19E-03	2,43E-01
t6	6,31E-01	6,06E-01	2,88E-01	6,61E-01	7,97E-01	3,66E-01	6,01E-01	2,22E-01	1,92E-01
t7	6,58E-01	5,01E-01	2,27E-01	6,29E-01	7,14E-01	2,89E-01	4,00E-01	1,82E-06	2,58E-01
t8	6,59E-01	4,10E-01	2,28E-01	6,55E-01	7,14E-01	2,90E-01	2,15E-01	4,35E-01	1,06E-01
t9	6,34E-01	3,35E-01	1,80E-01	6,35E-01	6,07E-01	2,90E-01	2,57E-01	1,01E-02	2,66E-05
t10	6,55E-01	3,35E-01	1,81E-01	6,55E-01	6,07E-01	2,90E-01	6,18E-01	3,05E-07	2,02E-01
t11	6,58E-01	2,76E-01	1,80E-01	6,60E-01	6,07E-01	2,27E-01	3,67E-01	3,05E-07	5,98E-02
t12	6,59E-01	2,29E-01	1,46E-01	6,61E-01	5,03E-01	2,29E-01	1,35E-01	3,04E-05	9,40E-02
t13	6,37E-01	1,92E-01	1,46E-01	6,35E-01	5,03E-01	1,81E-01	6,27E-01	5,08E-06	6,13E-05
t14	6,55E-01	1,63E-01	1,23E-01	6,57E-01	5,03E-01	1,46E-01	3,13E-01	3,82E-02	0,00E+00
t15	6,58E-01	1,62E-01	1,23E-01	6,36E-01	4,12E-01	1,24E-01	3,10E-01	2,67E-07	0,00E+00
t16	6,59E-01	1,62E-01	1,23E-01	6,55E-01	4,11E-01	1,24E-01	3,57E-02	2,67E-07	0,00E+00
t17	6,60E-01	1,39E-01	1,23E-01	6,59E-01	4,11E-01	1,24E-01	1,84E-03	5,50E-02	0,00E+00
t18	6,34E-01	1,39E-01	1,10E-01	6,60E-01	4,12E-01	1,24E-01	6,18E-01	2,67E-07	0,00E+00
t19	6,54E-01	1,39E-01	1,10E-01	6,42E-01	3,37E-01	1,11E-01	3,19E-01	2,67E-07	0,00E+00
t20	6,39E-01	1,39E-01	1,10E-01	6,57E-01	3,36E-01	1,11E-01	1,52E-02	2,91E-02	0,00E+00
t21	6,55E-01	1,39E-01	1,10E-01	6,60E-01	2,30E-01	1,11E-01	5,53E-01	7,46E-02	0,00E+00
t22	6,57E-01	1,20E-01	1,05E-01	6,61E-01	2,30E-01	1,11E-01	2,46E-01	6,68E-02	0,00E+00
t23	6,59E-01	1,20E-01	1,06E-01	6,42E-01	2,30E-01	1,06E-01	2,78E-02	6,43E-02	0,00E+00
t24	6,47E-01	1,04E-01	1,06E-01	6,61E-01	1,93E-01	1,07E-01	1,19E-04	3,39E-02	0,00E+00

Ao observar o gráfico de BoxPlot da Figura 5.9 podemos chegar a conclusões semelhantes. Para FDA1, os resultados associados ao *Dynamic* NSGA-II e *Dynamic* MNSGA-DO são muito semelhantes, sendo estes superiores aos resultados associados ao *Dynamic* NSGA-DO, o qual apresenta piores resultados com maiores variações. Em FDA2, somente os *outliers* do *Dynamic* NSGA-DO possuem valores semelhantes aos dos outros algoritmos, curiosamente, ao retornarmos a Tabela 5.10, notamos que um desses valores discrepantes foi considerado o segundo melhor resultado. No geral, considerando os três DMOPs, o *Dynamic* MNSGA-DO foi superior ao *Dynamic* NSGA-II.

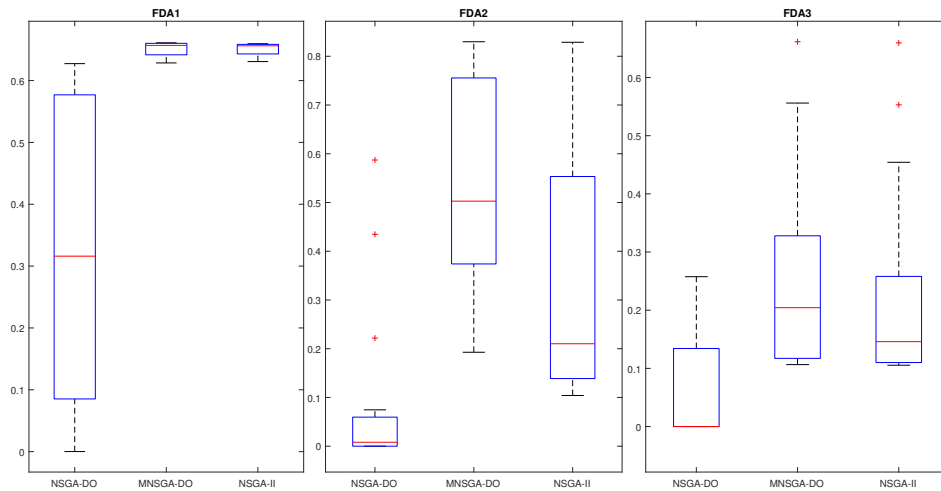


Figura 5.9: Valores de Hipervolume para os 24 primeiros instantes de tempo

Apesar dos bons resultados do MNSGA-DO, uma informação relevante é que o mesmo detecta menos mudanças quando em comparação com o NSGA-II. Isso acontece por conta do mecanismo utilizado na detecção de mudança, o qual se baseia em indicadores de qualidade (perceba que duas fronteiras diferentes podem retornar valores de qualidade muito próximos). Outra informação relevante é que, conforme os experimentos estáticos, o MNSGA-DO necessita de mais gerações para melhorar sua população e assim obter um resultado superior ao do NSGA-II. Isso significa que existe um determinado intervalo de tempo mínimo em que o NSGA-II certamente supera o MNSGA-DO, o que, em um contexto dinâmico, em um ambiente de rápidas mudanças, pode ser decisivo.

Também é interessante destacar a possibilidade de viés, visto que o experimento considerou apenas uma execução independente, além de utilizar uma métrica estática para medir a qualidade. A forma com a qual foi simulada a dinamicidade também interferiu positivamente nos resultados, observe que antes de uma mudança de ambiente, cada DMOA pode completar um ciclo de gerações.

5.4 Considerações Sobre os Experimentos

A análise dos resultados mostra que o algoritmo proposto consegue bons resultados diante de problemas bi-objetivos com diferentes características (fronteira linear, convexa, côncava, mista de convexa e côncava, multimodal e não uniforme). Mesmo para problemas com fronteiras descontínuas, em que pontos ideais são criados entre as discontinuidades, os resultados são competitivos. No entanto, como este é um AG em desenvolvimento, e com o intuito de identificar novas melhorias, é interessante algumas discussões.

Apesar do grande potencial do MNSGA-DO, algumas falhas, inerentes à sua definição, não foram corrigidas:

5.4.1 Evolução do Indicador de Hipervolume

Apesar de o MNSGA-DO encontrar melhores soluções para a grande maioria dos problemas considerados neste estudo, o mesmo necessita de mais gerações para melhorar a qualidade de sua população quando em comparação com o NSGA-II. A Figura 5.10 evidencia essa dificuldade ao apresentar a evolução do Hipervolume dos AGs em algumas instâncias de teste ¹.

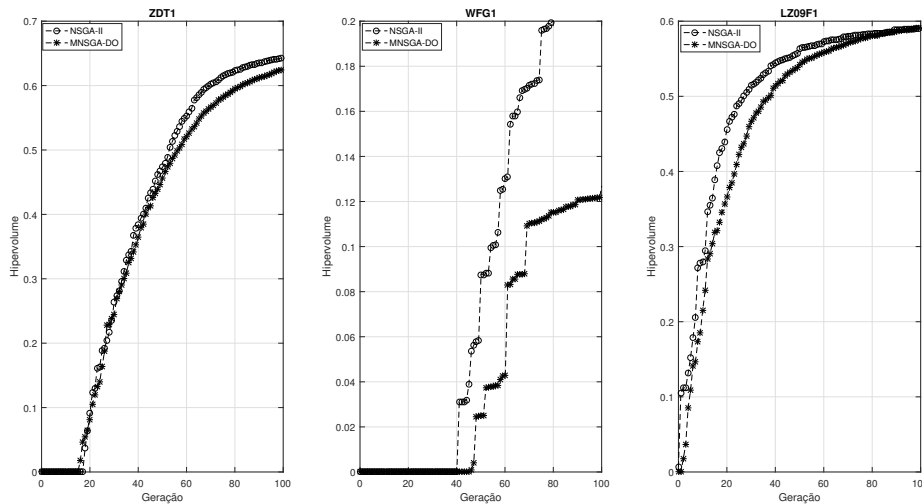


Figura 5.10: Evolução do Hipervolume para os AGs NSGA-II e MNSGA-DO.

Como podemos observar, mais rapidamente o NSGA-II atinge bons resultados de Hipervolume, sendo mais adiante superado pelo MNSGA-DO.

Isso acontece porque, quando em situações em que poucas soluções são selecionadas pelo algoritmo de preservação da diversidade do MNSGA-DO (geralmente nas

¹Desconsideramos apresentar os valores dos indicadores do NSGA-DO, visto que estes são muito ruins e alteram a escala dos gráficos, dificultando uma boa análise visual para os outros AGs.

primeiras gerações), poucos pontos ideais são criados. Sendo poucos pontos ideais, estes não conseguem uma boa representação da fronteira dominante. Como consequência, por não conseguir realizar uma amostragem satisfatória, o algoritmo não garante que as melhores soluções sejam escolhidas.

Explicando de uma outra forma, quando se trata de selecionar poucos indivíduos, o que ocorre geralmente nas primeiras gerações, qualquer conjunto de soluções que seja selecionado não será representativo da fronteira dominante. Em situações como essa, convém utilizar uma estratégia mais próxima a do NSGA-DO, em que a quantidade de pontos ideais é proporcional ao tamanho do *front* dominante (criando uma amostragem satisfatória) e as soluções mais próximas a qualquer destes pontos são selecionadas, com a garantia de que um ponto não selecione mais indivíduos do que outro. Ou seja, como não é possível contribuir em relação à diversidade, que a contribuição seja em relação à convergência.

Essa observação é interessante em um contexto de DMOO, especificamente na resolução de DMOPs com altas taxas de mudança, em que os DMOAs não dispõem de muitas gerações para obterem uma boa qualidade de soluções ótimas (PF_t^*). Nesse sentido, convém adicionar um mecanismo de aceleração da convergência, como por exemplo, paralelismo.

5.4.2 Criação de pontos ideais entre descontinuidades da PF

Para os problemas em que a fronteira ideal é descontínua, como o ZDT3 e o WFG2, pontos ideais são criados entre as descontinuidades, observe o gráfico a esquerda da Figura 5.11. O custo para resolver esta falha seria prejudicial em tempo, além de adicionar complexidade ao algoritmo e não ser preciso, pois qualquer solução aplicada teria dificuldades em distinguir um *gap* que precisa ser corrigido de uma descontinuidade que precisa ser ignorada.

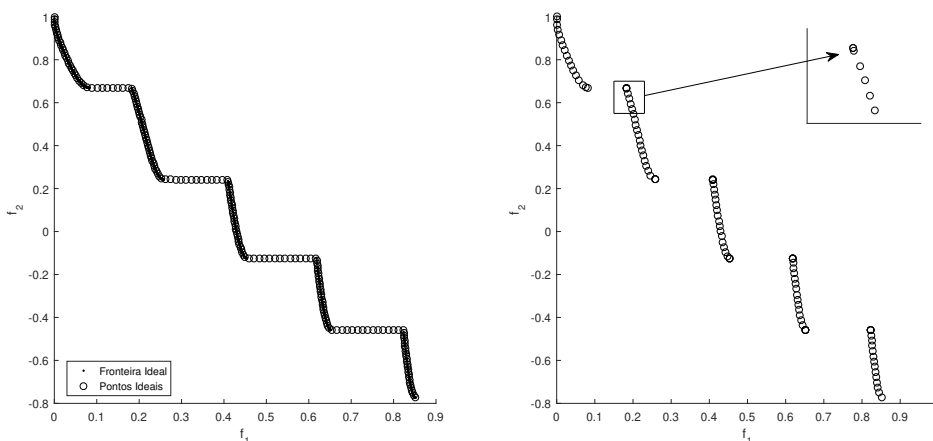


Figura 5.11: Pontos ideais criados entre as descontinuidades do problema ZDT3.

Para resolver esse problema foram desenvolvidas duas soluções baseadas no espaçamento ideal E definido pela Equação 4.1. As distâncias superiores à n vezes indicavam descontinuidades (com n sendo um valor arbitrário). As soluções desenvolvidas melhoraram a qualidade das soluções encontradas para o ZDT3 e o WFG2 e não alteraram a qualidade das soluções encontradas para os demais problemas, com exceção do WFG1. Para esse problema, ambas as soluções tratam os *gaps* como descontinuidades e como consequência, grande parte da região ótima não é povoada. Adaptar estas soluções para o WFG1 seria fácil, porém os tornam soluções especialistas.

Assim, é compreensível que o MNSGA-DO não apresente bons resultados em problemas cujas fronteiras sejam descontínuas, ou mesmo apresentando bons resultados, como aconteceu nos experimentos envolvendo o ZDT3 e WFG2, que os extremos de uma região povoada tenham uma maior densidade de soluções (observe o gráfico a direita da Figura 5.11).

5.4.3 Desempenho Diante de Problemas com Muitos Objetivos

A criação dos pontos ideais utiliza a fronteira menos dominada, se esta for ruim, os pontos ideais também serão. Mesmo o *Fast Non Dominated Sort* alocando para o primeiro *front* os melhores indivíduos, nas primeiras gerações estes ainda não são boas soluções, como consequência, os pontos ideais também não são.

Para dois objetivos, isso não é um problema, visto que, o algoritmo de preservação da diversidade inicialmente é o responsável pela seleção de algumas poucas soluções. Com isso, a convergência inicial é de responsabilidade maior do algoritmo de ordenação não dominada (*Fast Non-Dominated Sort*). Assim, quando os algoritmos de preservação da diversidade do NSGA-DO e MNSGA-DO tem maior influência na seleção, seus pontos ideais já são bons, pois já houve muitas gerações.

Por outro lado, diante de problemas com muitos objetivos, muito rapidamente seus respectivos algoritmos passam a selecionar muitas ou até todas as soluções. Isso porque a maioria dos indivíduos não são dominados uns em relação aos outros (Elarbi et al., 2017), estando estes em um mesmo *front*. Uma possibilidade de correção deste problema é utilizar estratégias de decomposição, que divide um MOOP de muitos objetivos em subproblemas de objetivo único, como o famoso MOEA/D (Zhang and Li (2007)).

A observação deste cenário nos permite repensar sobre a Figura 2.6 e 2.8 (b), pois nem sempre os pontos ideais selecionam indivíduos de outros *fronts* mas também do mesmo *front* F_1 , como mostra a Figura 5.12, porém, isso não deve ser visto como uma falha.

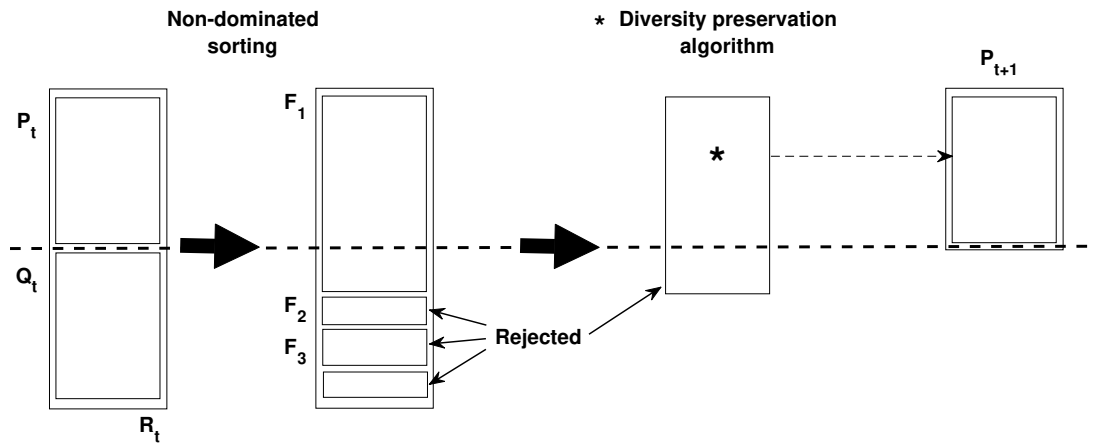


Figura 5.12: Situação em que o algoritmo de preservação da diversidade dos AGs variantes do NSGA-II passa a selecionar todos os indivíduos. Adaptado de Deb et al. (2002).

5.5 Respondendo as Questões de Pesquisa Q2 e Q3

Em Pimenta and de Arruda Camargo (2015), foi apresentado o NSGA-DO, um AG variante do NSGA-II, capaz de criar pontos ideais igualmente espaçados, utilizando-se da fronteira dominante, e assim orientar as soluções de forma a convergirem para pontos ideais próximos. Tal algoritmo, de estratégia simples, obteve resultados superiores aos resultados do NSGA-II quando aplicado em MOOPs discretos (Santana and Camargo, 2019), no entanto, o AG obtém resultados insatisfatórios quando aplicado em MOOPs contínuos, com espaço de busca complexo.

Nesse contexto, parte do objetivo desta pesquisa foi o de encontrar e corrigir falhas no processo de otimização do algoritmo, de modo que o mesmo fosse capaz de resolver de forma satisfatória MOOPs contínuos, tanto estáticos quanto dinâmicos. Para investigação de possíveis erros de convergência e erros relacionados à preservação da diversidade, foi construída uma ferramenta utilizando-se do *software* MATLAB, ferramenta esta que possibilitou visualizar todo o processo de otimização sob diferentes perspectivas. Tendo conhecimento das causas desse desempenho insatisfatório, foram aplicadas correções parcialmente embasadas em estratégias descritas na literatura.

Ao conduzir esta pesquisa, foi possível perceber que a principal causa de uma má distribuição das soluções ao longo da PF^* se deve a falta de um controle sobre o poder de seleção dos pontos ideais criados. Ao permitir que alguns destes pontos selecionem bem mais indivíduos que outros, determinadas regiões ótimas vão sendo privilegiadas, desencadeando *gaps* na PF^* , os quais se tornam maiores a cada gera-

ção. Outra percepção foi em relação à quantidade de pontos ideais, quantidade essa não associada à quantidade de indivíduos que se deve solucionar.

Como correção, inicialmente os pontos ideais são criados em proporção a duas vezes a quantidade de soluções que se deve solucionar. Como somente metade destes pontos terão o poder de seleção (pontos ativos), e para manter a uniformidade na distribuição, estes sempre estarão entre pontos sem o poder de seleção (pontos inativos). O processo de seleção segue com os pontos ativos sendo percorridos dos extremos ao centro, e com a solução mais próxima a este adicionada na nova população.

As modificações realizadas no NSGA-DO geraram o variante MNSGA-DO, o qual supera o NSGA-DO e até mesmo o NSGA-II em MOOPs contínuos com diferentes características. Também, tendo como base a estratégia do DNSGA-II-A descrito em Deb et al. (2007), versões dinâmicas dos três AGs foram desenvolvidos e comparados. Diante dos resultados obtidos, pode-se responder às questões de pesquisa Q2 e Q3.

Sobre a questão de pesquisa **Q2** (*Q2. Qual o desempenho do NSGA-DO na resolução de MOOPs contínuos?*), como demonstrado nos experimentos, diante de MOOPs contínuos, com ambientes de busca complexos, o NSGA-DO apresenta um resultado insatisfatório em relação a preservação da diversidade no conjunto de soluções ótimas encontrado. Como explicado, isso acontece porque os pontos ideais próximos às melhores soluções são sempre privilegiados, sendo estas soluções adicionadas na nova população e com os operadores de mutação e cruzamento gerando sempre soluções semelhantes, pertencentes às mesmas regiões.

Apesar desse comportamento, os pontos ideais criados pelo NSGA-DO são capazes de representar de forma adequada o conjunto de soluções dominantes (pertencentes ao *front* $F1$). Nesse sentido, simples correções na estratégia do NSGA-DO permitem o desenvolvimento de uma versão com melhores resultados em um contexto MOO e DMOO com espaço de busca complexo.

O MNSGA-DO (variante estático do NSGA-DO), foi comparado com o NSGA-DO e o NSGA-II considerando os *benchmarks* contínuos ZDT, WFG, LZ09 e DTLZ. Para avaliar a convergência e a diversidade no conjunto final de soluções dos três algoritmos, foram aplicados os indicadores de qualidade GD, IGD, Spread e HV, com seus valores sendo aplicados ao teste estatístico Wilcoxon para verificar a significância das diferenças.

Para todas as instâncias de teste, o MNSGA-DO superou o NSGA-DO, apresentando uma PF^* sem *gaps* e com soluções distribuídas ao longo de toda a região ótima. Em comparação com o NSGA-II, de modo geral, o novo AG obteve melhores resultados, no entanto, sendo mais sensível ao problema de separação de soluções em *fronts* quando em MOOPs com três ou mais objetivos.

Sobre a questão de pesquisa **Q3** (*Q3. É possível adaptar o NSGA-DO na resolução de DMOPs? Como é o seu desempenho?*), a versão dinâmica do NSGA-DO apresenta os mesmos erros de sua versão estática, aglomerando soluções em regiões específicas.

Porém, a versão dinâmica do MNSGA-DO apresentou resultados animadores quando em comparação com a versão dinâmica do NSGA-II.

Para avaliação de desempenho das versões dinâmicas dos AGs, foi utilizado os DMOPs FDA{1-3} em 24 instantes de tempo. Ao fim de cada instante, quando detectado uma mudança, as soluções dominantes dos algoritmos foram avaliadas considerando o indicador híbrido HV. Desconsiderando algumas poucas exceções, os melhores resultados estavam associados ao *Dynamic* MNSGA-DO.

Capítulo 6

Conclusões

Muitos problemas do mundo real envolvem muitos objetivos e são de natureza dinâmica. No entanto, muitas questões desse campo de pesquisa ainda necessitam de respostas, questões estas principalmente relacionadas ao desenvolvimento de algoritmos eficientes, bem como as formas de avaliar seu desempenho. Nesse contexto, essa pesquisa buscou contribuir para a resolução de algumas questões que ecoam nesse campo, assim como buscou desenvolver um AG capaz de encontrar soluções satisfatórias, tanto na resolução de MOOPs quanto na resolução de DMOPs.

Para alcançar tais objetivos, estudos de revisão publicados no campo de pesquisa da DMOO foram analisados. Com isso, pode-se verificar como as diferentes dificuldades evoluíram ao longo dos anos, bem como, se a comunidade científica está engajada na superação das mesmas. Os resultados da revisão da literatura também guiaram parte da experimentação. Com isso, uma estratégia simples pôde ser acoplada aos AGs estudados para torná-los aptos a resolver DMOPs.

O NSGA-DO é um algoritmo de fácil compreensão e implementação e com um custo computacional relativamente baixo. Atualmente, as modificações realizadas em sua estrutura resultaram no *Modified* NSGA-DO (MNSGA-DO), um AG com resultados muito competitivos, quando em comparação com o famoso NSGA-II. Nossos experimentos, que envolveram diferentes MOOPs, com diferentes características, evidenciaram que o novo algoritmo proposto supera o NSGA-DO em praticamente todos os experimentos, assim como supera o NSGA-II em diferentes aspectos.

6.1 Possíveis Trabalhos Futuros

Nesse estudo, buscou-se realizar adaptações no NSGA-DO mantendo sua estrutura, de modo que a compreensão do seu funcionamento, bem como a complexidade de sua implementação, se mantivesse simples. Com os resultados obtidos, pode-se dizer que tal objetivo foi cumprido, porém, melhorias ainda podem ser aplicadas ao AG, tais como: reduzir seu tempo de execução, por meio da implementação

de estruturas de dados mais eficientes em relação a busca; melhorar seu tempo de convergência, ou diminuir a quantidade de gerações que o algoritmo necessita para melhorar seu conjunto de soluções; e substituir o algoritmo de ordenação não dominada por outro que seja mais eficiente quando se trata de problemas com muitos objetivos, ou mesmo, modifica a forma de criação dos pontos ideais.

Referências

- Akbar, M. and Irohara, T. (2020). Nsga-ii variants for solving a social-conscious dual resource-constrained scheduling problem. *Expert Systems with Applications*, 162:113754.
- Azzouz, R., Bechikh, S., and Ben Said, L. (2015). Multi-objective optimization with dynamic constraints and objectives: new challenges for evolutionary algorithms. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 615–622.
- Azzouz, R., Bechikh, S., and Said, L. B. (2017). Dynamic multi-objective optimization using evolutionary algorithms: a survey. In *Recent advances in evolutionary multi-objective optimization*, pages 31–70. Springer.
- Bagnall, A. J., Rayward-Smith, V. J., and Whittle, I. M. (2001). The next release problem. *Information and software technology*, 43(14):883–890.
- Bradstreet, L., Barone, L., While, L., Huband, S., and Hingston, P. (2007). Use of the wfg toolkit and pisa for comparison of moeas. In *2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pages 382–389. IEEE.
- Chen, L., Zhou, C., Dai, G., Zhang, Y., and Hu, R. (2016). Entropy determined hybrid two-stage multi-objective evolutionary algorithm combining locally linear embedding. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 2565–2572. IEEE.
- Constantinou, D. et al. (2011). *Ant colony optimisation algorithms for solving multi-objective power-aware metrics for mobile ad hoc networks*. PhD thesis, University of Pretoria.
- de Queiroz Lafetá, T. F. and de Oliveira, G. M. B. (2020). Applying dynamic evolutionary optimization to the multiobjective knapsack problem. In *Brazilian Conference on Intelligent Systems*, pages 49–63. Springer.
- de Souza Ribeiro, R. (2016). *Comparação De Algoritmos Evolucionários Para Problemas Com Muitos Objetivos*. PhD thesis, Universidade Federal de Minas Gerais.

- Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary computation*, 7(3):205–230.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons.
- Deb, K. (2011). Single and multi-objective dynamic optimization: two tales from an evolutionary perspective. *Indian Institute of Technology*, 2:14.
- Deb, K., Agrawal, R. B., et al. (1995). Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148.
- Deb, K. and Agrawal, S. (1999). A niched-penalty approach for constraint handling in genetic algorithms. In *Artificial Neural Nets and Genetic Algorithms*, pages 235–243. Springer.
- Deb, K. and Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601.
- Deb, K., Karthik, S., et al. (2007). Dynamic multi-objective optimization and decision-making using modified nsga-ii: A case study on hydro-thermal power scheduling. In *International conference on evolutionary multi-criterion optimization*, pages 803–817. Springer.
- Deb, K., Pratap, A., Agarwal, S., and Meyerivian, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.
- Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer.
- del Sagrado, J., del Águila, I. M., and Orellana, F. J. (2010). Ant colony optimization for the next release problem: A comparative study. In *2nd International Symposium on Search Based Software Engineering*, pages 67–76.
- D’Souza, R. G., Sekaran, K. C., and Kandasamy, A. (2010). Improved nsga-ii based on a novel ranking scheme. *arXiv preprint arXiv:1002.4005*.
- Durillo, J. J. and Nebro, A. J. (2011). jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771.
- Durillo, J. J., Zhang, Y., Alba, E., and Nebro, A. J. (2009). A study of the multi-objective next release problem. In *2009 1st International Symposium on Search Based Software Engineering*, pages 49–58. IEEE.

- Elarbi, M., Bechikh, S., Gupta, A., Said, L. B., and Ong, Y.-S. (2017). A new decomposition-based nsga-ii for many-objective optimization. *IEEE transactions on systems, man, and cybernetics: systems*, 48(7):1191–1210.
- Esfe, M. H. and Tilebon, S. M. S. (2020). Statistical and artificial based optimization on thermo-physical properties of an oil based hybrid nanofluid using nsga-ii and rsm. *Physica A: Statistical Mechanics and its Applications*, 537:122126.
- Farina, M., Deb, K., and Amato, P. (2004). Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Transactions on evolutionary computation*, 8(5):425–442.
- Figueiredo, E. M. d. N. (2017). *Uma nova abordagem baseada em enxames de partículas para otimização de muitos objetivos*. PhD thesis, Universidade Federal de Pernambuco, <https://repositorio.ufpe.br/handle/123456789/25629>.
- Gomes, R. D., Alencar, M. S., Fonseca, I. E., and Lima Filho, A. C. (2014). Desafios de redes de sensores sem fio industriais. *Revista de Tecnologia da Informação e Comunicação*, 4(1):16–27.
- Gong, D., Sun, J., and Miao, Z. (2016). A set-based genetic algorithm for interval many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 22(1):47–60.
- Greer, D. and Ruhe, G. (2004). Software release planning: an evolutionary and iterative approach. *Information and software technology*, 46(4):243–253.
- Gu, F., Liu, H.-L., and Tan, K. C. (2012). A multiobjective evolutionary algorithm using dynamic weight design method. *International Journal of Innovative Computing, Information and Control*, 8(5 (B)):3677–3688.
- Helbig, M., Deb, K., and Engelbrecht, A. (2016). Key challenges and future directions of dynamic multi-objective optimisation. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1256–1261. IEEE.
- Helbig, M. and Engelbrecht, A. P. (2013a). Analysing the performance of dynamic multi-objective optimisation algorithms. In *2013 IEEE Congress on Evolutionary Computation*, pages 1531–1539. IEEE.
- Helbig, M. and Engelbrecht, A. P. (2013b). Benchmarks for dynamic multi-objective optimisation. In *2013 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, pages 84–91. IEEE.
- Helbig, M. and Engelbrecht, A. P. (2013c). Challenges of dynamic multi-objective optimisation. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, pages 254–261. IEEE.

- Helbig, M. and Engelbrecht, A. P. (2013d). Performance measures for dynamic multi-objective optimisation algorithms. *Information Sciences*, 250:61–81.
- Helbig, M. and Engelbrecht, A. P. (2014). Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. *Swarm and Evolutionary computation*, 14:31–47.
- Helbig, M. et al. (2012). *Solving dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation*. PhD thesis, University of Pretoria.
- Huband, S., Barone, L., While, L., and Hingston, P. (2005). A scalable multi-objective test problem toolkit. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 280–295. Springer.
- Huband, S., Hingston, P., Barone, L., and While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506.
- Ishibuchi, H., Imada, R., Setoguchi, Y., and Nojima, Y. (2016). Performance comparison of nsga-ii and nsga-iii on various many-objective test problems. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 3045–3052. IEEE.
- Ishibuchi, H., Masuda, H., Tanigaki, Y., and Nojima, Y. (2015). Modified distance calculation in generational distance and inverted generational distance. In *International conference on evolutionary multi-criterion optimization*, pages 110–125. Springer.
- Ishibuchi, H., Matsumoto, T., Masuyama, N., and Nojima, Y. (2020). Many-objective problems are not always difficult for pareto dominance-based evolutionary algorithms. In *ECAI 2020*, pages 291–298. IOS Press.
- Jain, H. and Deb, K. (2013). 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*, 18(4):602–622.
- Jiang, S., Zou, J., Yang, S., and Yao, X. (2022). Evolutionary dynamic multi-objective optimisation: A survey. *ACM Computing Surveys*, 55(4):1–47.
- Jozefowicz, N., Semet, F., and Talbi, E.-G. (2005). Enhancements of nsga ii and its application to the vehicle routing problem with route balancing. In *International Conference on Artificial Evolution (Evolution Artificielle)*, pages 131–142. Springer.
- Kanwar, V. and Kumar, A. (2020). Multiobjective optimization-based dv-hop localization using nsga-ii algorithm for wireless sensor networks. *International Journal of Communication Systems*, 33(11):e4431.

- Keele, S. et al. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Citeseer.
- Kwan, C., Yang, F., and Chang, C. (2007). A differential evolution variant of nsga ii for real world multiobjective optimization. In *Australian Conference on Artificial Life*, pages 345–356. Springer.
- Leung, Y.-W. and Wang, Y. (2003). U-measure: a quality measure for multiobjective programming. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 33(3):337–343.
- Li, H. and Zhang, Q. (2008). Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE transactions on evolutionary computation*, 13(2):284–302.
- Li, K., Deb, K., Zhang, Q., and Kwong, S. (2014). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716.
- Li, K., Deb, K., Zhang, Q., and Kwong, S. (2015). An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. *IEEE Trans. Evol. Comput.*, 19(5):694–716.
- Liu, H.-L., Gu, F., and Zhang, Q. (2013). Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE transactions on evolutionary computation*, 18(3):450–455.
- Martello, S. (1990). Knapsack problems: algorithms and computer implementations. *Wiley-Interscience series in discrete mathematics and optimization*.
- MATLAB (2022). Generate and plot pareto front. [urlhttps://www.mathworks.com/help/optim/ug/generate-and-plot-a-pareto-front.html](https://www.mathworks.com/help/optim/ug/generate-and-plot-a-pareto-front.html).
- Mehnen, J., Wagner, T., and Rudolph, G. (2006). Evolutionary optimization of dynamic multi-objective test functions. In *Proceedings of the Second Italian Workshop on Evolutionary Computation (GSICE2)*.
- Miettinen, K. (2012). *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media.
- Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- Moreno, J., Rodriguez, D., Nebro, A. J., and Lozano, J. A. (2020). Merge nondominated sorting algorithm for many-objective optimization. *IEEE Transactions on Cybernetics*, 51(12):6154–6164.
- Navas, M. M. and Urbaneja, A. J. N. (2013). Metaheurísticas multiobjetivo adaptativas. *Computación y Sistemas*, 17(1):53–62.

- Nebro, A. J. and Durillo, J. J. (2009). On the effect of applying a steady-state selection scheme in the multi-objective genetic algorithm nsga-ii. In *Nature-Inspired Algorithms for Optimisation*, pages 435–456. Springer.
- Nebro, A. J., Durillo, J. J., Coello, C. A. C., Luna, F., and Alba, E. (2008). A study of convergence speed in multi-objective metaheuristics. In *International Conference on Parallel Problem Solving from Nature*, pages 763–772. Springer.
- Nebro, A. J., Durillo, J. J., Garcia-Nieto, J., Coello, C. C., Luna, F., and Alba, E. (2009). Smpso: A new pso-based metaheuristic for multi-objective optimization. In *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, pages 66–73. IEEE.
- Nebro, A. J., Durillo, J. J., Machin, M., Coello, C. A. C., and Dorronsoro, B. (2013). A study of the combination of variation operators in the nsga-ii algorithm. In *Conference of the Spanish Association for Artificial Intelligence*, pages 269–278. Springer.
- Niazi, M. (2015). Do systematic literature reviews outperform informal literature reviews in the software engineering domain? an initial case study. *Arabian Journal for Science and Engineering*, 40(3):845–855.
- Niazi, M., Wilson, D., and Zowghi, D. (2006). Critical success factors for software process improvement implementation: an empirical study. *Software Process: Improvement and Practice*, 11(2):193–211.
- Nnodim, C. T., Arowolo, M. O., Agboola, B. D., Ogundokun, R. O., and Abiodun, M. K. (2021). Future trends in mechatronics. *IAES International Journal of Robotics and Automation*, 10(1):24.
- Parvizi, E. and Rezvani, M. H. (2020). Utilization-aware energy-efficient virtual machine placement in cloud networks using nsga-iii meta-heuristic approach. *Cluster Computing*, pages 1–23.
- Peng, X. and Xu, D. (2012). Intelligent online path planning for uavs in adversarial environments. *International Journal of Advanced Robotic Systems*, 9(1):3.
- Pimenta, A. H. and de Arruda Camargo, H. (2015). Nsga-do: Non-dominated sorting genetic algorithm distance oriented. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE.
- Rangel, E. O., Costa, D. G., and Loula, A. (2019). On redundant coverage maximization in wireless visual sensor networks: Evolutionary algorithms for multi-objective optimization. *Applied Soft Computing*, 82:105578.
- Richter, H. (2013). Dynamic fitness landscape analysis. In *Evolutionary computation for dynamic optimization problems*, pages 269–297. Springer.

- Santana, M. and Camargo, H. (2019). A parallel strategy for generation of fuzzy rule bases in big data problems using the nsga-do. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 48–53.
- Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the first international conference on genetic algorithms and their applications, 1985*. Lawrence Erlbaum Associates. Inc., Publishers.
- Scott, J. (1995). Fault tolerant design using single and multi-criteria genetic algorithms. *Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology*.
- Seada, H. and Deb, K. (2014). U-nsga-iii: A unified evolutionary algorithm for single, multiple, and many-objective optimization. *COIN report*, 2014022.
- Seada, H. and Deb, K. (2015). U-nsga-iii: a unified evolutionary optimization procedure for single, multiple, and many objectives: proof-of-principle results. In *International conference on evolutionary multi-criterion optimization*, pages 34–49. Springer.
- Srinivas, N. and Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248.
- Tantar, E., Tantar, A.-A., and Bouvry, P. (2011). On dynamic multi-objective optimization, classification and performance measures. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2759–2766. IEEE.
- Van Veldhuizen, D. A. (1999). *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Air Force Institute of Technology.
- Van Veldhuizen, D. A. and Lamont, G. B. (1998). Multiobjective evolutionary algorithm research: A history and analysis. Technical report, Citeseer.
- Wahle, J., Annen, O., Schuster, C., Neubert, L., and Schreckenberg, M. (2001). A dynamic route guidance system based on real traffic data. *European journal of operational research*, 131(2):302–308.
- Wang, Y. and Dang, C. (2008). An evolutionary algorithm for dynamic multi-objective optimization. *Applied Mathematics and Computation*, 205(1):6–18.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bull.*, 1(6):80–83.
- Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer.

- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pages 1–10.
- Xie, Y. and Steven, G. (1996). Evolutionary structural optimization for dynamic problems. *Computers & Structures*, 58(6):1067–1073.
- Xu, X., Fu, S., Li, W., Dai, F., Gao, H., and Chang, V. (2020). Multi-objective data placement for workflow management in cloud infrastructure using nsga-ii. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(5):605–615.
- Xuan, J., Jiang, H., Ren, Z., and Luo, Z. (2012). Solving the large scale next release problem with a backbone-based multilevel algorithm. *IEEE Transactions on Software Engineering*, 38(5):1195–1212.
- Zhang, L., Chen, L., Xia, S., Ge, Y., Wang, C., and Feng, H. (2020). Multi-objective optimization for helium-heated reverse water gas shift reactor by using nsga-ii. *International Journal of Heat and Mass Transfer*, 148:119025.
- Zhang, Q. and Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731.
- Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., Tiwari, S., et al. (2008). Multiobjective optimization test instances for the cec 2009 special session and competition. *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report*, 264:1–30.
- Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., and Tsang, E. (2007). Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. In *International conference on evolutionary multi-criterion optimization*, pages 832–846. Springer.
- Zhou, Y. and Zheng, S. (2020). Machine learning-based multi-objective optimisation of an aerogel glazing system using nsga-ii—study of modelling and application in the subtropical climate hong kong. *Journal of Cleaner Production*, 253:119964.
- Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications*, volume 63. Citeseer.
- Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195.
- Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271.

-
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132.

Apêndice A

Classificação da Pesquisa e Metodologia

A.1 Classificação da Pesquisa

Quanto à **abordagem**, tratou-se de uma pesquisa quantitativa, visto que a análise dos fenômenos sucedeu por meio de quantificações, sendo os resultados classificados, estatificados e apresentados em formas estruturadas, como, por exemplo, tabelas e gráficos.

Quanto aos **objetivos**, tratou-se de uma pesquisa descritiva explicativa. Descritiva por objetivar descrever características sobre a otimização dinâmica por meio da coleta de dados quantitativos, por exemplo, quais dos desafios desse campo já foram superados e quais as direções futuras; e explicativa por conectar a teoria com a prática para explicar fenômenos, compreender causas e efeitos, e desenvolver um conhecimento novo, essa é considerada a continuação da pesquisa descritiva.

Quanto aos **procedimentos**, tratou-se de uma pesquisa bibliográfica e experimental. Bibliográfica por considerar como fonte de dados materiais já publicados na obtenção de novas informações, como por exemplo, padrões de comparação de DMOAs; e experimental por aplicar as descobertas teóricas em um experimento controlado, no caso, a combinação da estratégia evolutiva do NSGA-DO com alguma estratégia utilizada por AGs na otimização dinâmica.

A.2 Metodologia

A metodologia adotada nesta pesquisa foi dividida em duas fases parcialmente intercaladas, sendo a primeira relacionada à coleta de dados e a segunda relacionada à condução dos experimentos.

A.2.1 Coleta de Dados

Essa primeira fase consistiu em uma revisão informal da literatura sobre o campo da DMOO, tendo como objetivo responder à questão de pesquisa Q1 além de fortalecer o embasamento teórico para a experimentação. Aqui, foram considerados somente estudos de revisão, visto a ampla quantidade de pesquisas menores publicadas no campo da DMOO, associado a possível impossibilidade de análise destes no período de tempo de um mestrado acadêmico. A Figura A.1 apresenta o fluxograma desta fase.

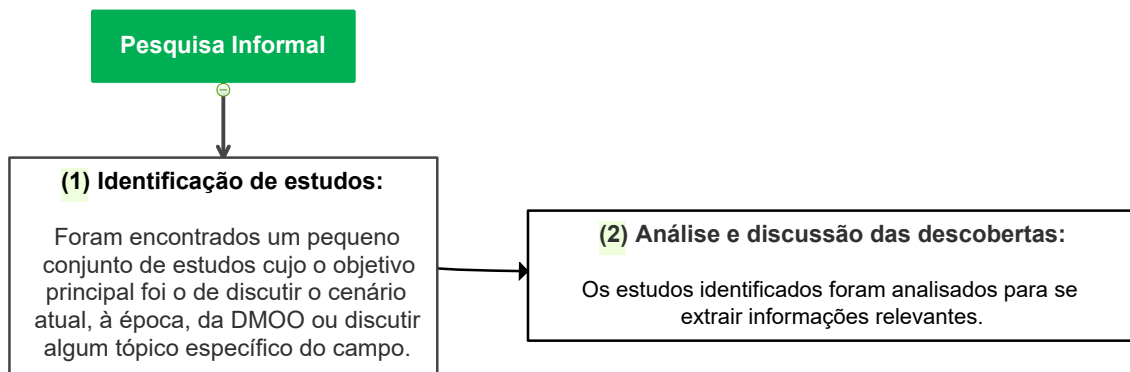


Figura A.1: Fluxograma da fase de coleta de dados.

A.2.2 Experimentos

A segunda fase teve como objetivo a realização de um estudo controlado, parcialmente embasado nas descobertas promovidas pela fase da coleta de dados, para então responder às questões de pesquisa Q2 e Q3.

Inicialmente a estratégia evolutiva do NSGA-DO foi investigada e adaptada para resolver de modo satisfatório diferentes MOOPs contínuos. Esse processo consistiu na aplicação de melhorias seguidas da aplicação de testes de avaliação de desempenho. As etapas desse processo são brevemente descritas a seguir:

Etapa 1: *Identificação de erros de convergência do NSGA-DO:* Inicialmente, a estratégia de evolução do NSGA-DO foi investigada para identificar as razões pelas quais o seu desempenho não é satisfatório na resolução de MOOPs e DMOPs contínuos.

Etapa 2: *Aplicação de melhorias:* Tendo conhecimento das deficiências do NSGA-DO, melhorias foram realizadas com o objetivo de corrigir as falhas de convergência,

bem como, dificuldades relacionadas à preservação da diversidade no conjunto final de soluções.

Etapa 3: *Aplicação de testes de desempenho:* Em paralelo à modificação do NSGA-DO, foram aplicados *benchmarks* e indicadores de qualidade para avaliar o desempenho dos seus variantes.

Etapa 4: *Avaliação por meio de estudos comparativos:* O variante do NSGA-DO foi comparado com sua versão original e o NSGA-II. Posteriormente, foi aplicado o teste *Wilcoxon signed-rank* (Wilcoxon, 1945), com nível de significância de 0.05, para avaliar a significância das diferenças em relação à qualidade das populações encontradas.

Tais etapas foram inicialmente aplicadas em um contexto de otimização estático e posteriormente em um contexto de otimização dinâmico. A Figura A.2 apresenta o fluxograma dessa segunda fase.

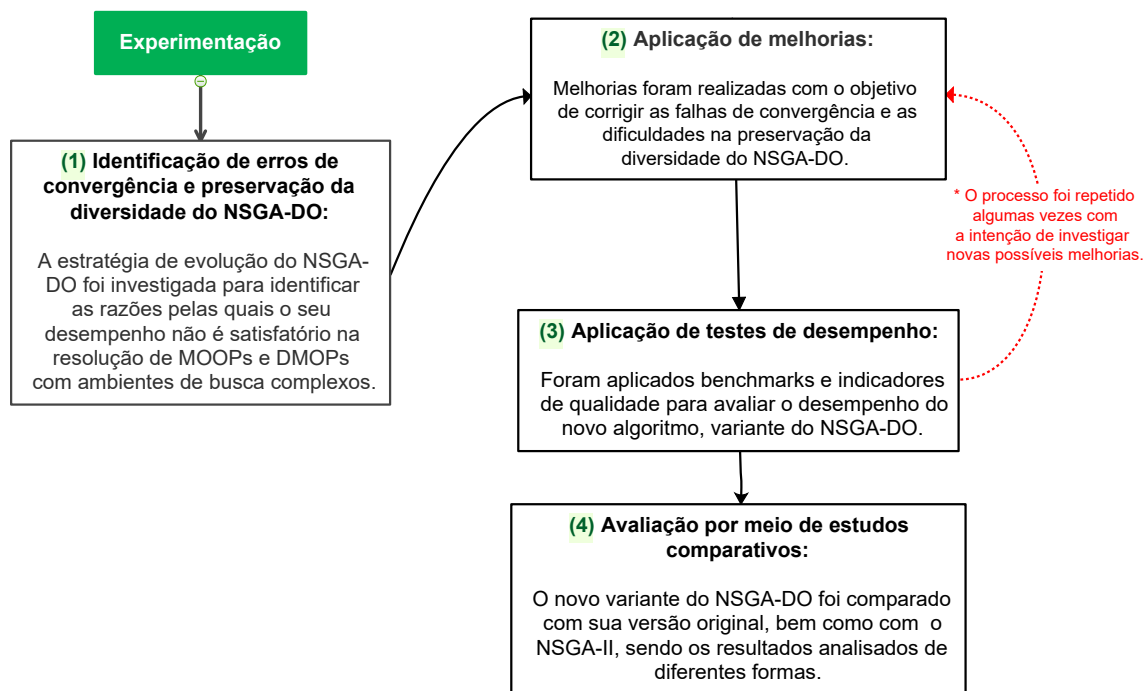


Figura A.2: Fluxograma da fase experimental.

Apêndice B

Revisão Sistemática da Literatura

Para Keele et al. (2007), uma Revisão Sistemática da Literatura (SLR, do inglês *Systematic Literature Review*) é uma forma definida e metódica de identificar, avaliar e analisar estudos primários publicados, a fim de investigar questões de pesquisa específicas. Essa metodologia de busca possibilita descobrir a estrutura e os padrões da pesquisa existente e, assim, identificar lacunas que podem ser preenchidas por pesquisas futuras (Keele et al. (2007), Niazi et al. (2006)).

O objetivo de um SLR é identificar o maior número possível de estudos relevantes. Porém, como destacado por Wohlin (2014), esse objetivo é improvável que aconteça na prática, sobretudo em áreas de estudo mais amplas. Tendo considerado esta questão, uma abordagem que enfrenta este desafio é a combinação de diferentes estratégias, como Procedimento Bola de Neve (*Snowballing*), Busca em Bibliotecas Digitais e Busca Informal.

B.1 Procedimento Bola de Neve (*Snowballing*)

Conforme apresentado por Wohlin (2014), o procedimento de Bola de Neve é representado na Figura B.1 e é descrito brevemente a seguir.

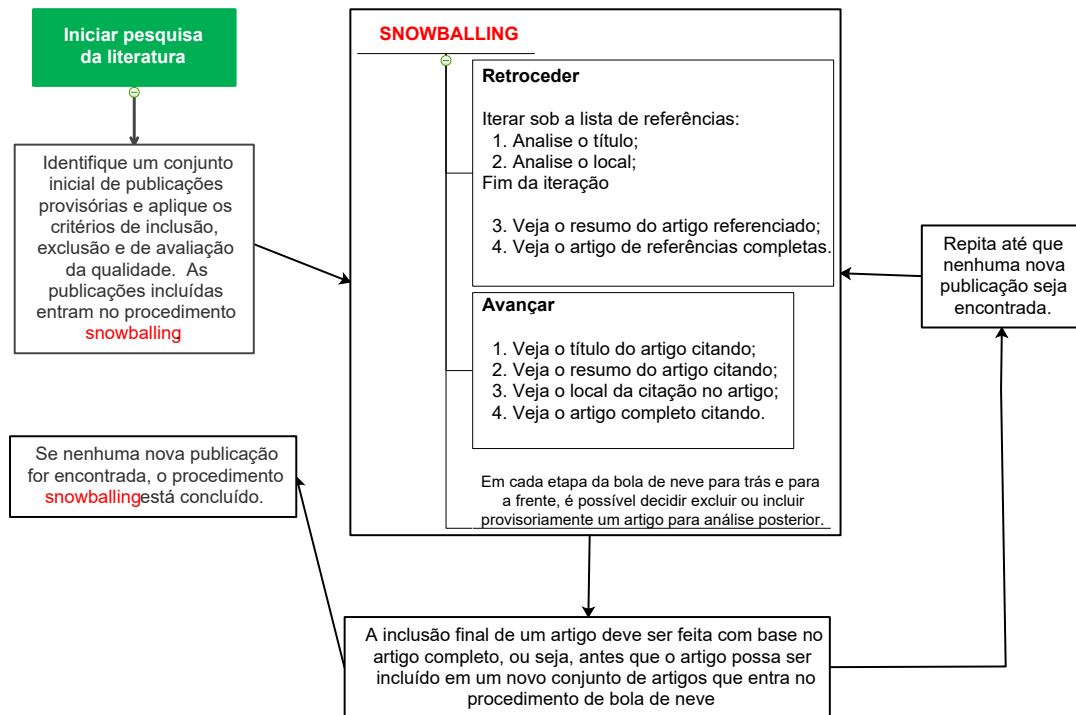


Figura B.1: Procedimento *Snowballing*. Adaptado de Wohlin (2014)

Passo 1: *Seleção do conjunto inicial de estudos.* Também conhecido como conjunto provisório, estes são os estudos iniciais utilizados na primeira iteração do procedimento. O conjunto inicial real é formado pelos estudos selecionados a partir do conjunto provisório, estes estudos são examinados em sua totalidade antes da decisão de inclusão. Busca-se identificar as seguintes características no conjunto provisório:

- Estudos oriundos de diferentes comunidades;
- Número de estudos não muito pequeno;
- Se muitos artigos forem encontrados deve-se escolher os artigos relevantes e altamente citados;
- Conjunto diverso, abrangendo várias editoras, anos e autores diferentes;
- Busca guiada a partir de palavras-chave nas questões de pesquisa.

Outra característica importante é a inclusão de pelo menos um artigo original e altamente citado na área de estudo sistemático da literatura. A Figura B.2 apresenta um exemplo de conjunto inicial de estudos, considerando o tema desta pesquisa.

Palavra Chave Inicial: Dynamic Multi-objective Optimisation									
Data: 26/07/21									
Nota 1: A quantidade de citações é a indicada pelo Google Scholar.									
Nota 2: As versões consideradas são as indexadas pelo Google Scholar.									
	Titulo do artigo	Journal / Conference	Editora / Organization	Ano	Qtd. Citações	Qtd. Referências	Qtd. Versões	Tipo	
1	Performance measures for dynamic multi-objective optimisation algorithms	Information Sciences	Elsevier	2013	81	76	8	Periódico	
2	Benchmarks for dynamic multi-objective optimisation algorithms	ACM Computing Surveys (CSUR)	ACM New York, NY, USA	2014	59	74	4	Periódico	
3	Analysing the performance of dynamic multi-objective optimisation algorithms	2013 IEEE Congress on Evolutionary Computation	IEEE	2013	38	39	3	Conferência	
4	Benchmarks for dynamic multi-objective optimisation	2013 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)	IEEE	2013	34	33	3	Conferência	
5	Dynamic multi-objective optimisation using PSO	Springer, Berlin, Heidelberg	Springer, Berlin, Heidelberg	2010	37	29	9	Periódico	
6	Dynamic multi-objective optimisation for machining gradient materials	CIRP annals	Elsevier	2008	44	15	5	Periódico	
7	Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems	Swarm and Evolutionary computation		2014	43	131	2	Periódico	
8	Key Challenges and future directions of dynamic multi-objective optimisation	2016 IEEE Congress on Evolutionary Computation (CEC)	IEEE	2016	14	39	4	Conferência	
9	Dynamic multi-objective optimisation using deep reinforcement learning: benchmark, algorithm and an application to identify vulnerable zones based on water quality	Engineering Applications of Artificial Intelligence	Elsevier	2019	7		10	Periódico	
10	Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation	2011 IEEE Congress of Evolutionary Computation (CEC)	IEEE	2011	45	21	2	Conferência	
11	Heterogeneous dynamic vector evaluated particle swarm optimisation for dynamic multi-objective optimisation	2014 IEEE Congress on Evolutionary Computation (CEC)	IEEE	2014	22	33	1	Conferência	
12	Analyses of guide update approaches for vector evaluated particle swarm optimisation on dynamic multi-objective optimisation problems	2012 IEEE Congress on Evolutionary Computation	IEEE	2012	23	15	5	Conferência	
13	Solving dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation	Theses and Dissertations (University of Pretoria)	University of Pretoria	2012	27	173	2	Tese de Doutorado	
14	Multi-objective optimisation using evolutionary algorithms: an introduction	Multi-objective Evolutionary Optimisation for Product Design and Manufacturing	Springer	2011	19328	71	22	Capítulo eBook	
15	Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation	2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)	IEEE	2008	87	27	7	Conferência	
16	Dynamic multi-objective optimization using evolutionary algorithms: a survey	Recent Advances in Evolutionary Multi-objective Optimization	Springer	2017	57	97	6	Capítulo eBook	
17	Design and Development of a Benchmark for Dynamic Multi-objective Optimisation Problem in the Context of Deep Reinforcement Learning	2019 22nd International Conference on Computer and Information Technology (ICCIIT)	IEEE	2019	1	24	1	Conferência	
18	Challenges of dynamic multi-objective optimisation	2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence	IEEE	2013	3	90	4	Conferência	
19	Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation	Journal of Mathematical Modelling and Algorithms	Springer	2004	182	25	9	Periódico	
20	Multi-objective Optimisation Based on Relation Favour	International Conference on Evolutionary Multi-Criterion Optimization	Springer	2001	134	16	14	Conferência	

Figura B.2: Exemplo de um conjunto inicial de estudos para o Procedimento Bola de Neve (*Snowballing*) construído a partir da *string* de busca “Dynamic Multi-objective Optimisation” através do Google Scholar.

Considerando as particularidades dessa estratégia de busca, para evitar preconceitos em favor de qualquer editora, uma opção interessante é encontrar o conjunto provisório de publicações utilizando o Google Scholar através de alguma *string* de busca que descreve bem o tema, por exemplo, “*Multi-Objective Dynamic Optimization*”.

A partir do conjunto provisório, o conjunto real deve ser encontrado aplicando-se critérios de inclusão, exclusão e de qualidade. Critérios de inclusão podem ser entendidos como características essenciais, presentes na publicação, que ajudarão a responder questões de pesquisa. Critérios de exclusão podem ser entendidos como características que, apesar de atenderem aos critérios de inclusão, podem interferir pesquisa. Também é interessante a aplicação de critérios de qualidade, isso para evitar que estudos mal conduzidos, com resultados duvidosos, sejam incluídos e a partir desses, inferências erradas sejam deduzidas.

As Tabelas B.1, B.2 e B.3 apresentam exemplos de critérios de inclusão, critérios de exclusão e critérios de qualidade, respectivamente. Sobre os critérios de qualidade, por exemplo, a seleção pode considerar apenas publicações com pontuação igual a quatro.

Tabela B.1: Critérios de Inclusão.

Critérios de Inclusão
Estudos publicados em conferências, journals e revistas após o ano de 2000. Dissertação de mestrado e teses de doutorado. Publicações que, quando propõem novos problemas de teste e/ou métricas de desempenho, os descrevem matematicamente. Estudos relacionados ao tema: algoritmos evolutivos, problemas de teste e métricas de desempenho no contexto da otimização dinâmica.

Tabela B.2: Critérios de Exclusão.

Critérios de Exclusão
Estudos publicados antes do ano 2000. Livros didáticos impressos e eletrônicos. Estudos em outros domínios do conhecimento. Relatórios técnicos, white papers, resumos ou slides. Projetos de graduação, anais de seminários de Iniciação Científica. Estudos publicados em idiomas diferentes do inglês e português. Publicações múltiplas ou duplicatas (a versão mais completa é incluída).

Tabela B.3: Critérios da Avaliação de Qualidade

Critérios de Qualidade	Pontuação
Os resultados e descobertas estão claramente apresentados?	1
Os argumentos são bem apresentados e justificados?	1
A tecnologia apresentada está claramente definida?	1
A pesquisa apresentada é passível de replicação?	1

Passo 2: Iterações. Consiste em analisar as referências e citações dos estudos do conjunto real em busca de novos estudos relevantes.

- *Backward Snowballing*: Utiliza a lista de referências para identificar novos estudos a serem incluídos. Os sub passos desse passo são: (1) Triagem inicial, feita com base nas informações encontradas na lista de referências (título, local de publicação, autores); (2) Com base na triagem inicial, são selecionados os estudos que formam o grupo de candidatos à inclusão; e (3) Busca pelos

artigos potencialmente novos, leitura de seus resumos e partes relevantes para posterior decisão de inclusão.

- *Forward Snowballing*: Analisa os estudos que citam o estudo que está sendo examinado. As citações podem ser analisadas utilizando o Google Scholar. Os sub passos desse passo são: (1) Triagem inicial, feita com base nas informações fornecidas no Google Scholar; (2) Se as informações da triagem inicial forem insuficientes para uma decisão de inclusão provisória, o artigo que cita deve ser estudado com mais detalhes por meio da leitura de seu resumo e investigação de seu local de publicação; (3) Se as informações ainda forem insuficientes, outras partes do texto devem ser estudadas para se tomar uma decisão sobre o novo artigo.
- Aplicação dos critérios de inclusão e exclusão de estudos: Antes de utilizar um estudo no procedimento *Snowballing*, é preciso verificar se este atende aos critérios de inclusão, exclusão e de avaliação da qualidade, estes, previamente definidos. Se, em uma iteração posterior, for identificado um estudo que não deveria ter sido incluído no procedimento, todos os artigos encontrados a partir deste devem ser removidos.

Deve-se desconsiderar também estudos examinados em iterações anteriores.

Passo 3: *Extração dos dados.* A extração de informações dos estudos selecionados deve ser feita ao fim de cada iteração do procedimento, sendo conduzido de acordo com as questões de pesquisa. Para facilitar o processo de extração de informações, conforme sugerido por Keele et al. (2007), pode ser utilizado um formulário, como o apresentado na Figura B.4.

Passo 4: *Fim do procedimento.* O loop é encerrado quando nenhum artigo novo é encontrado nas iterações.

B.2 Busca em Bibliotecas Digitais

A segunda estratégia de busca, Busca em Bibliotecas Digitais, seguirá os protocolos descritos por Keele et al. (2007), delineado nas etapas da Figura B.3 e descrito brevemente a seguir.

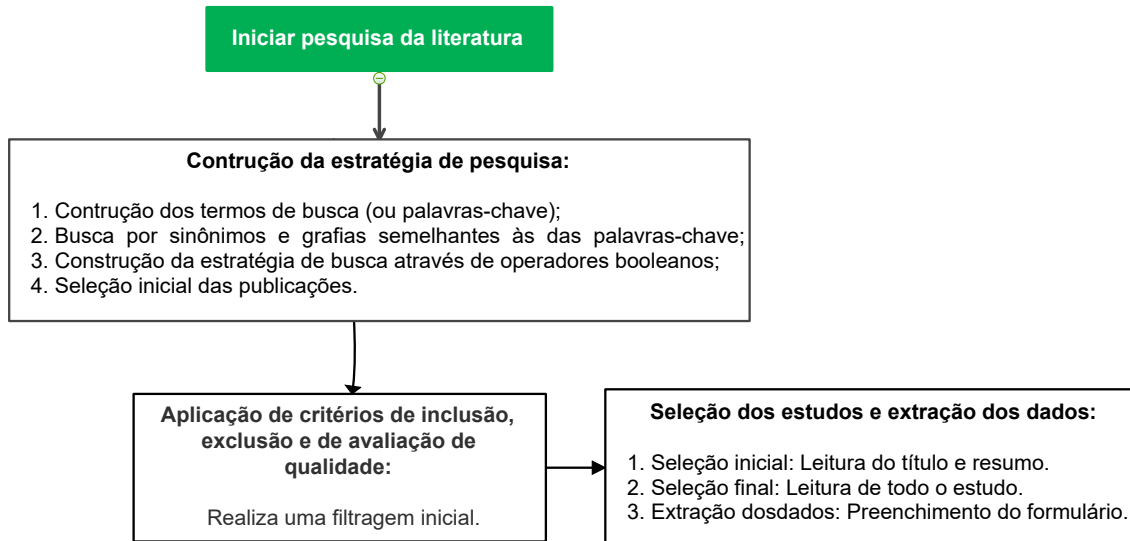


Figura B.3: Procedimento de busca por estudos em bibliotecas digitais

Etapa 1: *Construção da estratégia de pesquisa e busca por estudos relevantes.*

- Definição dos termos de busca: Para encontrar os termos de busca, ou palavras-chave, pode-se utilizar os critérios População, Intervenção, Comparação e Resultados (PICO, do inglês *Population, Intervention, Comparison, Outcome*). Como sugestão, a base para a construção desses termos pode ser os títulos dos artigos encontrados no procedimento *Snowballing*.
- Busca por sinônimos e grafias semelhantes às das palavras-chave: Nesta etapa, os termos encontrados anteriormente são derivados para identificar sinônimos e grafias semelhantes.
- Construção da *string* de busca através de operadores booleanos: Através dos operadores booleanos, por exemplo, AND e OR, os termos e seus derivados são conectados para formar uma *string* de busca padrão. Um exemplo de *string* padrão simples seria "(problem OR benchmarks OR algorithm OR quality indicator) AND (dynamic multi-objective optimisation)".
- Seleção inicial das publicações: A partir da *string* definida na sub etapa anterior, os estudos são selecionados por meio de bibliotecas digitais, alguns exemplos são: ACM Digital Library; IEEE Explore; Science Direct e; Springer Link. Como sugestão, a escolha das bibliotecas podem considerar os resultados da aplicação do procedimento *Snowballing*.

Etapa 2: *Aplicação de critérios de inclusão, exclusão e de avaliação de qualidade.* Nessa etapa, devem ser aplicados os critérios de inclusão e exclusão previamente definidos, assim como os critérios de avaliação da qualidade. Estes podem ser os mesmos definidos para o procedimento *Snowballing*.

Etapa 3: *Seleção dos estudos e extração dos dados.*

- O processo de seleção dos estudos compreende duas etapas, o processo de seleção inicial e o processo de seleção final (Niazi, 2015). No processo de seleção inicial, o título e resumo são lidos para tomada de decisão sobre sua inclusão ou não, se estas informações forem insuficientes, outras partes do estudo são investigadas. O processo final consiste na leitura de todas as partes do estudo.
- De modo semelhante ao procedimento *Snowballing*, o processo de extração dos dados pode consistir no preenchimento de determinado formulário, podendo, inclusive, ser o mesmo utilizado anteriormente.

B.3 Busca Informal

Para complementar a busca por estudos relevantes, os autores mais recorrentes podem ser identificados e suas publicações examinadas, por exemplo, por meio da análise de seus perfis no site Google Scholar. Também, podem ser contactados especialistas e pesquisadores locais que trabalham na área para perguntar-lhes se sabem de algum resultado não publicado ou estudo não incluído.

Ao fim dos diferentes procedimentos de busca por estudos relevantes, descritos acima, os resultados devem ser sintetizados para se obter respostas às questões de pesquisa realizadas.

Para verificar a consistência da seleção dos estudos e da extração dos dados, pode ser aplicado o processo de teste-reteste. Referente à seleção, pode-se selecionar aleatoriamente, pelo menos, dez estudos do conjunto provisório e verificar a utilização destes na pesquisa. Referente à extração de dados, pode-se selecionar aleatoriamente, pelo menos, dez estudos do conjunto final e verificar se a extração de seus dados condiz com a extração feita anteriormente.

Informações Gerais	
Data da revisão:	
Título:	
Autores:	
Local:	
Ano:	
Editora:	

O estudo apresenta ou enfatiza dificuldades relacionadas a DMOO?
<input type="checkbox"/> Sim
<input type="checkbox"/> Não
Se sim: O estudo apresenta ou enfatiza alguma estratégia de superação das dificuldades?
Observações:

O Estudo apresenta novas tecnologias?
<input type="checkbox"/> DMOAs
<input type="checkbox"/> Benchmarks
<input type="checkbox"/> Métricas
Observações:

O estudo realiza análises comparativas?
Se sim: Quais algoritmos foram utilizados?
Se sim: Quais indicadores de qualidade foram utilizados?
Observações:

Figura B.4: Formulário de extração dos dados.

Apêndice C

Modelagem de um Problema Real NRP (*Next Release Problem*)

Para exemplificação da modelagem de um problema real NRP, consideramos o projeto de *software* descrito em Greer and Ruhe (2004), o qual foi posteriormente aplicado em del Sagrado et al. (2010). Nesse projeto são especificados 20 requisitos $R = \{r_1, r_2, \dots, r_{20}\}$ com 5 clientes $C = \{c_1, c_2, \dots, c_5\}$.

Cada requisito tem uma estimativa de esforço associada em termos de uma pontuação que varia entre 1 e 10 (tabela C.1), sendo o esforço total em cada incremento menor ou igual a 25. Cada cliente atribui uma prioridade a cada requisito considerado, utilizando uma escala de importância que varia de 1 a 5. Esses valores são apresentados nas tabelas C.2 e C.3.

Tabela C.1: Custo dos requisitos (r_{11}, \dots, r_{20}) para a empresa

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
1	4	2	3	4	7	10	2	1	3

r_{11}	r_{12}	r_{13}	r_{14}	r_{15}	r_{16}	r_{17}	r_{18}	r_{19}	r_{20}
2	5	8	2	1	4	10	4	8	4

Tabela C.2: Importância dos requisitos (r_1, \dots, r_{10}) para os clientes

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
c_1	4	2	1	2	5	5	2	4	4	4
c_2	4	4	2	2	4	5	1	4	4	5
c_3	5	3	3	3	4	5	2	4	4	4
c_4	4	5	2	3	3	4	2	4	2	3
c_5	5	4	2	4	5	4	2	4	5	2

Tabela C.3: Importância dos requisitos (r_{11}, \dots, r_{20}) para os clientes

	r_{11}	r_{12}	r_{13}	r_{14}	r_{15}	r_{16}	r_{17}	r_{18}	r_{19}	r_{20}
c_1	2	3	4	2	4	4	4	1	3	2
c_2	2	3	2	4	4	2	3	2	3	1
c_3	2	4	1	5	4	1	2	3	3	2
c_4	5	2	3	2	4	3	5	4	3	2
c_5	4	5	3	4	4	1	1	2	4	1

Tabela C.4: Importância dos clientes para o projeto

Clientes	c_1	c_2	c_3	c_4	5
Importância	4	4	3	5	5

Os clientes são avaliados do ponto de vista do gerente de projetos, o qual atribui a cada cliente um peso de importância, como mostra a tabela C.4.

As restrições do problema incluem precedência de requisitos e exigência de que alguns sejam implementados em conjunto (em um mesmo release). A definição de precedência é definida a seguir, o qual indica, por exemplo, que r_4 deve vir antes de r_8 e r_{17} , r_8 antes de r_{17} e assim por diante.

$$\psi = \begin{cases} (r_4, r_8), (r_4, r_{17}), \\ (r_8, r_{17}), \\ (r_9, r_3), (r_9, r_6), (r_9, r_{12}), (r_9, r_{19}), \\ (r_{11}, r_{19}) \end{cases} \quad (\text{C.1})$$

Os requisitos que devem ser implementados em conjunto estão descritos a seguir, o qual indica, por exemplo, que o requisito r_3 e r_{12} , devem ser implementados em conjunto, assim como os requisitos r_{11} e r_{13} .

$$\varepsilon = \begin{cases} (r_3, r_{12}), \\ (r_{11}, r_{13}), \end{cases} \quad (\text{C.2})$$

Em algumas implementações, as empresas solicitam que os clientes apenas listem os requisitos de seu interesse, de modo que as tabelas C.2 e C.3 teriam apenas valores binários, com 1 indicando o interesse e 0 não.