



Pós-Graduação em Ciência da Computação

ELLIACKIN MESSIAS DO NASCIMENTO FIGUEIREDO

**UMA NOVA ABORDAGEM BASEADA EM
ENXAMES DE PARTÍCULAS PARA OTIMIZAÇÃO
DE MUITOS OBJETIVOS**



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE
2017

Elliackin Messias do Nascimento Figueiredo

**Uma Nova Abordagem Baseada em Enxames de Partículas para Otimização de Muitos
Objetivos**

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

ORIENTADORA: Profa. Teresa Bernarda Ludermir
COORIENTADOR: Prof. Carmelo José Albanez Bastos-Filho

RECIFE
2017

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

F475n Figueiredo, Elliackin Messias do Nascimento
Uma nova abordagem baseada em enxames de partículas para otimização
de muitos objetivos / Elliackin Messias do Nascimento Figueiredo. – 2017.
149 f.:il, fig., tab.

Orientadora: Teresa Bernarda Ludermir.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da
Computação, Recife, 2017.
Inclui referências.

1. Inteligência artificial. 2. Otimização. 3. Algoritmos evolucionários. I.
Ludermir, Teresa Bernarda (orientadora). II. Título.

006.3

CDD (23. ed.)

UFPE- MEI 2017-157

Elliackin Messias do Nascimento Figueiredo

**UMA NOVA ABORDAGEM BASEADA EM ENXAMES DE
PARTÍCULAS PARA OTIMIZAÇÃO DE MUITOS OBJETIVOS**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação

Aprovado em: 17/02/2017.

Orientador: Profa. Dra. Teresa Bernarda Ludermir

BANCA EXAMINADORA

Prof. Dr. Cleber Zanchettin
Centro de Informática / UFPE

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

Prof. Dr. Rafael Stubs Parpinelli
Departamento de Ciência da Computação / UESC

Prof. Dr. Danilo Ricardo Barbosa de Araújo
Departamento de Estatística e Informática / UFRPE

Profa. Dra. Aurora Trinidad Ramirez Pozo
Departamento de Informática / UFPR

Dedico esta tese, com todo meu amor, a Otávia Nascimento por sua presença indelével e apaixonante na minha vida.

Agradecimentos

À Otávia Nascimento que, para minha felicidade, surgiu de repente na minha vida. Obrigado por compartilhar um pouco da sua vida comigo. Não importa o que aconteça, quando aconteça, como aconteça, nunca te esquecerei.

Aos meus orientadores.

Aos meus pais.

Para todos aqueles que querem realmente meu bem.

A todas essas pessoas, meus mais sinceros agradecimentos e sentimentos de dívida e gratidão.

*“A cachoeira! Paulo Afonso! O abismo!
A briga colossal dos elementos!
As garras do Centauro em paroxismo
Raspando os flancos dos parcéis sangrentos.
Relutantes na dor do cataclismo
Os braços do gigante suarentos
Agüentando a ranger (espanto! assombro!)
O rio inteiro, que lhe cai do ombro.”*
(Castro Alves)

Resumo

Many-Objective Optimization Problems (MaOPs) são uma classe especial de problemas multiobjetivos que apresentam quatro ou mais objetivos. Algoritmos evolucionários ou de enxame de partículas tradicionais falham ao tentarem resolver MaOPs pois eles se tornam ineficazes ou ineficientes nestes problemas. Em virtude disso, alguns pesquisadores propuseram diferentes estratégias para contornar as dificuldades impostas por MaOPs, sobretudo para *Multiobjective Evolutionary Algorithms* (MOEAs) tradicionais para esses problemas. Em contrapartida, muito pouco tem sido feito no sentido de adaptar algoritmos de *Multiobjective Particle Swarm Optimizations* (MOPSOs) para MaOPs. Não obstante, algoritmos baseados em enxames de partículas no geral são reconhecidos pela rápida velocidade de convergência em problemas com um único objetivo e por isso parecem ser também adequados para problemas multiobjetivos e com muitos objetivos. Desta forma, existe a necessidade de se desenvolver MOPSOs para lidar com MaOPs. Sendo assim, o objetivo desta tese foi desenvolver um MOPSO para resolver MaOPs no sentido de promover um maior balanceamento entre convergência para Frente de Pareto e diversidade de soluções nesses problemas. Para isso, o algoritmo proposto nesta tese usa um conjunto de pontos de referência para impor uma pressão de convergência para a Frente de Pareto enquanto permite um maior gerenciamento da diversidade. Além disso, a abordagem proposta usa um arquivo externo em que são armazenadas soluções não-dominadas e do qual são retirados os líderes sociais das partículas de acordo com duas medidas que foram propostas nesta tese, a saber, a medida de convergência e a medida de densidade. O objetivo dessas medidas é promover a convergência para a Frente de Pareto e promover a diversidade ao longo dela simultaneamente. A proposta foi avaliada usando seis problemas da família DTLZ com dois, três, cinco, sete e dez objetivos e usando métricas bem estabelecidas na literatura (distância geracional, distância geracional invertida e hipervolume) para medir a convergência e diversidade do conjunto solução obtido pelo algoritmo; e ele foi comparado com duas abordagens baseadas em enxames (SMPSO e CDAS-SMPSO) e três abordagens evolucionárias (CEGA, MDFA, e NSGA-III) afim de mostrar suas vantagens e pontos de melhorias frente a outros algoritmos bem estabelecidos na literatura. Os resultados mostraram que o algoritmo proposto foi bem sucedido em equilibrar convergência e diversidade nos problemas testados apresentando resultados equivalentes ou superiores ao NSGA-III que é uma das propostas mais bem sucedidas até o momento para lidar com problemas com muitos objetivos. Além disso, o algoritmo proposto foi aplicado em um problema real de projeto de redes ópticas de alta capacidade com o intuito de avaliar sua utilidade prática.

Palavras-chaves: Otimização de Muitos Objetivos. Algoritmos Evolucionários. Enxame de Partículas. Inteligência de Enxames.

Abstract

Many-Objective Optimization Problems (MaOPs) are an especial class of multiobjective problems in which four or more objectives are optimized simultaneously. Currently, these problems have attracted attention of the researchers due the two reasons: (i) many real-world applications are naturally many-objective problems and (ii) population-based heuristics presents great difficulties for solving these problems. For example, Pareto-dominance based algorithms such as Multiobjective Evolutionary Algorithms (MOEAs) and Multiobjective Particle Swarm Optimization (MOPSO) algorithms are ineffective in these problems because almost all solutions in the population become non-dominated solutions, resulting in loss of convergence pressure for the Pareto front. Because of this, researchers have proposed new strategies for dealing with this problems, mainly for MOEAs. However, very little has been done to make the MOPSOs effective in these scenarios. Regardless of this, Particle Swarm Optimization (PSO) algorithms are known by fast speed of convergence in single-objective problems and they seem be suitable for MaOPs. Moreover, to create algorithms that are capable of balancing both convergence and diversity is a research challenge. Therefore, there are a necessity to develop PSO-based algorithms for dealing with MaOPs. Thus, this thesis proposes a new algorithm based on PSO whose aim is to promote a better balancing between convergence toward the Pareto front and diversity of solutions. To achieve this aim, the proposed algorithm uses a set of reference points to impose a selection pressure to Pareto front while it allows a better control of the diversity. Furthermore, our algorithm use an external archive where it maintains non-dominated solutions and from which the social leaders are picked in according to two measures proposed for us, namely, the convergence measure and the density measure. The objective of these measures is to promote the convergence toward Pareto Front and to promote the diversity along it simultaneously. To evaluate our algorithm we used three well-known metrics in the literature, namely, generational distance, inverted generational distance, and hypervolume; and six benchmark problems of the DTLZ family with two, three, five, seven and ten objectives. Moreover, the algorithm was compared to two PSO-based algorithms (SMPSO and CDAS-SMPSO) and three evolutionary algorithms (MDFA, CEGA and NSGA-III). The results showed that our algorithm was successful in balancing the requirements of convergence and diversity in the test problems compared to other algorithms in the literature. Besides, our algorithm was applied in a real-world problem involving the physical topology design of optical network in which it obtained good results.

Keywords: Many-Objective Problems. Evolutionary Algorithm. Particle Swarm Optimization. Swarm Intelligence.

Lista de ilustrações

| | |
|--|----|
| Figura 1 – Relação entre um espaço de decisão bidimensional e um espaço objetivo bidimensional (Figura adaptada de (FIGUEIREDO, 2013)). | 33 |
| Figura 2 – Relações possíveis entre as soluções em um espaço objetivo bidimensional (figura adaptada de (ZITZLER, 1999)). | 34 |
| Figura 3 – Ilustração de uma Frente de Pareto global e local em espaço objetivo bidimensional (figura adaptada de (FIGUEIREDO, 2013)). | 35 |
| Figura 4 – Requisitos desejáveis do conjunto aproximação (conjunto solução) de um algoritmo multiobjetivo (figura retirada de (FIGUEIREDO, 2013)). | 37 |
| Figura 5 – Exemplo do cálculo do <i>Crowding Distance</i> (CD) em um espaço objetivo bidimensional. | 42 |
| Figura 6 – Regiões de dominância para uma dada solução (retirado de (FIGUEIREDO, 2013)). | 45 |
| Figura 7 – Ilustração do processo de geração de pontos de referência. | 55 |
| Figura 8 – Construção do hiperplano a partir das soluções extremas do arquivo externo. | 57 |
| Figura 9 – Projeção dos pontos de referência originais no hiperplano. | 58 |
| Figura 10 – Funcionamento do operador de densidade. | 59 |
| Figura 11 – Ilustração do método ASF. | 61 |
| Figura 12 – Processo de seleção do líder social no <i>Many-Objective Particle Swarm Optimization</i> (MaOPSO) para o sub-exame \bar{S}_1 | 64 |
| Figura 13 – Processo de seleção do líder cognitivo no MaOPSO. | 66 |
| Figura 14 – Resultados de GD em função do número de iterações para o problema DTLZ1 com 10 objetivos. | 83 |
| Figura 15 – Resultados de GD em função do número de iterações para o problema DTLZ2 com 10 objetivos. | 84 |
| Figura 16 – Resultados de GD em função do número de iterações para o problema DTLZ3 com 10 objetivos. | 84 |
| Figura 17 – Resultados de GD em função do número de iterações para o problema DTLZ4 com 10 objetivos. | 85 |
| Figura 18 – Resultados de GD em função do número de iterações para o problema DTLZ5 com 10 objetivos. | 85 |
| Figura 19 – Resultados de GD em função do número de iterações para o problema DTLZ6 com 10 objetivos. | 86 |
| Figura 20 – Resultados da métrica GD para o problema DTLZ1 (depois de 100 000 iterações). | 91 |

| | |
|--|-----|
| Figura 21 – Resultados da métrica GD para o problema DTLZ2 (depois de 100 000 iterações). | 91 |
| Figura 22 – Resultados da métrica GD para o problema DTLZ3 (depois de 100 000 iterações). | 91 |
| Figura 23 – Resultados da métrica GD para o problema DTLZ4 (depois de 100 000 iterações). | 92 |
| Figura 24 – Resultados da métrica GD para o problema DTLZ5 (depois de 100 000 iterações). | 92 |
| Figura 25 – Resultados da métrica GD para o problema DTLZ6 (depois de 100 000 iterações). | 92 |
| Figura 26 – Resultados da métrica IGD para o problema DTLZ1 (depois de 100 000 iterações). | 93 |
| Figura 27 – Resultados da métrica IGD para o problema DTLZ2 (depois de 100 000 iterações). | 93 |
| Figura 28 – Resultados da métrica IGD para o problema DTLZ3 (depois de 100 000 iterações). | 93 |
| Figura 29 – Resultados da métrica IGD para o problema DTLZ4 (depois de 100 000 iterações). | 94 |
| Figura 30 – Resultados da métrica IGD para o problema DTLZ5 (depois de 100 000 iterações). | 94 |
| Figura 31 – Resultados da métrica IGD para o problema DTLZ6 (depois de 100 000 iterações). | 94 |
| Figura 32 – Resultados da métrica HV para o problema DTLZ1 (depois de 100 000 iterações). | 95 |
| Figura 33 – Resultados da métrica HV para o problema DTLZ2 (depois de 100 000 iterações). | 95 |
| Figura 34 – Resultados da métrica HV para o problema DTLZ3 (depois de 100 000 iterações). | 95 |
| Figura 35 – Resultados da métrica HV para o problema DTLZ4 (depois de 100 000 iterações). | 96 |
| Figura 36 – Resultados da métrica HV para o problema DTLZ5 (depois de 100 000 iterações). | 96 |
| Figura 37 – Resultados da métrica HV para o problema DTLZ6 (depois de 100 000 iterações). | 96 |
| Figura 38 – Resultados da métrica GD para o problema DTLZ1 (depois de 1 000 iterações). | 101 |
| Figura 39 – Resultados da métrica GD para o problema DTLZ2 (depois de 1 000 iterações). | 101 |

| | |
|--|-----|
| Figura 40 – Resultados da métrica GD para o problema DTLZ3 (depois de 1 000 iterações). | 102 |
| Figura 41 – Resultados da métrica GD para o problema DTLZ4 (depois de 1 000 iterações). | 102 |
| Figura 42 – Resultados da métrica GD para o problema DTLZ5 (depois de 1 000 iterações). | 102 |
| Figura 43 – Resultados da métrica GD para o problema DTLZ6 (depois de 1 000 iterações). | 103 |
| Figura 44 – Resultados da métrica IGD para o problema DTLZ1 (depois de 1 000 iterações). | 103 |
| Figura 45 – Resultados da métrica IGD para o problema DTLZ2 (depois de 1 000 iterações). | 103 |
| Figura 46 – Resultados da métrica IGD para o problema DTLZ3 (depois de 1 000 iterações). | 104 |
| Figura 47 – Resultados da métrica IGD para o problema DTLZ4 (depois de 1 000 iterações). | 104 |
| Figura 48 – Resultados da métrica IGD para o problema DTLZ5 (depois de 1 000 iterações). | 104 |
| Figura 49 – Resultados da métrica IGD para o problema DTLZ6 (depois de 1 000 iterações). | 105 |
| Figura 50 – Resultados da métrica HV para o problema DTLZ1 (depois de 1 000 iterações). | 105 |
| Figura 51 – Resultados da métrica HV para o problema DTLZ2 (depois de 1 000 iterações). | 105 |
| Figura 52 – Resultados da métrica HV para o problema DTLZ3 (depois de 1 000 iterações). | 106 |
| Figura 53 – Resultados da métrica HV para o problema DTLZ4 (depois de 1 000 iterações). | 106 |
| Figura 54 – Resultados da métrica HV para o problema DTLZ5 (depois de 1 000 iterações). | 106 |
| Figura 55 – Resultados da métrica HV para o problema DTLZ6 (depois de 1 000 iterações). | 107 |
| Figura 56 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ1 com três objetivos. | 111 |
| Figura 57 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ2 com três objetivos. | 111 |
| Figura 58 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ3 com três objetivos. | 112 |

| | |
|--|-----|
| Figura 59 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ4 com três objetivos. | 112 |
| Figura 60 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ5 com três objetivos. | 113 |
| Figura 61 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ6 com três objetivos. | 113 |
| Figura 62 – Coordenadas paralelas dos conjuntos referências dos algoritmos para os problemas de teste com dez objetivos. | 114 |
| Figura 63 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ1 com dez objetivos. | 114 |
| Figura 64 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ2 com dez objetivos. | 115 |
| Figura 65 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ3 com dez objetivos. | 115 |
| Figura 66 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ4 com dez objetivos. | 116 |
| Figura 67 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ5 com dez objetivos. | 116 |
| Figura 68 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ6 com dez objetivos. | 117 |
| Figura 69 – Média do hipervolume ao longo das 5000 gerações para o NSGA-III e MaOPSO. | 122 |
| Figura 70 – Conjunto aproximação do MaOPSO e do NSGA-III depois de 5000 iterações. | 123 |
| Figura 71 – Topologia física das soluções destacadas na Figura 70. | 124 |

Lista de tabelas

| | |
|---|-----|
| Tabela 1 – Parâmetros usados nos algoritmos de acordo com número de objetivos. | 78 |
| Tabela 2 – Número de iterações e avaliações de aptidão por algoritmo e para cada objetivo. | 80 |
| Tabela 3 – Número de vezes em que o MaOPSO foi estatisticamente superior (B), igual (E) ou inferior (W) aos outros algoritmos considerando 100 000 iterações. | 108 |
| Tabela 4 – Número de vezes em que o MaOPSO foi estatisticamente superior (B), igual (E) ou inferior (W) aos outros algoritmos considerando 1 000 iterações. | 108 |
| Tabela 5 – Média e desvio padrão do tempo de execução dos algoritmos em segundos. | 118 |

Lista de abreviaturas e siglas

θ -DEA *θ -Dominance-based Evolutionary Algorithm.*

ABC *Artificial Bee Colony.*

ACO *Ant Colony Optimization.*

ASF *Achievement Scalarizing Function.*

CAPEX *Capital Expenditure.*

CD *Crowding Distance.*

CDAS *Control of Dominance Area of Solutions.*

CDAS-SMPSO *SMPSO with Control of Dominance Area of Solutions.*

CEGA *Clustering-based Elitist Genetic Algorithm.*

crEA *clustering-ranking Evolutionary Algorithm.*

DMO *Diversity Management Operator.*

EA *Evolutionary Algorithm.*

GD *Generational Distance.*

GDR *Global Detriment Ranking.*

GrEA *Grid-based Evolutionary Algorithm.*

HV *Hypervolume.*

HypE *Hypervolume Estimation Algorithm.*

I-DBEA *Improved Decomposition-based Evolutionary Algorithm.*

IBEA *Indicator-Based Evolutionary Algorithm.*

IGD *Inverted Generational Distance.*

MaOEA-DDFC *Many-Objective Evolutionary Algorithm (MaOEA) based on Directional Diversity (DD) and Favorable Convergence (FC).*

MaOP *Many-Objective Optimization Problem.*

MaOPSO *Many-Objective Particle Swarm Optimization.*

MDFA *Multi-Directional Fitness Assignment.*

MO-NSGA-II *Many-Objective NSGA-II.*

MOEA *Multiobjective Evolutionary Algorithm.*

MOEA/D *Multiobjective Evolutionary Algorithm based on Decomposition.*

MOEA/DD *Multiobjective Evolutionary Algorithm based on Decomposition and Dominance.*

MOP *Multiobjective Optimization Problem.*

MOPSO *Multiobjective Particle Swarm Optimization.*

MOPSO-GD *Multiobjective Particle Swarm Optimization with Global Detriment.*

MSOPS *Multiple Single Objective Pareto Sampling.*

MSOPS-II *Multiple Single Objective Pareto Sampling II.*

NBI *Normal Boundary Intersection.*

NSGA-II *Nondominated Sorting Genetic Algorithm II.*

NSGA-III *Nondominated Sorting Genetic Algorithm III.*

PCSEA *Pareto corner search evolutionary algorithm.*

PICEA *Preference-Inspired Coevolutionary Algorithm.*

PSO *Particle Swarm Optimization.*

RVEA *Reference Vector guided Evolutionary Algorithm.*

SDE *Shift-based Density Estimation.*

SI *Swarm Intelligence.*

SMPSO *Speed-constrained Multi-objective PSO.*

SMS-EMOA *S Metric Selection Evolutionary Multiobjective Optimization Algorithm.*

SPEA2 *Strength Pareto Evolutionary Algorithm 2.*

VaEA *Vector angle-based Evolutionary Algorithm.*

Lista de símbolos

| | |
|--------------------|--|
| x_{max} | limite superior da variável de decisão x |
| x_{min} | limite inferior da variável de decisão x |
| \bar{S}_1 | primeiro sub-enxame do enxame S |
| \bar{S}_2 | segundo sub-enxame do enxame S |
| δ | especificação OXC |
| η_m | índice da distribuição polinomial da mutação |
| Λ | conjunto de pontos de referência original |
| Λ^* | conjunto de pontos de referência projetado |
| λ_{N-1} | conectividade algébrica da rede |
| \mathcal{O} | região factível |
| \mathcal{PF}^* | Frente de Pareto global |
| μ_a | medida de densidade de uma solução a |
| Ω | conjunto factível |
| ω | coeficiente de inércia |
| Ω^* | conjunto Pareto-ótimo |
| π | proporção de partículas que são afetadas pelo operador de mutação |
| Ψ | conjunto de soluções extremas do arquivo externo A_t na iteração t |
| ρ_a | medida de convergência de uma solução a |
| $\hat{y}_i(t)$ | melhor posição encontrada até o momento t pelas partículas vizinhas da partícula i |
| \mathbf{e}_j | solução extrema na direção do objetivo j |
| \mathbf{v}_i | velocidade da partícula i |
| \mathbf{y}_i | melhor posição pessoal da partícula i |
| \mathbf{z}^* | ponto ideal |
| \mathbf{z}^{min} | ponto ideal estimado |

| | |
|-----------|--|
| ζ | fator de divisão do enxame |
| A_t | arquivo externo na iteração t |
| c_1 | coeficiente de aceleração cognitivo |
| c_2 | coeficiente de aceleração social |
| l_i | líder social da partícula i |
| m | número de objetivos |
| N | número de indivíduos ou partículas |
| n | número de variáveis de decisão |
| p | número de divisões do hiperplano |
| P^* | conjunto de referência de pontos da Frente de Pareto |
| $P^{(t)}$ | população de indivíduos na geração t |
| p_m | probabilidade de mutação |
| PB | probabilidade de bloqueio |
| PF | conjunto aproximação da Frente de Pareto do algoritmo |
| $Q^{(t)}$ | população de descendentes na geração t |
| t_{max} | número máximo de iterações |
| w | número comprimento de onda |
| z_{ref} | ponto de referência para o cálculo do hipervolume especificado |
| Z_t | arquivo externo de soluções extremas na iteração t |

Sumário

| | | |
|----------|---|----|
| 1 | Introdução | 20 |
| 1.1 | Motivação e Objetivo | 26 |
| 1.2 | Organização da Tese | 31 |
| 2 | Conceitos Básicos | 32 |
| 2.1 | Otimização Multiobjetivo | 32 |
| 2.2 | Otimização Evolucionária Multiobjetivo | 38 |
| 2.3 | Otimização Multiobjetivo por Enxames de Partículas | 40 |
| 2.4 | Otimização de Problemas com Muitos Objetivos | 43 |
| 2.5 | Abordagens Para Lidar com Problemas com Muitos Objetivos | 46 |
| 2.6 | Considerações Finais | 50 |
| 3 | Otimização por Enxame de Partículas de Muitos Objetivos (MaOPSO) | 51 |
| 3.1 | Arcabouço do MaOPSO | 51 |
| 3.2 | Componentes do MaOPSO | 53 |
| 3.2.1 | Método de Atribuição de Aptidão | 53 |
| 3.2.1.1 | Geração de Pontos de Referência | 53 |
| 3.2.1.2 | Deslocamento das Soluções | 55 |
| 3.2.1.3 | Construção do Hiperplano | 55 |
| 3.2.1.4 | Operador de Densidade | 58 |
| 3.2.1.5 | Operador de Convergência | 59 |
| 3.2.2 | Seleção dos Líderes Sociais | 60 |
| 3.2.2.1 | Seleção dos Líderes Sociais para \bar{S}_1 | 62 |
| 3.2.2.2 | Seleção dos líderes sociais para \bar{S}_2 | 63 |
| 3.2.3 | Seleção dos líderes cognitivos | 65 |
| 3.2.4 | Perturbação do Enxame | 65 |
| 3.2.5 | Atualização do Arquivo Externo | 68 |
| 3.2.6 | Considerações Finais | 69 |
| 4 | Experimentos e Resultados | 70 |
| 4.1 | Arranjo Experimental | 70 |
| 4.1.1 | Algoritmos Utilizados Para Comparação | 70 |
| 4.1.1.1 | Speed-constrained Multi-objective PSO (SMPSO) | 72 |
| 4.1.1.2 | SMPSO with Control of Dominance Area of Solutions (CDAS) | 73 |
| 4.1.1.3 | Clustering-based Elitist Genetic Algorithm - CEGA | 73 |

| | | |
|----------|--|------------|
| 4.1.1.4 | Multi-Directional Fitness Assignment - MDFA | 74 |
| 4.1.1.5 | NSGA-III | 74 |
| 4.1.2 | Problemas de Teste | 75 |
| 4.1.3 | Métricas de Qualidade | 75 |
| 4.1.4 | Delineamento Experimental e Configurações do Parâmetros dos Algoritmos | 77 |
| 4.1.4.1 | Algoritmo Proposto: MaOPSO | 79 |
| 4.1.4.2 | SMPSO & CDAS-SMPSO | 80 |
| 4.1.4.3 | CEGA & MDFA | 81 |
| 4.1.4.4 | NSGA-III | 81 |
| 4.2 | Resultados nas Funções de Teste da Família DTLZ | 81 |
| 4.2.1 | Análise de Convergência | 81 |
| 4.2.1.1 | Considerações Finais Sobre Análise de Convergência | 83 |
| 4.2.2 | Resultados para GD, IGD, e Hipervolume para 100 000 iterações | 86 |
| 4.2.3 | Resultados para GD, IGD, e Hipervolume para 1 000 iterações | 97 |
| 4.2.4 | Análise de Significância Estatística Dos Resultados | 107 |
| 4.2.5 | Visualização das Frentes de Pareto | 109 |
| 4.2.6 | Análise do Tempo de Execução | 117 |
| 4.3 | Estudo de Caso: Aplicação do MaOPSO no Problema de Projeto de Redes Ópticas | 118 |
| 4.4 | Considerações Finais | 125 |
| 5 | Conclusão e Trabalhos Futuros | 126 |
| 5.1 | Considerações Finais e Conclusão | 126 |
| 5.1.1 | Contribuições da Tese | 127 |
| 5.1.2 | Limitações do Trabalho | 128 |
| 5.2 | Trabalhos Futuros | 133 |
| 5.3 | Publicações Descorrentes do Doutorado | 137 |
| | Referências | 139 |

1 Introdução

Otimização é um problema ubíquo (YANG, 2010b; YANG, 2010a; TALUKDER, 2011; NOCEDAL; WRIGHT, 1999). Ela está em tudo e em todos os lugares (YANG, 2010b; YANG, 2010a; TALUKDER, 2011; NOCEDAL; WRIGHT, 1999), na natureza e nas atividades humanas (YANG, 2010b; YANG, 2010a; TALUKDER, 2011; NOCEDAL; WRIGHT, 1999). Otimização pode ser definida, grosso modo, como o processo de encontrar as condições que fornecem o valor máximo ou mínimo, dependendo do que se deseja, de uma função (RAO, 2009). Colocando em termos mais simples, otimização é o processo de obter o melhor resultado (solução) possível sob certas circunstâncias (RAO, 2009; TALUKDER, 2011). Sendo assim, otimização está presente e ocorre de modo corriqueiro em praticamente todos os campos do conhecimento científico, desde engenharia e ciência da computação passando pelas ciências naturais e ciências sociais como a economia (YANG, 2010b; YANG, 2010a; RAO, 2009; TALBI, 2009; NOCEDAL; WRIGHT, 1999), tendo, por isso, incontáveis aplicações (TALBI, 2009). É possível encontrar problemas de otimização nas mais diversas atividades humanas que envolvam algum tipo de tomada de decisão (RAO, 2009). Investidores, por exemplo, buscam criar portfólios no sentido de evitar riscos excessivos (minimizar o risco) ao mesmo tempo em que procuram uma alta taxa de retorno (maximizar o lucro) (NOCEDAL; WRIGHT, 1999). Enquanto que industriais desejam maximizar a eficiência no projeto e operação de seus processos de produção (NOCEDAL; WRIGHT, 1999; TALBI, 2009). A natureza também otimiza uma vez que sistemas físicos tendem a um estado de mínima energia (NOCEDAL; WRIGHT, 1999; TALBI, 2009). Isso pode ser visto no caso de moléculas de um sistema químico isolado que reagem entre si até a energia potencial total dos seus elétrons ser minimizada. Outro exemplo é a luz que segue caminhos que minimizam o tempo de propagação (NOCEDAL; WRIGHT, 1999). Finalmente, outra razão porque otimização é importante é porque ela é uma ferramenta fundamental em processos de tomada de decisão (NOCEDAL; WRIGHT, 1999).

Otimização é o ato de obter o melhor resultado sob certas circunstâncias (RAO, 2009). Ela pode ser definida como o processo de buscar por uma ou mais soluções factíveis que correspondam as melhores soluções possíveis dentro de um conjunto de soluções (NOCEDAL; WRIGHT, 1999; OROSA, 2011). Uma tarefa importante e crítica em otimização é a *modelagem* do problema ou sistema a ser otimizado (NOCEDAL; WRIGHT, 1999; BRANKE et al., 2008). Na modelagem de um problema de otimização, é necessário identificar algum *objetivo*, uma medida quantitativa do desempenho do sistema em análise (NOCEDAL; WRIGHT, 1999; BRANKE et al., 2008). Este objetivo pode ser tempo, custo, energia potencial, risco, qualidade, eficiência ou qualquer quantidade ou combinação de quantidades que possam ser representadas por um único número (NOCEDAL; WRIGHT, 1999; TALBI, 2009). O objetivo

ou função objetivo (NOCEDAL; WRIGHT, 1999; RAO, 2009; BOYD; VANDENBERGHE, 2004; BRANKE et al., 2008) depende de certas características do sistema chamadas de variáveis de decisão (BRANKE et al., 2008) ou de projeto (NOCEDAL; WRIGHT, 1999; RAO, 2009; ABRAHAM; JAIN; GOLDBERG, 2005) que devem ou não atender certas restrições. Sendo assim, resolver um problema de otimização consiste, em termos mais formais, em encontrar valores para as variáveis de decisão de modo a minimizar ou maximizar a função objetivo e que atendam as restrições do problema (NOCEDAL; WRIGHT, 1999; RAO, 2009; TALUKDER, 2011; BRANKE et al., 2008). Além da função objetivo, a tarefa de modelagem envolve a identificação das variáveis de decisão (BRANKE et al., 2008) ou projeto e também a identificação das restrições do problema (NOCEDAL; WRIGHT, 1999; BRANKE et al., 2008). Uma vez que o problema a ser otimizado tenha sido modelado, um algoritmo de otimização pode ser utilizado para encontrar a sua solução (NOCEDAL; WRIGHT, 1999; BRANKE et al., 2008).

Embora alguns problemas do mundo real possam ser modelados por meio de um único objetivo, frequentemente é difícil definir todos os aspectos do problema em termos de um único objetivo (ABRAHAM; JAIN; GOLDBERG, 2005). Definir múltiplos objetivos geralmente permite uma modelagem mais satisfatória do problema (ABRAHAM; JAIN; GOLDBERG, 2005). De fato, muitos problemas de otimização do mundo real requerem a otimização simultânea de dois ou mais objetivos geralmente conflitantes entre si e de medidas não-comensuráveis (HU; YEN, 2015; ADRA, 2007; Coello Coello; LAMONT; VELDHUIZEN, 2007; ENGELBRECHT, 2006). Na literatura, um problema de otimização que envolve múltiplos objetivos que devem ser otimizados ao mesmo tempo e que podem apresentar uma natureza conflitante entre si, é chamado de problema de otimização multiobjetivo ou *Multiobjective Optimization Problem* (MOP) em inglês (RAO, 2009; Coello Coello; LAMONT; VELDHUIZEN, 2007).

Ao contrário de um problema que possui um único objetivo, um MOP possui um conjunto de soluções consideradas ótimas e que representam os diversos e melhores compromissos possíveis entre os objetivos conflitantes do problema de otimização (Coello Coello; LAMONT; VELDHUIZEN, 2007; ABRAHAM; JAIN; GOLDBERG, 2005; ENGELBRECHT, 2006). Ou seja, na ausência de qualquer informação sobre a preferência do usuário a respeito das soluções, o conceito de dominância de Pareto (TAN et al., 2013; BRANKE et al., 2008; Coello Coello; LAMONT; VELDHUIZEN, 2007; ADRA, 2007) é utilizado para impor um ordenamento parcial (DEB et al., 2002) entre as soluções do espaço de busca (ABRAHAM; JAIN; GOLDBERG, 2005; BRANKE et al., 2008). Assim, segundo o conceito de dominância de Pareto uma solução pode dominar, ser dominada ou ser não-dominada por outras soluções (ABRAHAM; JAIN; GOLDBERG, 2005). Uma solução domina outra solução se ela não é pior que esta última em todos os objetivos e é melhor em pelo menos um objetivo (ABRAHAM; JAIN; GOLDBERG, 2005). A partir do conceito de dominância de Pareto, é possível definir uma solução ótima para o problema. Sendo assim, uma solução é Pareto-ótima quando ela não é dominada

por nenhuma solução do espaço de busca (ABRAHAM; JAIN; GOLDBERG, 2005). O conjunto de soluções Pareto-ótima é chamado conjunto Pareto-ótimo e representa o conjunto de todos os melhores compromissos entre os objetivos (ABRAHAM; JAIN; GOLDBERG, 2005). O conjunto Pareto-ótimo quando projetado no espaço objetivo, o espaço composto pelo produto cartesiano das funções objetivo, é denominado de Frente de Pareto (Coello Coello; LAMONT; VELDHUIZEN, 2007). Embora existam vários modos de se lidar com MOPs, o principal objetivo quando se busca resolver um MOP é aproximar o seu conjunto Pareto-ótimo (ABRAHAM; JAIN; GOLDBERG, 2005).

Existem várias técnicas para resolver um MOP. Entretanto, o uso de Algoritmos Evolucionários ou *Evolutionary Algorithms* (EAs) em inglês (BÄCK; SCHWEFEL, 1993) na solução destes problemas tem se destacado como uma das mais promissoras (ZHOU et al., 2011; Coello Coello; LAMONT; VELDHUIZEN, 2007). EAs são meta-heurísticas (BOUSSAÏD; LEPAGNOT; SIARRY, 2013; TALBI, 2009) inspiradas nos princípios de Darwin de sobrevivência do mais apto (seleção natural) (BRANKE et al., 2008) e que são caracterizadas por uma população de soluções candidatas e um processo de reprodução que combina essas soluções de alguma maneira de modo a gerar novas soluções candidatas, possivelmente mais promissoras que as anteriores no contexto da otimização; e por um processo de seleção responsável por escolher as soluções mais promissoras para a próxima geração (BOUSSAÏD; LEPAGNOT; SIARRY, 2013). Desta forma, EAs são particularmente adequados para resolver MOPs devido à sua natureza baseada em população que lhes permite encontrar o conjunto Pareto-ótimo em uma única execução, ao invés de realizar uma série de execuções separadas, como é o caso de algumas técnicas clássicas (BRANKE et al., 2008; ABRAHAM; JAIN; GOLDBERG, 2005; Coello Coello; LAMONT; VELDHUIZEN, 2007; PURSHOUSE, 2003). Além disso, EAs não requerem conhecimento sobre o problema para serem aplicados, não utilizam informação de gradiente, são menos suscetíveis a forma e continuidade da Frente de Pareto, e são fáceis de implementar (ABRAHAM; JAIN; GOLDBERG, 2005). Em outras palavras, EAs atuam em problemas em que os métodos clássicos (analíticos) não podem ser utilizados por exigirem informação de derivada (ADRA, 2007; PURSHOUSE, 2003). Algoritmos Evolucionários projetados para resolver MOPs são chamados de Algoritmos Evolucionários Multiobjetivos ou *Multiobjective Evolutionary Algorithms* (MOEAs) em inglês (ZHOU et al., 2011; BOUSSAÏD; LEPAGNOT; SIARRY, 2013; Coello Coello; LAMONT; VELDHUIZEN, 2007). Nesta perspectiva, o objetivo de um MOEA ou de qualquer outra meta-heurística baseada em população aplicada nesses problemas é encontrar um conjunto representativo de soluções da Frente de Pareto completa ou de uma parte dela (BRANKE et al., 2008) denominada de *conjunto aproximação* (ADRA, 2007; PURSHOUSE, 2003).

Outro paradigma de algoritmos empregado na solução de problemas de otimização são aquelas baseadas em Inteligência de Enxames (BOUSSAÏD; LEPAGNOT; SIARRY, 2013; ENGELBRECHT, 2006; CHAN; TIWARI, 2009; ENGELBRECHT, 2007) que é uma classe de algoritmos que reúne todos os algoritmos que se inspiram no comportamento social

(coletivo) de animais, tais como formigas, abelhas, vespas, pássaros e peixes (TALBI, 2009). Algoritmos de Inteligência de Enxames ou *Swarm Intelligence* (SI), em inglês, são um conjunto de algoritmos que utilizam um conjunto de agentes que trocam informações entre si sobre as melhores regiões do espaço de busca do problema e com isso a busca evolui do sentido de convergir para uma solução aproximadamente ótima. Dentre esses algoritmos, destaca-se o algoritmo denominado de Otimização por Enxame de Partículas ou *Particle Swarm Optimization* (PSO) na sigla em inglês (KENNEDY; EBERHART, 1995; BRATTON; KENNEDY, 2007; MEZA et al., 2017; POLI; KENNEDY; BLACKWELL, 2007; TALUKDER, 2011; Coello Coello; LAMONT; VELDHUIZEN, 2007; ZHOU et al., 2011; LAZINICA, 2009).

O PSO é um algoritmo inspirado no comportamento social de grupos de animais como bando de pássaros e cardumes e que foi proposto por James Kennedy e Russell Eberhart em 1995 (KENNEDY; EBERHART, 1995; BRATTON; KENNEDY, 2007; POLI; KENNEDY; BLACKWELL, 2007; IMRAN; HASHIM; KHALID, 2013). O PSO é caracterizado por ser conceitualmente simples, fácil de implementar, e computacionalmente eficiente, em algumas situações, quando comparado aos algoritmos evolucionários (KENNEDY; EBERHART, 2001; HU; YEN, 2015; MEZA et al., 2017). De fato, EAs requerem muitas gerações (iterações) para se obter uma boa aproximação da solução do problema (MEZA et al., 2017), enquanto o PSO é reconhecido por sua alta capacidade de convergência (KENNEDY; EBERHART, 2001; MEZA et al., 2017; COELLO; PULIDO; LECHUGA, 2004). Além disso, ele tem se destacado pelo seu bom desempenho na otimização de problemas mono-objetivo, isto é, problemas com um único objetivo a ser otimizado (HU; YEN, 2015; NEBRO; DURILLO; COELLO, 2013; COELLO; LECHUGA, 2002). No PSO, a posição de cada partícula representa uma solução candidata do problema. As partículas, durante a evolução do algoritmo, interagem entre si e trocam informações sobre as melhores soluções encontradas (BOUSSAÏD; LEPAGNOT; SIARRY, 2013). A partir dessa troca de informações, espera-se que as partículas avancem em direção a solução ótima do problema.

A simplicidade relativa e a aplicação bem sucedida do PSO na vários problemas de otimização (Coello Coello; LAMONT; VELDHUIZEN, 2007; COELLO; PULIDO; LECHUGA, 2004; COELLO; LECHUGA, 2002; PARSOPOULOS; VRAHATIS, 2002; KENNEDY; EBERHART, 2001; RAO, 2009) aliada a sua alta capacidade de convergência (KENNEDY; EBERHART, 2001; MEZA et al., 2017; COELLO; PULIDO; LECHUGA, 2004) motivaram os pesquisadores a adaptar o PSO, originalmente proposto para solução de problemas com um único objetivo, para resolverem problemas multiobjetivos (HU; YEN, 2015). Como o PSO é um algoritmo baseado em população pois utiliza um conjunto de soluções denominadas de partículas, e como eles também não utilizam informação de gradiente, eles também são bons candidatos a solução de MOPs assim como as abordagens evolucionárias (MISHRA; DEHURI; CHO, 2015; Coello Coello; LAMONT; VELDHUIZEN, 2007). As modificações do PSO no sentido de adaptá-los a problemas multiobjetivos deram origem aos algoritmos denominados genericamente de Algoritmos de Otimização Multiobjetiva por Enxame de Partículas ou *Multiobjective*

Particle Swarm Optimization (MOPSO) na sigla em inglês (MISHRA; DEHURI; CHO, 2015; ZHOU et al., 2011; HU; YEN, 2015; Coello Coello; LAMONT; VELDHUIZEN, 2007).

Os MOPSOs e os MOEAs são as abordagens mais representativas no contexto da otimização de múltiplos objetivos utilizando meta-heurísticas (MISHRA; DEHURI; CHO, 2015; MEZA et al., 2017; Coello Coello; LAMONT; VELDHUIZEN, 2007; HU; YEN, 2015; RAO, 2009). Sendo assim, nesta tese, concentrou-se nessas duas classes de algoritmos.

Uma classe importante de MOEAs e MOPSOs são aqueles que utilizam dominância de Pareto (PURSHOUSE, 2003) para discriminar as soluções da população, de modo que se possa determinar se uma solução é melhor que a outra (MEZA et al., 2017; HU; YEN, 2015; Coello Coello; LAMONT; VELDHUIZEN, 2007; PURSHOUSE, 2003; ADRA, 2007). MOEAs e MOPSOs baseados em dominância de Pareto têm obtido bastante sucesso na solução de problemas multiobjetivos de modo que muitos dos algoritmos mais bem sucedidos e modernos propostos na literatura de otimização multiobjetiva evolucionária e de enxames utilizam esse conceito (MEZA et al., 2017; MISHRA; DEHURI; CHO, 2015; HU; YEN, 2015; Coello Coello; LAMONT; VELDHUIZEN, 2007; PURSHOUSE, 2003; ADRA, 2007). O uso do conceito de dominância de Pareto (PURSHOUSE, 2003) para discriminar as soluções representou um marco na história do desenvolvimento de algoritmos evolucionários multiobjetivos de modo que virtualmente todos os algoritmos que surgiram a partir dessa descoberta passaram a adotá-lo de uma forma ou de outra em sua estrutura (Coello Coello, 2006; ADRA, 2007; PURSHOUSE, 2003). O uso de dominância de Pareto é importante sobretudo por estimular a convergência dos algoritmos em direção a Frente de Pareto (Coello Coello, 2006). De fato, o uso do conceito de dominância permitiu que os algoritmos que o utilizavam obtivessem excelentes resultados onde outros algoritmos baseados em técnicas mais simples, como ponderação de objetivos e ordenamento lexicográfico, por exemplo, falhavam (Coello Coello, 2006; HORN; NAFLIOTIS; GOLDBERG, 1994). Esses algoritmos baseados em dominância de Pareto formaram a primeira geração de algoritmos evolucionários multiobjetivos (Coello Coello, 2006). Contudo, um problema sutil mas importante dos algoritmos da primeira geração é que eles podem perder soluções não-dominadas com o passar das gerações (Coello Coello, 2006). Desse modo a incorporação de mecanismos de elitismo nesses algoritmos representou um outro marco na história do desenvolvimento dos algoritmos evolucionários multiobjetivos (Coello Coello, 2006; ADRA, 2007). Desta forma, a partir de 1990 o *elitismo* foi introduzido nesses algoritmos. Com o uso de elitismo, esses algoritmos conseguiam preservar as soluções não-dominadas encontradas durante a busca e com isso obter resultados superiores (Coello Coello, 2006). Com efeito, elitismo é um requisito teórico para garantir a convergência de um MOEA e daí sua importância (Coello Coello, 2006; ADRA, 2007). A incorporação de elitismo mais dominância de Pareto levou ao surgimento da segunda geração de algoritmos evolucionários multiobjetivos (Coello Coello, 2006) cujos principais expoentes são o *Nondominated Sorting Genetic Algorithm II* (NSGA-II) (DEB et al., 2002), *Strength Pareto Evolutionary Algorithm 2* (SPEA2) (ZITZLER; LAUMANN; THIELE,

2001) e o MOPSO de Coello Coello *et al.* (COELLO; PULIDO; LECHUGA, 2004). Desta forma, em suma, o conceito de dominância de Pareto está no cerne dos algoritmos evolucionários e dos algoritmos de inteligência de enxames mais eficazes e modernos (MEZA *et al.*, 2017; HU; YEN, 2015; MISHRA; DEHURI; CHO, 2015), sendo ele um dos principais responsáveis pela promoção da convergência desses algoritmos para a Frente de Pareto (Coello Coello, 2006).

Uma classe especial de MOPs são os problemas com quatro ou mais objetivos que são conhecidos na literatura como problemas de otimização com *muitos* objetivos ou *Many-Objective Optimization Problems* (MaOPs) em inglês (MISHRA; DEHURI; CHO, 2015; LI *et al.*, 2015a; PURSHOUSE, 2003). MaOPs têm chamado atenção da comunidade de Otimização Evolucionária Multiobjetiva por duas razões: 1) muitas aplicações do mundo real apresentam frequentemente quatro ou mais objetivos (JR.; SANTANA; POZO, 2016; LI *et al.*, 2015a; CAI *et al.*, 2015) e 2) MaOPs impõem grandes desafios à MOEAs e MOPSOs para sua solução (LI *et al.*, 2015a). O principal desafio está relacionado com o fato de que o conceito de dominância de Pareto se torna ineficaz em MaOPs (M.FARINA; AMATO, 2002). Isso ocorre porque o número de soluções não-dominadas a uma dada solução cresce exponencialmente com o número de objetivos (FARINA; AMATO, 2004; M.FARINA; AMATO, 2002; ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008; LI *et al.*, 2015a). Isso por sua vez faz com que MOEAs e MOPSOs baseados em dominância de Pareto se tornem ineficazes na solução desses problemas, pois todos os membros da população ou enxame tendem a se tornar não-dominados entre si de modo que, com isso, esses algoritmos não conseguem impor uma direção de busca e conseqüentemente promover a convergência para a Frente de Pareto (LI *et al.*, 2015a). Além desse problema, outro fenômeno que surge em MOEAs e MOPSOs quando eles são aplicados em MaOPs é o fenômeno chamado de *promoção ativa de diversidade* (PURSHOUSE; FLEMING, 2003). A promoção ativa de diversidade consiste no uso da informação da densidade como único critério de seleção para as soluções da população ou enxame. Como é bem conhecido na literatura, os MOEAs e MOPSOs mais bem sucedidos utilizam informação de dominância (Coello Coello; LAMONT; VELDHUIZEN, 2007; ADRA, 2007; Coello Coello, 2006) para promover a convergência para a Frente de Pareto e informação de densidade em torno de cada solução para promover uma maior diversidade de soluções e evitar a convergência para uma única solução (Coello Coello, 2006). Como em MaOPs as soluções frequentemente são não-dominadas entre si, o critério para seleção dos indivíduos ou partículas é baseado quase que exclusivamente pela informação de densidade que privilegia sobretudo soluções em áreas dispersas no espaço de busca. Desta forma, MOEAs e MOPSOs baseados em dominância de Pareto podem apresentar um comportamento divergente ao invés de convergente para Frente de Pareto quando aplicados em problemas com muitos objetivos (MISHRA; DEHURI; CHO, 2015; PURSHOUSE, 2003).

Além do problema mencionado anteriormente que afeta os MOEAs e MOPSOs

baseados em dominância de Pareto, MaOPs colocam outros desafios, tais como o aumento do número de soluções necessárias para aproximar a Frente de Pareto com uma dada resolução; e a questão da visualização das soluções que pode levar a dificuldades na escolha da solução final e também no projeto de algoritmos para lidar com esses problemas (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008).

Diante dos desafios impostos pelos problemas com muitos objetivos aos MOEAs e MOPSOs tradicionais, e de sua importância uma vez que aparecem naturalmente em problemas práticos do mundo real, pesquisadores começaram a propor estratégias para contornar as dificuldades inerentes a esses problemas. Entre as primeiras delas estão a proposição de novas formas relaxadas de dominância e métodos de agregação de objetivos que permitem uma discriminação mais fina entre as soluções da população (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008; LI et al., 2015a). Contudo, essas abordagens têm a desvantagem de favorecer a convergência em comprometimento à diversidade de modo que equilibrar convergência e diversidade constituem atualmente o principal desafio no projeto de novos algoritmos para problemas com muitos objetivos (LI et al., 2015a). Além dessas, existem outras abordagens mas que também apresentam limitações de custo computacional ou de aplicação (LI et al., 2015a). A partir de 2012, começaram a surgir algoritmos evolucionários que utilizavam um conjunto de pontos de referência para permitir uma estimativa de densidade privilegiando sempre soluções próximas a esses pontos de referência enquanto que a convergência era promovida pelo uso da dominância de Pareto (DEB; JAIN, 2012; DEB; JAIN, 2014). Esses algoritmos obtiveram sucesso no sentido de aliar convergência e diversidade e são atualmente uma das abordagens mais promissoras em se tratando de otimização com muitos objetivos.

1.1 Motivação e Objetivo

Algoritmos Evolucionários ou *Evolutionary Algorithms* (EAs), em inglês, têm sido profundamente investigados no contexto da otimização de muitos objetivos. E, além disso, muitos algoritmos desse paradigma foram propostos para tentar resolver esses problemas (LI et al., 2015a). Apesar dos avanços na proposição de EAs para otimização de muitos objetivos, muito pouco tem sido feito no sentido de se adaptar algoritmos baseados em Inteligência de Enxames para problemas com muitos objetivos. Algoritmos de Inteligência de Enxames (MISHRA; DEHURI; CHO, 2015; PIOTROWSKI et al., 2017), ou *Swarm Intelligence* (SI), em inglês, são algoritmos que se inspiram no comportamento social presente em vários organismos vivos para realizar o processo de otimização e que têm se destacado na solução de problemas complexos por meio de mecanismos simples.

A deficiência no número de adaptações de algoritmos de SI na otimização de problemas com muitos objetivos pode ser explicada pela seguinte razão. Diferente dos EAs,

os vários tipos de algoritmos de SI não compartilham um arcabouço comum de operadores como acontece no paradigma evolucionário em que operadores como cruzamento ou mutação estão de alguma forma ou de outra presentes (Coello Coello; LAMONT; VELDHUIZEN, 2007). Basicamente, os vários tipos de EAs se diferenciam na forma de representação da solução do problema (ENGELBRECHT, 2007; PURSHOUSE, 2003). De fato, apesar do termo algoritmo evolucionário ser empregado nesta tese, os algoritmos evolucionários estudados aqui são essencialmente algoritmos genéticos. A heterogeneidade dos algoritmos de SI, por sua vez, torna difícil tratá-los como um corpo comum de maneira que não se pode aplicar princípios e inovações presentes em um tipo de algoritmo em outro tipo de algoritmo de inteligência de enxames de modo transparente e simples como acontece no paradigma evolucionário (MISHRA; DEHURI; CHO, 2015).

Diante da heterogeneidade e singularidade dos vários tipos de algoritmos de SI (MISHRA; DEHURI; CHO, 2015; ALAM et al., 2014; BONABEAU; MEYER, 2001; MARTENS; BAESSENS; FAWCETT, 2011; ENGELBRECHT, 2006; ENGELBRECHT, 2007), e também em virtude desse paradigma ainda ser pouco explorado no contexto de otimização de muitos objetivos, existe ainda a necessidade de adaptá-los para problemas dessa natureza (MISHRA; DEHURI; CHO, 2015), sobretudo para se avaliar suas potencialidades e limitações em cenários como esse, e com isso poder comparar a utilização dessas abordagens com o paradigma evolucionário nesses problemas. Esse fato foi a motivação geral para o desenvolvimento desta tese.

Alguns exemplos de algoritmos de SI são a Otimização por Colônia de Formiga (DORIGO, 1992; DORIGO; BLUM, 2005), a Otimização por Enxame de Partículas (KENNEDY; EBERHART, 1995), a Colônia de Abelhas Artificiais (KARABOGA, 2005; KARABOGA; BASTURK, 2007; KARABOGA; BASTURK, 2008), Algoritmos dos Morcegos (GANDOMI et al., 2013), Algoritmo de Otimização por Bactérias Farejantes (PASSINO, 2002), Otimização por Colônia de Cupins (HEDAYATZADEH et al., 2010) e Otimização por Enxame de Vespas (RUNKLER, 2008), e a Busca Por Cardumes de Peixes (FILHO et al., 2008; BASTOS-FILHO; aES, 2015). Contudo, apesar dessa grande quantidade de algoritmos, três têm se popularizado e se sobressaem com relação aos demais, são eles: a Otimização por Enxame de Partículas, ou *Particle Swarm Optimization* (PSO) em inglês, a Otimização por Colônia de Formigas, ou *Ant Colony Optimization* (ACO) em inglês, e a Otimização por Colônia de Abelhas Artificiais, ou *Artificial Bee Colony* (ABC) em inglês (MISHRA; DEHURI; CHO, 2015), sendo que o ACO e PSO são os algoritmos seminais em Inteligência de Enxames. Desta forma, esses algoritmos, por serem amplamente reconhecidos na literatura de SI, se constituem, de certa forma, em algoritmos representantes do paradigma de Inteligência de Enxames (MISHRA; DEHURI; CHO, 2015).

Dessas três abordagens de SI (ACO, ABC, PSO), o PSO, até onde vai o conhecimento do autor, foi o primeiro e continua sendo o único algoritmo pertencente ao

paradigma de SI a ser adaptado de maneira apropriada para lidar com Problemas com Muitos Objetivos ou *Many-Objective Optimization Problems* (MaOPs) em inglês.

Uma explicação plausível para o foco dado na literatura ao PSO no contexto da otimização de muitos objetivos, se deve ao fato de que o PSO já foi aplicado de maneira bem sucedida em vários problemas de otimização de natureza contínua e já tiveram seu comportamento profunda e teoricamente estudado (ENGELBRECHT, 2007; Van Den Bergh; ENGELBRECHT, 2006; Van Den Bergh, 2006; BRATTON; KENNEDY, 2007), e, por tanto, ele é um algoritmo relativamente bem estabelecido e maduro na comunidade de SI. O mesmo não pode ser dito sobre os algoritmos baseados em ACO que foram desenvolvidos principalmente para lidar com problemas de otimização combinatorial discreta, como achar o menor caminho em um grafo por exemplo (ENGELBRECHT, 2006), e, por tanto, não é um paradigma que possa ser aplicado de modo direto e natural em cenários que envolvam a otimização de problemas multiobjetivos e com muitos objetivos contínuos. E o mesmo também não pode ser dito quanto aos algoritmos mais recentes de SI como o ABC, por exemplo. De fato, provavelmente a primeira aplicação do algoritmo ABC em problemas com muitos objetivos foi publicada apenas recentemente (2017) e enquanto esta tese estava sendo produzida (LUO et al., 2017). Contudo, esse algoritmo não apresentou mecanismos promissores no contexto de otimização de multiobjetivos e além disso seus resultados se mostraram limitados em cenários com muitos objetivos sobretudo para problemas com muitas Frente de Pareto locais.

Além disso, o PSO é particularmente adequado para problemas multiobjetivos e possivelmente para problemas com muitos objetivos sobretudo pela alta velocidade de convergência que esses algoritmos apresentam em problemas com um único objetivo (JR.; SANTANA; POZO, 2016; COELLO; PULIDO; LECHUGA, 2004). Mais ainda, o núcleo do mecanismo de atualização do PSO se baseia em somente duas equações simples de atualização e o processo de atualizar as partículas por iteração é rápido comparado ao mecanismo de reprodução (cruzamento e mutação) que é computacionalmente custoso em um algoritmo evolucionário típico (ABRAHAM; JAIN; GOLDBERG, 2005). Desta forma, tendo tudo isso em vista, o PSO se tornou a porta de entrada nos estudos de adaptação de algoritmos de SI para problemas de otimização com muitos objetivos. E, pelas mesmas razões apresentadas, o PSO foi o paradigma escolhido dentre os muitos outros algoritmos de SI para ser o foco desta tese no contexto de MaOPs.

Mesmo havendo adaptações e estudos do PSO em problemas com muitos objetivos, estes estudos ainda são incipientes e as adaptações propostas apresentam várias deficiências e limitações (JR.; SANTANA; POZO, 2016). Os poucos trabalhos existentes na literatura no sentido de aplicar o PSO em MaOPs apresentam dificuldades tais como parâmetros difíceis de ajustar (DE CARVALHO; POZO, 2012), dificuldades de convergência para Frente de Pareto (DE CARVALHO; POZO, 2012; KACHROUDI; GROSSARD, 2010), e necessidades

de informação de preferência (WICKRAMASINGHE; LI, 2009). Mais ainda, algumas dessas adaptações, apesar de conseguir uma boa convergência, apresentam uma diversidade muito reduzida de maneira que o conjunto aproximação produzido por elas apresentam soluções de pequena parte da Frente de Pareto. Esse é o caso por exemplo do algoritmo MOPSO-GD (FIGUEIREDO, 2013) um algoritmo que foi desenvolvido pelo autor desta tese em parceria com seus orientadores, a saber, Carmelo José Albanez Bastos-Filho e Teresa Bernarda Ludermir e que foi publicado na dissertação intitulada "*Algoritmo Baseado em Enxame de Partículas para Otimização de Problemas com Muitos Objetivos*" (FIGUEIREDO, 2013) e que foi defendida em fevereiro de 2013 na Universidade Federal de Pernambuco.

O MOPSO-GD é um algoritmo baseado em um método de agregação denominado Detrimento Global ou *Global Detriment Ranking* (GDR), em inglês, que agrega o vetor objetivo de uma solução em um escalar e com isso ele consegue estabelecer uma discriminação mais fina entre as soluções. O GDR é um método de agregação proposto por Garza-Fabre, Pulido e Coello-Coello em (Garza Fabre; PULIDO; Coello Coello, 2009; GARZA-FABRE; PULIDO; Coello Coello, 2010; Garza Fabre, 2009). Ele é um método que permite uma discriminação bastante fina das soluções e que por isso mesmo favorece uma boa convergência das soluções para a Frente de Pareto. De fato, mesmo atualmente (2017), o GDR tem inspirado o desenvolvimento de novas estratégias de atribuição de aptidão como o artigo de Wanliang e Wang publicado em (LI; WANG; XU, 2017). Contudo, apesar da sua excepcional convergência, como demonstra os resultados publicados em (FIGUEIREDO, 2013), o MOPSO-GD apresenta um grave problema de diversidade pois o conjunto aproximação produzido por ele geralmente consiste de soluções concentradas em uma pequena região da Frente de Pareto.

Sendo assim, o desenvolvimento de um algoritmo baseado em PSO capaz de unir convergência e diversidade ainda é um problema importante e em aberto cujo estudo ainda merece muita atenção e inovação. Neste sentido, o objetivo desta tese é desenvolver um novo algoritmo baseado em PSO para problemas com muitos objetivos de modo a unir convergência, diversidade e uma boa cobertura da Frente de Pareto, requisitos que constituem um grande desafio em cenários com muitos objetivos (JR.; SANTANA; POZO, 2016).

Para alcançar esse objetivo, buscou-se aplicar no contexto da otimização por enxame de partículas uma abordagem que tem se mostrado bastante promissora quando aplicada em algoritmos evolucionários e que consiste em usar um conjunto de pontos de referência (DEB; JAIN, 2012; DEB; JAIN, 2014; LI et al., 2015a; ASAFUDDOULA; RAY; SARKER, 2015; CHENG; YEN; ZHANG, 2015; CAI et al., 2015; LI et al., 2015b; CHENG et al., 2016a; YUAN et al., 2016; XIANG et al., 2017). A ideia de usar um conjunto de pontos de referência é permitir uma estimação da densidade das soluções encontradas até o momento sem com isso prejudicar a convergência. Ou seja, esse método de estimação de densidade

usando pontos de referência tende a favorecer soluções possivelmente bem distribuídas ao longo de toda a Frente de Pareto ao mesmo tempo em que, por sua vez, penaliza soluções *resistentes a dominação* (PURSHOUSE, 2003). Isto é, soluções com valores muito bons em alguns objetivos mas valores extremamente ruins em outros objetivos e que, por isso, tendem a ser não-dominadas e ao mesmo tempo bastante dispersas no espaço objetivo e frequentemente distantes da Frente de Pareto (PURSHOUSE, 2003). Além disso, esses pontos de referência podem servir de pontos de atração para promover a convergência de soluções para Frente de Pareto. O algoritmo proposto foi denominado de Algoritmo de Otimização por Enxame de Partículas para Muitos Objetivos ou *Many-Objective Particle Swarm Optimization* (MaOPSO) em inglês. O MaOPSO também apresenta outras características tais como o uso de sub-enxames para realizar pesquisas mais refinadas pelos extremos da Frente de Pareto e assim permitir uma maior capacidade de cobertura e convergência para a Frente de Pareto. Outra inovação também é o uso de uma seleção por torneio que fornece uma maior pressão de seleção pois o número de concorrentes depende do número de objetivos. Desta maneira, quanto maior o número de objetivos maior é o número de soluções concorrentes a se tornarem líderes e com isso uma maior pressão de seleção é exercida em direção a Frente de Pareto. Como se pode perceber, o MaOPSO consiste em um algoritmo mais inovador que a proposta desenvolvida no mestrado do autor desta tese, o MOPSO-GD. Diferente do MOPSO-GD, que simplesmente utiliza um mecanismo que consiste em atribuir um escalar a uma solução pela comparação com outras soluções do arquivo externo, o MaOPSO atribui dois valores a cada solução do arquivo externo, a saber, uma medida de convergência e uma medida de densidade, que são calculadas com base em pontos de referência, que são utilizadas para determinar a busca de soluções Pareto-ótima em direções pré-determinadas e com o objetivo de espalhar essas soluções Pareto-ótimas o mais uniformemente possível.

Em suma, o objetivo de pesquisa desta tese é investigar o efeito de se aplicar *pontos de referência* no processo de atribuição de aptidão de um algoritmo de enxame de partículas em termos de *convergência* e *diversidade* no contexto da otimização de *problemas com muitos objetivos*. Sendo assim, a hipótese de pesquisa desta tese é a de que os pontos de referência possam permitir uma discriminação mais fina das soluções entre as soluções não-dominadas encontradas durante o processo de busca em cenários com muitos objetivos de maneira a privilegiar tanto a convergência como a promoção da diversidade. E, desta forma, por meio dessa discriminação das soluções, permitir a convergência do algoritmo de enxame de partículas para a Frente de Pareto ao mesmo tempo em que se busca garantir a cobertura e a diversidade da Frente de Pareto.

Por fim, a partir de tudo o que foi exposto acima, as contribuições desta tese para a área de otimização de muitos objetivos são:

1. proposição do uso de uma abordagem de *pontos de referência* no contexto de um

algoritmo de enxame de partículas.

2. proposição de um esquema inovador para discriminação das soluções do arquivo externo usando duas medidas que são calculadas a partir dos pontos de referência, a saber, a *medida de convergência* e a *medida de densidade*. Essas medidas são avaliadas de modo hierárquico, sendo que a medida de densidade é primeiro avaliada e, no caso em que duas soluções tem a mesma medida, a medida de convergência é usada como segundo critério;
3. proposição de dois esquemas distintos de atribuição dos líderes sociais. Nesse caso, o enxame é dividido em dois sub-enxames. O primeira sub-enxame recebe como líderes sociais as soluções que são consideradas extremas. E o segundo sub-enxame recebe líderes sociais com base nas medidas de convergência e diversidade usando torneio com o número de participantes igual ao número de objetivos.
4. proposição de seleção por torneio na seleção dos líderes sociais cujo número de concorrentes é igual ao número de objetivos. Ou seja, proposição de seleção por torneio adaptativa. Isso permite que se aumente a pressão de seleção do algoritmo a depender do número de objetivos. Como, à medida que o número de objetivos aumenta, existe um aumento exponencial de soluções não-dominadas, esse mecanismo pode impor uma maior pressão de seleção para a Frente de Pareto. Essa ideia é um dos componentes chaves do MaOPSO.

1.2 Organização da Tese

Esta tese está organizada como segue. O Capítulo 2 apresenta a revisão da literatura em que são apresentados os conceitos básicos e os fundamentos do problema e algoritmos tratados nesta tese. O Capítulo 3 apresenta o algoritmo proposto pelo autor da tese, o MaOPSO. O Capítulo 4 apresenta o arranjo experimental e os resultados obtidos. Nesse capítulo, o MaOPSO foi avaliado por meio do uso seis problemas de teste da família DTLZ em termos de sua capacidade de convergência e de preservação da diversidade. Também neste capítulo, o MaOPSO foi avaliado quanto a sua utilização prática por meio de sua aplicação em um problema real de engenharia envolvendo o projeto de redes ópticas de alta capacidade. Finalmente, o Capítulo 5 apresenta a conclusão e as sugestões de trabalhos futuros.

2 Conceitos Básicos

Neste capítulo, são fornecidos os fundamentos para se compreender o restante desta tese. Tendo isso em vista, os conceitos básicos e fundamentos sobre otimização multiobjetivo, algoritmos evolucionários multiobjetivos, otimização multiobjetivo por enxame de partículas, e sobre otimização de problemas com muitos objetivos são apresentados de modo sucinto nas próximas seções.

2.1 Otimização Multiobjetivo

Um problema de otimização multiobjetivo ou *Multiobjective Optimization Problem* (MOP) em inglês pode ser definido sem perda de generalidade como a minimização simultânea de duas ou mais funções objetivos, em que geralmente essas funções estão em conflito entre si. Essas funções objetivos podem ser representadas por um vetor de funções \mathbf{f} que mapeia um vetor de n parâmetros (variáveis de decisão) para um vetor com m funções objetivo, com $m \geq 2$. Formalmente, esse problema pode ser definido, sem perda de generalidade ¹, como:

$$\text{minimizar } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}, \quad (2.1)$$

sujeito à

$$\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}) \ h_2(\mathbf{x}) \ \dots \ h_p(\mathbf{x})]^T = \mathbf{0}, \quad (2.2)$$

e

$$\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}) \ g_2(\mathbf{x}) \ \dots \ g_q(\mathbf{x})]^T \leq \mathbf{0}, \quad (2.3)$$

em que $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T \in \mathbb{R}^n$ é o vetor de variáveis de decisão ou de projeto, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$ são as funções objetivo e $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ são, respectivamente, as inequações e equações que modelam as restrições do problema. Os vetores $\mathbf{y} = \mathbf{f}(\mathbf{x}) \ \forall \mathbf{x} \in \mathbb{R}^n$ são chamados de vetores objetivo. O conjunto \mathbb{R}^n é chamado de espaço de variáveis de decisão ou espaço de decisão e o conjunto \mathbb{R}^m é chamado de espaço objetivo. O conjunto factível $\Omega \subset \mathbb{R}^n$ é definido como o conjunto de vetores de decisão \mathbf{x} que obedecem às restrições g_i, h_j para $i = 1, 2, \dots, p$ e $j = 1, 2, \dots, q$. Isto é,

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \text{ e } \mathbf{h}(\mathbf{x}) = \mathbf{0}\}. \quad (2.4)$$

¹ Todas as considerações, definições, equações e observações que serão apresentadas daqui em diante irão assumir a minimização de todas as funções objetivo.

A imagem de Ω no espaço objetivo é denominada de região factível e é denotada por \mathcal{O} . A Figura 1 ilustra a região factível de um problema multiobjetivo com dois objetivos. A Figura 1 também apresenta todos os conceitos já apresentados. Na Figura 1, o gráfico da esquerda mostra o espaço de decisão \mathbb{R}^n com a região factível Ω . Nessa figura, o gráfico da direita ilustra o espaço objetivo. Essa figura também apresenta o mapeamento entre um vetor no espaço de decisão para um vetor objetivo.

Uma vez definido o que é um problema multiobjetivo, resta agora definir o significado da palavra minimizar no contexto de um problema de otimização multiobjetivo. Para isso, algumas definições são apresentadas a seguir.

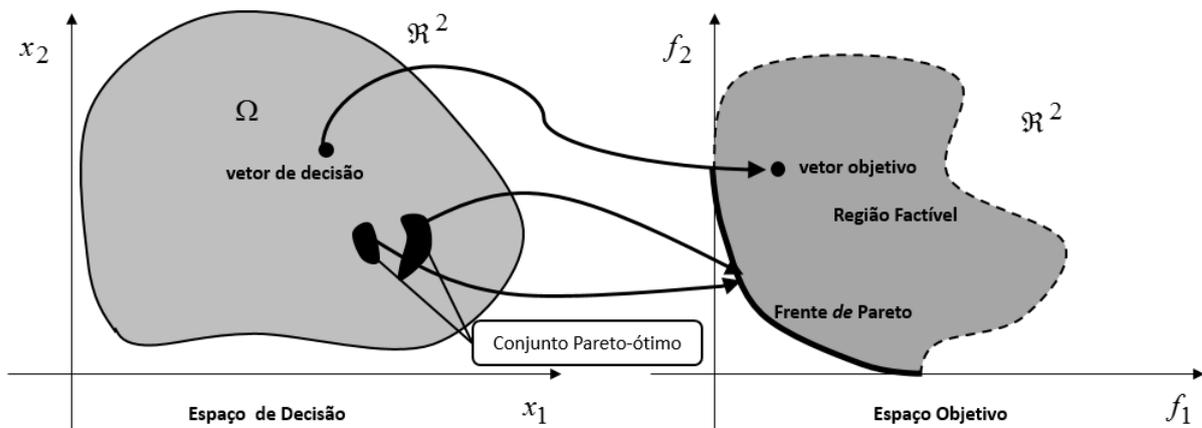


Figura 1 – Relação entre um espaço de decisão bidimensional e um espaço objetivo bidimensional (Figura adaptada de (FIGUEIREDO, 2013)).

Definição 2.1 (Dominância de Pareto). *Dados dois vetores $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$, diz-se que o vetor \mathbf{u} domina o vetor \mathbf{v} , denotado por $\mathbf{u} \prec \mathbf{v}$, se e somente se, $u_i \leq v_i, \forall i \in \{1, 2, \dots, m\}$ e se $\exists i \in \{1, 2, \dots, m\}$ tal que $u_i < v_i$ (AZEVEDO, 2011)(ZITZLER et al., 2003).*

Definição 2.2 (Dominância Forte de Pareto). *Dados dois vetores $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$, diz-se que o vetor \mathbf{u} domina estritamente (fortemente) o vetor \mathbf{v} , denotado por $\mathbf{u} \prec\prec \mathbf{v}$, se e somente se, $u_i < v_i, \forall i \in \{1, 2, \dots, m\}$ (ZITZLER et al., 2003).*

Definição 2.3 (Soluções incomparáveis). *Dados dois vetores $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$, diz-se que o vetor \mathbf{u} é incomparável ao vetor \mathbf{v} , denotado por $\mathbf{u} \parallel \mathbf{v}$, se e somente se, \mathbf{u} não domina fracamente \mathbf{v} e \mathbf{v} não domina fracamente \mathbf{u} (ZITZLER et al., 2003). Nesta tese, duas soluções incomparáveis também serão chamadas de não-dominadas entre si.*

A Figura 2 apresenta a interpretação geométrica das possíveis relações de dominância existentes entre as soluções no espaço objetivo bidimensional. Uma solução A divide esse espaço em quatro regiões retangulares. As soluções na região retangular acima e a direita (região em em cinza claro com triângulos) de A representam as soluções dominadas

pela solução A . As soluções que estão na região retangular abaixo e a esquerda (região em cinza escuro com quadrados) de A representam as soluções que dominam a solução A . Finalmente, as soluções que estão nas duas regiões retangulares restantes (região em branco com círculos em cinza) são soluções incomparáveis a solução A . A solução C é um exemplo de uma solução fortemente dominada por A , enquanto a solução B é um exemplo de uma solução dominada por A mas que não é dominada fortemente por ela. A solução A é uma exemplo de uma solução dominada fortemente por D .

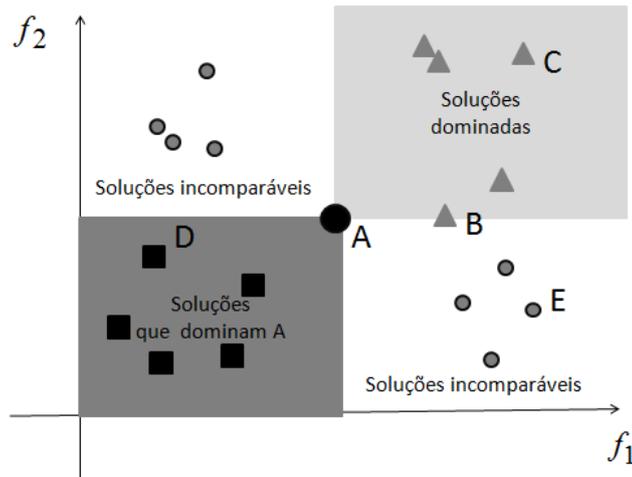


Figura 2 – Relações possíveis entre as soluções em um espaço objetivo bidimensional (figura adaptada de (ZITZLER, 1999)).

Definição 2.4 (Otimalidade de Pareto). Um vetor de variáveis de decisão $\mathbf{x}^* \in \Omega \subset \mathbb{R}^n$ é uma solução Pareto-ótima, ou uma solução eficiente, se e somente se, $\nexists \mathbf{x} \in \Omega$ tal que $\mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}^*)$.

Definição 2.5 (Conjunto Pareto-ótimo). O conjunto Pareto-ótimo Ω^* é definido por:

$$\Omega^* = \{\mathbf{x} \in \Omega \mid \mathbf{x} \text{ é Pareto-ótimo}\}. \tag{2.5}$$

Definição 2.6 (Frente de Pareto global). A Frente de Pareto global \mathcal{PF}^* ou simplesmente Frente de Pareto é definida por:

$$\mathcal{PF}^* = \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m \mid \mathbf{x} \in \Omega^*\}. \tag{2.6}$$

A Figura 1 ilustra os conceitos de conjunto Pareto-ótimo e Frente de Pareto, bem como a relação existente entre esses dois conceitos.

A Figura 3 apresenta exemplos de uma Frente de Pareto global e local em um espaço objetivo bidimensional. A linha tracejada consiste em uma Frente de Pareto local, enquanto a linha sólida retrata uma Frente de Pareto global. A definição formal de Frente

de Pareto local é um pouco complexa e é deixada de lado nesta tese. Para efeitos de explicação e argumentação ao longo desta tese é suficiente uma definição intuitiva do que consiste uma Frente de Pareto local. Um conjunto de soluções é considerado uma Frente de Pareto local se ela é formado por apenas soluções que são incomparáveis entre si, ou seja, essas soluções são não-dominadas entre si, e se, além disso, cada uma dessas soluções não é dominada por quaisquer soluções dentro de uma determinada vizinhança. Por exemplo, as soluções G e H pertencentes a Frente de Pareto local na Figura 3 são incomparáveis entre si, isto é, essas duas soluções são não-dominadas entre si. Além disso, a solução G não é dominada por nenhuma solução dentro de sua vizinhança (o círculo na Figura 3). Esse último fato faz com a solução G seja um exemplo de uma solução localmente não-dominada e por isso pertencente a Frente de Pareto local da Figura 3. Finalmente, F é um exemplo de uma solução dominada por G e que, portanto, está fora da Frente de Pareto local da Figura 3. Por fim, é importante enfatizar, que a solução A que pertence a Frente de Pareto global é capaz de dominar todas as soluções da Frente de Pareto local. Finalmente, as soluções C e D são dois exemplos de soluções que são dominadas pela solução B da Frente de Pareto global mas que não são dominadas por nenhuma das soluções da Frente de Pareto local e nem pertencem a ela como é ilustrado na figura. Para uma definição formal de Frente de Pareto local, o leitor é convidado a ler a tese de Eckart Zitzler (ZITZLER, 1999).

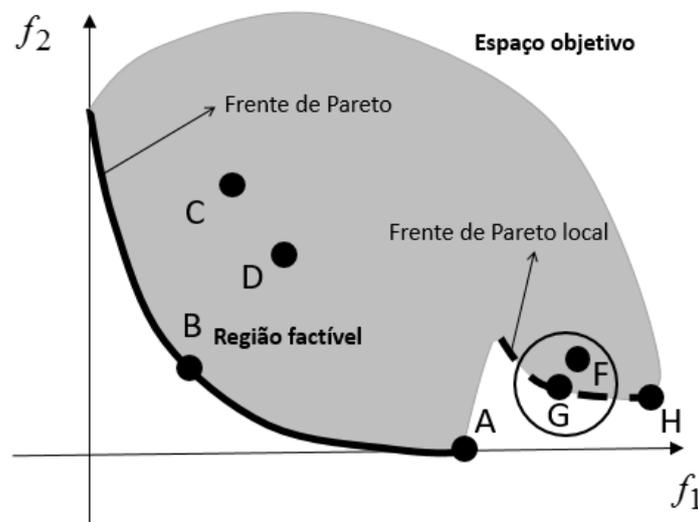


Figura 3 – Ilustração de uma Frente de Pareto global e local em espaço objetivo bidimensional (figura adaptada de (FIGUEIREDO, 2013)).

Uma vez definidos os conceitos sobre otimização multiobjetivo, pode-se agora definir no que consiste resolver um problema de otimização multiobjetivo.

Definição 2.7 (Solução de um problema multiobjetivo). Resolver um MOP consiste em determinar o conjunto Pareto-ótimo Ω^* a partir do conjunto Ω .

Outro conceito importante que envolve a definição de dominância é o de solução localmente não-dominada. Segundo este conceito, uma solução é dita localmente não-dominada com respeito a um dado conjunto S de soluções na qual ela faz parte se ela for não-dominada com respeito a cada uma das soluções do conjunto S com exceção dela mesma. Este conceito é importante porque ele permite definir no que consiste a resolução de um problema multiobjetivo por intermédio de um algoritmo. Formalmente, o conceito de solução localmente não-dominada (PURSHOUSE, 2003) é definida segundo a Definição 2.8.

Definição 2.8 (Solução localmente não-dominada). *Uma solução $\mathbf{s} \in S$ (vetor no espaço objetivo) é dita ser localmente não-dominada com respeito ao conjunto S de soluções, se e somente se $\nexists \mathbf{x} \in S$ tal que $\mathbf{x} \prec \mathbf{s}$.*

É importante mencionar que uma solução é sempre não-dominada com respeito a um determinado conjunto. Outra definição importante é o de conjunto não-dominado (KHARE, 2002):

Definição 2.9 (Conjunto não-dominado). *Um conjunto \mathcal{N} é dito ser um conjunto não-dominado com respeito a um conjunto S de soluções, se \mathcal{N} contém todas as soluções localmente não-dominadas de S (ZITZLER, 1999; KHARE, 2002). Ou seja,*

$$\mathcal{N} = \{s \mid s \text{ é localmente não-dominado com relação à } S\}. \quad (2.7)$$

Nesta tese, quando se mencionar apenas que um conjunto \mathcal{N} é um conjunto não-dominado isto significa que todas as suas soluções são localmente não-dominadas com relação a ele mesmo.

O conjunto Pareto-ótimo pode conter um número enorme ou até mesmo infinito de soluções Pareto-ótimas. Sendo assim, a tarefa de um algoritmo que se proponha a resolver um problema multiobjetivo deve consistir em fornecer uma amostra desse conjunto devido a limitações de recursos computacionais. O conjunto de soluções produzido (retornado) por um algoritmo de otimização multiobjetivo que consiste numa suposta amostra da Frente de Pareto é denominado de *conjunto aproximação* ou *conjunto solução*. Formalmente:

Definição 2.10 (Conjunto aproximação). *O conjunto aproximação denotado por $PF \subseteq \mathcal{O}$ é um conjunto não-dominado de vetores objetivos retornado por algum algoritmo de otimização multiobjetivo e que consiste em uma suposta representação da Frente de Pareto (Coello Coello; LAMONT; VELDHUIZEN, 2007; Garza Fabre, 2009; ZITZLER et al., 2003).*

É necessário mencionar que o conjunto aproximação de algoritmo multiobjetivo deve conter apenas soluções não-dominadas entre si, mas essas soluções não precisam ser necessariamente Pareto-ótimas. No entanto, um bom algoritmo deve fornecer um conjunto aproximação que contenha apenas soluções sobre a Frente de Pareto do problema e que

estas soluções estejam bem distribuídas ao longo da Frente de Pareto. Em outras palavras, o conjunto aproximação produzido por um algoritmo deve obedecer a dois requisitos básicos² para que possa ser considerado uma boa representação da Frente de Pareto (ZITZLER, 1999; LI et al., 2015b; KHARE, 2002; LI et al., 2012). São eles:

- **Proximidade.** O conjunto aproximação PF deve conter apenas vetores objetivo próximos da Frente de Pareto. Idealmente, todos os seus vetores objetivo devem pertencer à Frente de Pareto;
- **Diversidade.** O requisito de diversidade normalmente diz respeito às soluções no espaço objetivo. No espaço objetivo, o conjunto aproximação PF deve conter uma boa distribuição de soluções em termos de extensão (cobertura) e uniformidade, ou seja, o conjunto aproximação deveria cobrir toda Frente de Pareto e apresentar soluções uniformemente espalhadas sobre ela.

A Figura 4 ilustra um conjunto aproximação gerado por um algoritmo multiobjetivo ideal. Nessa figura, todas as soluções estão sobre a Frente de Pareto. Além disso, elas cobrem toda a extensão da Frente de Pareto e estão uniformemente distribuídas ao longo dela.

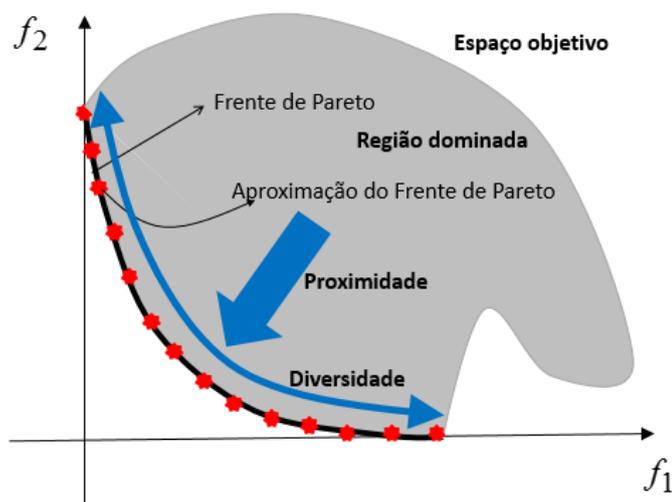


Figura 4 – Requisitos desejáveis do conjunto aproximação (conjunto solução) de um algoritmo multiobjetivo (figura retirada de (FIGUEIREDO, 2013)).

² Em (PURSHOUSE, 2003), Purshouse destacou um outro aspecto além da proximidade e diversidade que é a pertinência. Pertinência consiste em avaliar se as soluções do conjunto aproximação estão na região de interesse do tomador de decisão. Nesta tese apenas a proximidade e a convergência são consideradas pois em todos os problemas tratados nesta tese é assumido que o tomador não tem nenhuma preferência por alguma região específica da Frente de Pareto.

2.2 Otimização Evolucionária Multiobjetivo

Um Algoritmo Evolucionário ou *Evolutionary Algorithm* (EA) em inglês é qualquer heurística inspirada no processo de seleção natural e que é aplicada na resolução de problemas de otimização. EAs emulam os fenômenos de cruzamento, mutação e sobrevivência do mais apto presentes na natureza para desenvolver um conjunto inicial de soluções até que elas possivelmente convirjam para uma solução próxima da ótima ou eventualmente ótima.

Devido ao fato de utilizarem uma população de soluções candidatas, EAs são particularmente interessantes na solução de problemas multiobjetivos em que se busca encontrar um conjunto de soluções simultaneamente em uma única execução do algoritmo, ao invés de ter de realizar uma série de execuções separadas, como no caso de técnicas de programação matemática tradicionais (ABRAHAM; JAIN; GOLDBERG, 2005). Além disso, EAs são menos suscetíveis a forma ou continuidade da Frente de Pareto, enquanto técnicas de programação matemática são mais sensíveis a essas questões (Coello Coello; LAMONT; VELDHUIZEN, 2007). Quando EAs são aplicados à resolução de problemas multiobjetivos eles recebem o nome especial de Algoritmos Evolucionários Multiobjetivo ou *Multiobjective Evolutionary Algorithms* (MOEAs) em inglês.

O Algoritmo 1 apresenta o pseudo-código de um MOEA elitista básico (LI et al., 2015b; CHENG; YEN; ZHANG, 2015; XIANG et al., 2017; Coello Coello; LAMONT; VELDHUIZEN, 2007). Como se pode perceber, um MOEA é estruturalmente similar a um algoritmo evolucionário mono-objetivo, como é declarado por Coello Coello, Lamont, e Van Veldhuizen (Coello Coello; LAMONT; VELDHUIZEN, 2007). Sendo assim, o algoritmo começa inicializando a população no espaço de busca segundo uma distribuição aleatória. Em seguida, o algoritmo realiza uma *seleção-para-variação* sobre a população de indivíduos de modo a escolher os indivíduos que participarão do processo de reprodução, e com isso deixarão descendentes para a próxima geração. Para isso, antes que essa seleção ocorra é necessário realizar a *atribuição de aptidão* aos indivíduos da população $P^{(t)}$ de modo que cada indivíduo tenha uma aptidão e possa competir com os outros indivíduos da população na fase de seleção para variação. Nesse processo de seleção, indivíduos mais aptos têm mais chances de serem selecionados e uma vez selecionados integram a população $M^{(t)}$. Após essa etapa, realiza-se a reprodução e mutação dos indivíduos em $M^{(t)}$ na fase de *variação* para formar uma população de descendentes $Q^{(t)}$. Novamente, realiza-se a *atribuição de aptidão* mas nesta etapa somente para $Q^{(t)}$, visando que esses indivíduos possam participar da etapa de *seleção-para-sobrevivência*. Na etapa de *seleção-para-sobrevivência*, a população original é combinada com a de descendentes para que se possa selecionar, de modo geralmente determinístico, os melhores indivíduos dessa população combinada para formar uma nova população $P^{(t+1)}$ para a próxima geração. Esse esquema é repetido até que um critério de parada seja atingido, geralmente um número máximo de gerações (iterações) ou até que não haja uma mudança significativa na população que aproxima a Frente de Pareto. Como

já foi mencionado, um MOEA é estruturalmente similar a um algoritmo evolucionário mono-objetivo, a maior diferença reside na forma como a atribuição de aptidão é realizada. Enquanto em um algoritmo evolucionário, em geral, a aptidão de um indivíduo é seu valor na função-objetivo ou pode ser calculada a partir de uma transformação matemática desse valor (ZITZLER et al., 2003); em um MOEA, a aptidão de um indivíduo pode envolver a comparação do vetor objetivo desse indivíduo com os vetores objetivos do resto da população ou uma transformação desse vetor objetivo em um valor escalar. Ou seja, um MOEA requer necessariamente um passo de processamento a mais que um algoritmo evolucionário que consiste na transformação do vetor objetivo de cada indivíduo em um valor escalar que lhe serve como medida de aptidão (Coello Coello; LAMONT; VELDHUIZEN, 2007). De fato, um dos pontos-chave de qualquer MOEA é o método de atribuição de aptidão de modo que esses algoritmos se diferenciam basicamente na maneira como essa atribuição é realizada. Os mais promissores e atuais MOEAs utilizam o conceito de dominância de Pareto para qualificar as soluções, de modo a favorecer soluções não-dominadas sobre as dominadas (LI; WANG; XU, 2017; ZHOU et al., 2011; Coello Coello; LAMONT; VELDHUIZEN, 2007). Além disso, eles aplicam técnicas de estimação de densidade, de modo a privilegiar soluções em espaços menos densamente povoados sobre aquelas em regiões mais densamente povoadas por soluções. Assim, eles evitam que o algoritmo convirja para um único ponto. Em outras palavras, quando soluções apresentam a mesma aptidão segundo o critério de dominância de Pareto, a solução que estiver na região menos densamente povoada é considerada mais apta.

Algoritmo 1: Pseudo-código de um MOEA básico.

```

1 Inicialize a população  $P^{(0)}$ ;
2  $t = 0$ ;
3 enquanto condição de parada não for alcançada faça
4   Avalie cada indivíduo de  $P^{(t)}$  usando o MOP;
5   Realize a atribuição de aptidão de cada indivíduo de  $P^{(t)}$  a partir dos vetores
   objetivos de  $P^{(t)}$  ;
6    $M^{(t)} \leftarrow$  seleção-para-variação (  $P^{(t)}$  ) ;
7    $Q^{(t)} \leftarrow$  variação (  $M^{(t)}$  ) ;
8   Avalie cada indivíduo de  $Q^{(t)}$  usando o MOP;
9   Realize a atribuição de aptidão de cada indivíduo de  $Q^{(t)}$  a partir dos vetores
   objetivos de  $Q^{(t)}$ ;
10   $P^{(t+1)} \leftarrow$  seleção-ambiental (  $P^{(t)} \cup Q^{(t)}$  ) ;
11 retorna  $P^{(t)}$ 

```

2.3 Otimização Multiobjetivo por Enxames de Partículas

O algoritmo de Otimização por Enxame de Partículas ou *Particle Swarm Optimization* (PSO) em inglês é uma meta-heurística proposta por Kennedy e Eberhart (KENNEDY; EBERHART, 1995) em 1995, inspirado no comportamento social e cooperativo de organismos biológicos, tais como cardumes e bandos de pássaros (BRATTON; KENNEDY, 2007). Ele é geralmente aplicado na otimização de funções não-lineares de alta dimensionalidade e com variáveis contínuas (BRATTON; KENNEDY, 2007). No PSO, cada solução candidata corresponde a uma partícula e o conjunto dessas partículas constitui um enxame de partículas em que ocorre um processo de troca de informação sobre boas regiões no espaço de busca. As partículas se movem no espaço de busca de alta dimensionalidade buscando por boas soluções, usando uma combinação de atração para a melhor solução encontrada até o momento pela própria partícula; e de uma atração para a melhor solução encontrada até o momento pelas partículas consideradas vizinhas. A vizinhança de uma partícula é o conjunto de partículas com as quais ela é capaz de trocar informações sobre boas regiões do espaço de busca já visitadas (BRATTON; KENNEDY, 2007). Cada partícula i do enxame S tem uma posição \mathbf{x}_i , que consiste de um vetor de n dimensões cujos componentes representam as variáveis de decisão da função objetivo, uma velocidade \mathbf{v}_i e uma posição \mathbf{y}_i chamada de *melhor posição pessoal* ou *pbest*, que representa a melhor posição encontrada pela partícula até um momento em consideração. Desta maneira, as partículas se movimentam no espaço de busca de acordo com duas equações:

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_{ij}(t) - x_{ij}(t)], \quad (2.8)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (2.9)$$

em que o termo $\hat{\mathbf{y}}_i(t)$ consiste na melhor posição encontrada até o momento t pelas partículas vizinhas da partícula i no enxame. Quando a vizinhança das partículas consiste no enxame inteiro, a posição $\hat{\mathbf{y}}_i(t)$ é denominada de *gbest* (melhor posição global). O termo ω é o coeficiente de inércia e os termos c_1 e c_2 são constantes reais positivas conhecidas como coeficientes de aceleração. Os termos $r_{1j}(t)$ e $r_{2j}(t)$ são valores aleatórios gerados a partir de uma distribuição de probabilidade uniforme no intervalo $[0, 1]$, ou seja, $r_{1j}(t), r_{2j}(t) \sim U[0, 1]$.

Assim como ocorre com EAs, o PSO também pode ser adaptado para resolver problemas de otimização multiobjetivo por causa de sua natureza baseada em população. Uma adaptação do PSO para problemas multiobjetivos é chamada genericamente de Algoritmo de Otimização Multiobjetivo por Enxame de Partículas ou em inglês *Multiobjective Particle Swarm Optimization* (MOPSO) (REYES-SIERRA; COELLO, 2006).

Da mesma maneira como acontece com MOEAs, um MOPSO deve atender a dois requisitos básicos: convergência e diversidade. Isso, por sua vez, não é uma tarefa trivial e adaptar um PSO para problemas multiobjetivos envolve várias decisões de projeto que devem ser tomadas com cuidado para evitar que o algoritmo forneça um conjunto pouco diverso de soluções ou com poucas soluções retornadas (REYES-SIERRA; COELLO, 2006).

O Algoritmo 2 apresenta o pseudo-código de um MOPSO genérico. Inicialmente, as partículas são distribuídas no espaço de busca segundo uma distribuição uniforme. Em seguida, as soluções não-dominadas presentes no enxame inicial são inseridas em uma memória auxiliar denominada de arquivo externo denotado aqui por $A^{(t)}$. O arquivo externo é uma estrutura de dados presentes em muitos algoritmos multiobjetivos (REYES-SIERRA; COELLO, 2006; Coello Coello; LAMONT; VELDHUIZEN, 2007; ZHOU et al., 2011) tanto da abordagem evolucionária (ZITZLER, 1999; ZITZLER; LAUMANN; THIELE, 2001; ZITZLER; THIELE, 1999; ZHOU et al., 2011) como da abordagem de enxame de partículas (REYES-SIERRA; COELLO, 2006; COELLO; PULIDO; LECHUGA, 2004; COELLO; LECHUGA, 2002; NEBRO et al., 2009; LI et al., 2012; ZHOU et al., 2011). O arquivo externo cumpre um papel muito importante em qualquer MOPSO pois é nele que são armazenadas as soluções não-dominadas com respeito a todas as soluções visitadas até o momento (REYES-SIERRA; COELLO, 2006; MEZA et al., 2017; NEBRO; DURILLO; COELLO, 2013; Coello Coello; LAMONT; VELDHUIZEN, 2007; ZHOU et al., 2011). Sendo assim, o arquivo externo é um conjunto não-dominado, isto é, todas as suas soluções são não-dominadas entre si.

O arquivo externo também tem uma segunda função. Como foi mencionado acima, no PSO típico cada partícula do enxame é guiada pelo *gbest* que é a melhor posição visitada pelo enxame até o momento. No entanto, em problemas multiobjetivos não existe um *gbest*, mas sim um conjunto de soluções em que não se pode dizer qual solução é a melhor. Essas soluções são justamente as soluções não-dominadas presentes no arquivo externo e que correspondem as melhores soluções encontradas até o momento. Na literatura de MOPSO, essas soluções são chamadas de líderes globais ou líderes sociais (REYES-SIERRA; COELLO, 2006; SANTANA; PONTES; BASTOS-FILHO, 2009; COELLO; PULIDO; LECHUGA, 2004). Nesta tese se preferiu usar o tempo líder social (SANTANA; PONTES; BASTOS-FILHO, 2009; COELLO; PULIDO; LECHUGA, 2004). Assim, uma decisão de projeto importante que surge é como selecionar um líder social para cada partícula entre as soluções do arquivo externo (REYES-SIERRA; COELLO, 2006; NEBRO; DURILLO; COELLO, 2013; ZHOU et al., 2011; Coello Coello; LAMONT; VELDHUIZEN, 2007). Como as soluções não-dominadas no arquivo externo são responsáveis por promover a convergência (REYES-SIERRA; COELLO, 2006), um critério natural para diferenciar as soluções incomparáveis do arquivo externo é usar o critério da diversidade (REYES-SIERRA; COELLO, 2006; Coello Coello; LAMONT; VELDHUIZEN, 2007; RAQUEL; NAVAL JR., 2005). Desta forma, as soluções do arquivo são geralmente diferenciadas pelo uso de um estimador de densidade em que soluções em regiões menos densamente povoadas são favorecidas (REYES-SIERRA; COELLO, 2006; MEZA et al., 2017;

RAQUEL; NAVAL JR., 2005; Coello Coello; LAMONT; VELDHUIZEN, 2007). Para evitar que o MOPSO convirja para um único ponto, cada partícula do enxame seleciona um líder social usando algum mecanismo de seleção como por exemplo seleção por torneio binário (NEBRO et al., 2009). Um exemplo de estimador de densidade é o *crowding distance* (DEB et al., 2002; NEBRO et al., 2009; RAQUEL; NAVAL JR., 2005; SANTANA; PONTES; BASTOS-FILHO, 2009) que foi proposto por Deb *et al.* (DEB et al., 2002) para ser usado no NSGA-II (DEB et al., 2002) como estimador de densidade e critério de ordenamento (classificação) de soluções não dominadas no seu processo de seleção para variação e no seu processo de seleção ambiental.

O *Crowding Distance* (CD) consiste no semi-perímetro do hipercubo conectando as soluções mais próximas da solução para qual a medida de densidade está sendo calculada. A Figura 5 ilustra a interpretação geométrica do CD em um espaço objetivo de duas dimensões. Como se pode observar pela Figura 5, as soluções adjacentes à solução i são as soluções a e b . O retângulo conectando as soluções a e b está hachurado. O CD da solução i consiste no semi-perímetro desse retângulo, isto é, na soma das distâncias d_1 e d_2 .

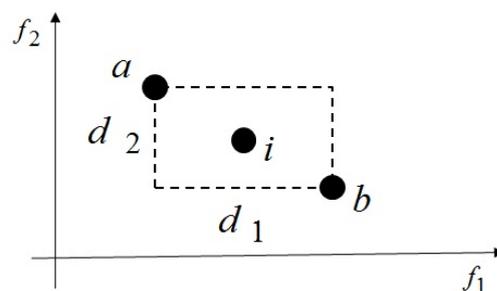


Figura 5 – Exemplo do cálculo do CD em um espaço objetivo bidimensional.

Assim, a aptidão de cada líder social consiste em alguma medida de densidade de modo que toda vez que o arquivo externo é atualizado deve-se calcular a densidade das soluções presentes neste arquivo. Uma vez definidos os líderes sociais de cada partícula, as partículas se movimentam no espaço de busca de acordo com as equações usuais do PSO ou com alguma variação dessas equações.

Feito isso, geralmente se realiza em seguida uma operação de turbulência que é uma operação similar a mutação em EAs para permitir uma maior diversidade de soluções levando as partículas para áreas ainda não exploradas (REYES-SIERRA; COELLO, 2006; MEZA et al., 2017; RAQUEL; NAVAL JR., 2005; Coello Coello; LAMONT; VELDHUIZEN, 2007; SANTANA; PONTES; BASTOS-FILHO, 2009; COELLO; PULIDO; LECHUGA, 2004). Em seguida, uma vez que as partículas se movimentaram, é necessário atualizar o *pbest* de cada partícula. No contexto de um MOPSO, o *pbest* é denominado em alguns trabalhos de líder cognitivo e tal terminologia foi usada nesta tese (SANTANA; PONTES; BASTOS-FILHO, 2009). Aqui novamente existe uma diferença com relação ao PSO, pois no caso em que

Algoritmo 2: Pseudo-código de um MOPSO genérico.

```

1 Inicialize o enxame  $S^{(0)}$  no espaço de busca;
2 Inicialize o arquivo externo  $A^{(0)}$  com soluções localmente não-dominadas de  $S^{(0)}$ ;
3 para  $t$  de 1 a  $t_{max}$  faça
4   Realize a atribuição de aptidão das soluções em  $A^{(t)}$ ;
5   Atualize líderes sociais das partículas;
6   Atualize velocidade das partículas;
7   Atualize posição das partículas;
8   Aplique algum operador de turbulência para promover a diversidade;
9   Atualize líderes cognitivos das partículas;
10  Atualize arquivo externo  $A^{(t)}$  com soluções localmente não-dominadas de
     $A^{(t)} \cup S^{(t)}$ ;
11  Realize a poda do arquivo externo  $A^{(t)}$  se necessário;
12 retorna  $A^{(t)}$ .

```

a nova posição e o novo líder cognitivo são incomparáveis, é necessário usar um outro critério para se determinar qual delas será o novo líder cognitivo (MEZA et al., 2017). Uma estratégia utilizada por exemplo é utilizar a solução mais recente como novo líder cognitivo (REYES-SIERRA; COELLO, 2006). Desta forma, toda nova solução descoberta substituirá o líder cognitivo caso esta nova solução domine o líder cognitivo atual ou no caso de ambas as soluções serem incomparáveis entre si. Vale salientar que isso é só um exemplo de decisão de projeto para a seleção do líder cognitivo. Outras possibilidades são perfeitamente plausíveis (SANTANA; PONTES; BASTOS-FILHO, 2009; Coello Coello; LAMONT; VELDHUIZEN, 2007).

A última decisão de projeto a ser tratada em um MOPSO é a questão da poda do arquivo externo (RAQUEL; NAVAL JR., 2005; MEZA et al., 2017). Devido a limitações de recursos computacionais, o arquivo externo geralmente tem um tamanho limitado (MEZA et al., 2017). Desta forma, toda vez que uma solução é inserida em um arquivo externo com tamanho máximo, o arquivo deve decidir qual solução entre as soluções do arquivo e a solução recentemente inserida deve ser eliminada (MEZA et al., 2017). Novamente, o critério de densidade no espaço de objetivos geralmente é utilizado para decidir entre as várias soluções não-dominadas (empates) presentes no arquivo externo, de modo que a solução na região de maior densidade geralmente é eliminada deterministicamente.

2.4 Otimização de Problemas com Muitos Objetivos

MOEAs e MOPSOs baseados em dominância de Pareto têm obtido sucesso em problemas multiobjetivos com dois e três objetivos. No entanto, esses algoritmos têm o seu desempenho severamente prejudicado em problemas multiobjetivos com quatro ou mais objetivos (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008; JAIMES; COELLO; CHAKRABORTY,

2008). Isso ocorre porque o número de soluções não-dominadas aumenta exponencialmente com o número de objetivos de modo que a probabilidade de uma solução ser dominada por outra solução é cada vez menor (JAIMES; COELLO; CHAKRABORTY, 2008; FARINA; AMATO, 2004; ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008).

Em 2004, Farina e Amato (FARINA; AMATO, 2004) deram uma explicação teórica para a ineficácia da dominância de Pareto em discriminar soluções em problemas multi-objetivos em que vários objetivos estão presentes. Resumidamente, eles realizaram uma análise probabilística e concluíram que a probabilidade de uma solução dominar outra diminui rapidamente com o aumento do número de objetivos, de modo que à medida que o número de objetivos aumenta, maior é a tendência de encontrar soluções que sejam incomparáveis entre si. Considere que o espaço objetivo seja contínuo e considere o mapeamento $e : \mathcal{O} \mapsto [0, 1]$ de modo que, para qualquer $\mathbf{y}^* \in \mathcal{O}$, $e(\mathbf{y}^*)$ é a razão entre a *medida* do conjunto de soluções incomparáveis ao vetor \mathbf{y}^* e a *medida* da região factível \mathcal{O} . A *medida* de um conjunto é, a grosso modo, o seu tamanho de forma que se o conjunto é geometricamente um quadrado, sua medida é a área desse quadrado; e caso o conjunto seja geometricamente um cubo, sua medida é o volume desse cubo. Desta maneira, $e(\mathbf{y}^*)$ é definido conforme a Equação 2.10

$$e(\mathbf{y}^*) = \frac{\mu\{\mathbf{y} : \mathbf{y} \parallel \mathbf{y}^*\}}{\mu(\mathcal{O})}, \quad (2.10)$$

em que μ é qualquer medida aditiva, como a medida de Lebesgue³, por exemplo. Considere agora que o espaço objetivo é um hipercubo de medida 1. Desta forma, o valor esperado de e denotado por \tilde{e} é dado pela Equação 2.11

$$\tilde{e}(m) = \int_{\mathcal{O}} 1 - (y_1 y_2 \dots y_m) - ((1 - y_1)(1 - y_2) \dots (1 - y_m)) dy, \quad (2.11)$$

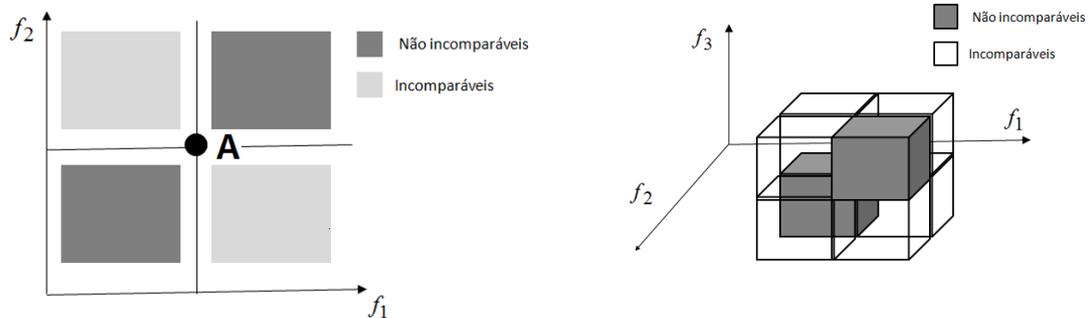
em que $\mathbf{y} = [y_1, y_2, \dots, y_m]$. Essa integral pode ser expressa, de acordo com (FARINA; AMATO, 2004), de um modo mais simples como

$$\tilde{e}(m) = \frac{2^m - 2}{2^m}. \quad (2.12)$$

A Equação 2.12 pode ser interpretada por meio da Figura 6. Para a Figura 6a, observa-se que uma solução A divide o espaço objetivo em quatro regiões retangulares, duas das quais são de soluções incomparáveis à solução em questão. Desse modo, $\tilde{e}(2) = \frac{2^2 - 2}{2^2} = \frac{2}{4}$. Para a Figura 6b, observa-se que essa mesma solução divide o espaço objetivo em oito regiões retangulares, seis das quais são de soluções incomparáveis à solução em consideração. Desse modo, $\tilde{e}(3) = \frac{2^3 - 2}{2^3} = \frac{6}{8}$. Desta forma, observa-se que, quando o número de objetivos

³ Em Teoria da Medida, o conceito matemático de *medida* formaliza a noção intuitiva de tamanho de um conjunto, tais com seu comprimento, área e volume. Assim, por exemplo, a medida de Lebesgue do conjunto $[0, 1]^2$ é igual a um que corresponde a área do retângulo corresponde no sistema de coordenadas cartesianas.

m tende ao infinito, o valor esperado $\tilde{e}(m)$ tende a 1. Ou seja, à medida que o número de objetivos aumenta, todas as soluções do espaço objetivo tendem a ser incomparáveis a uma dada solução. Tal comportamento é geral e independente do problema (FARINA; AMATO, 2004).



(a) Espaço objetivo bidimensional.

(b) Espaço objetivo tridimensional.

Figura 6 – Regiões de dominância para uma dada solução (retirado de (FIGUEIREDO, 2013)).

Da análise acima, pode-se concluir que quando o número de objetivos aumenta, todas as soluções tendem a ser não-dominadas entre si. Em consequência disso, os indivíduos da população ou as partículas do enxame tendem a ser incomparáveis entre si, de modo que MOEAs e MOPSOs baseados em dominância de Pareto não conseguem impor uma pressão seletiva em direção à Frente de Pareto (LI et al., 2015a). Desta forma, esses algoritmos são ineficazes em resolver problemas multiobjetivos com quatro ou mais objetivos. Por outro lado, esses problemas surgem naturalmente em problemas práticos do mundo real (JAIMES; COELLO; CHAKRABORTY, 2008; CAI et al., 2015). Sendo assim, devido a importância desses problemas e devido aos obstáculos que esses mesmos problemas oferecem aos algoritmos baseados em dominância de Pareto (CURRY; DAGLI, 2014), esses problemas receberam um nome especial. Desta forma, um problema multiobjetivo com quatro ou mais objetivos é denominado de Problema com Muitos Objetivos ou *Many-Objective Optimization Problem* (MaOP) em inglês.

Além da dificuldade já exposta com respeito a dominância de Pareto, MaOPs oferecem outros desafios tais como (LI et al., 2015a; LI et al., 2015b; ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008):

1. aumento exponencial do número de soluções necessárias para aproximar toda Frente de Pareto com uma dada resolução em função do número de objetivos (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008) (JAIMES; Coello Coello, 2009);
2. dificuldade de visualização das soluções. Essa dificuldade não representa um problema em si para os algoritmos, mas, por outro lado, representa uma dificuldade

para o tomador de decisão na escolha de uma ou mais soluções fornecidas pelo algoritmo (JAIMES; Coello Coello, 2009; LI et al., 2015b);

3. aumento do custo computacional. Dependendo das estruturas de dados utilizadas e dos mecanismos utilizados para atribuição de aptidão, como por exemplo a contribuição no hipervolume de uma solução para o conjunto total de soluções não-dominadas, o custo computacional dos algoritmos pode crescer exponencialmente com o número de objetivos (Garza Fabre, 2009).

Nesta tese, técnicas para lidar com questão da falta de pressão de seleção para a convergência dos algoritmos em problemas com muitos objetivos e a questão da geração da diversidade serão abordados nos próximos capítulos (LI et al., 2015b). Além disso, a questão da aproximação da Frente de Pareto por um número limitado de soluções (item 1) e do custo computacional (item 3) serão abordadas também. Uma técnica para visualização de soluções em espaços multidimensionais chamada de coordenadas paralelas (XIANG et al., 2017; LUO et al., 2017; ISHIBUCHI et al., 2017; SEADA; DEB, 2016) também será discutida nos próximos capítulos.

2.5 Abordagens Para Lidar com Problemas com Muitos Objetivos

Nesta seção, apresentam-se as abordagens propostas para lidar com MaOPs. As abordagens são baseadas na classificação realizada por Li *et al.* (LI et al., 2015a) dos algoritmos. Como a área de otimização de muitos objetivos ainda está em amadurecimento (LI et al., 2015b), outras classificações são possíveis. As abordagens são (LI et al., 2015a; LI et al., 2015b):

- Formas relaxadas de dominância. Nessa abordagem, o conceito de dominância de Pareto é modificado com o objetivo de impor uma pressão de seleção para Frente de Pareto (LI et al., 2015a; ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008) e com isso contornar a falta de pressão de convergência em cenários de muitos objetivos (HADKA; REED, 2012). Essa modificação pode ser feita de duas maneiras (LI et al., 2015a): (1) pela modificação dos valores originais dos objetivos de modo a aumentar a área de dominação, ou (2) pela contagem do número de objetivos em que uma solução é melhor, igual, ou pior que a outra (LI et al., 2015a). Alguns exemplos da abordagem (1) são (i) a relação de dominância denominada de *Control the Dominance Area of Solutions* (CDAS) proposta por Sato *et al.* (SATO; AGUIRRE; TANAKA, 2007), que permite a contração ou expansão da área de dominância; (ii) a cone-dominância proposta por Batista *et al.* (BATISTA et al., 2011); entre outras abordagens (HADKA; REED, 2012; HADKA; REED; SIMPSON, 2012; YANG et al., 2013; AGUIRRE; TANAKA, 2009). Alguns exemplos do segundo tipo de implementação de formas relaxadas de

dominância incluem (i) a relação de Favor (*Favour*) de Drechsler *et al.* (DRECHSLER; DRECHSLER; BECKER, 2001), em que uma solução é considerada melhor que a outra pelo número de objetivos em que ela é superior a outra, e com isso essa relação consegue estabelecer uma classificação entre as soluções de uma maneira mais fina; e (ii) k -otimalidade e sua contrapartida *fuzzy* proposta por Farina e Amato (FARINA; AMATO, 2004).

- Novos esquemas de estimação de diversidade. Essa abordagem busca realizar um maior controle do mecanismo de preservação da diversidade de maneira que se possa conciliar a diversidade das soluções ao mesmo tempo em que soluções não-dominadas afastadas da Frente de Pareto são penalizadas. Desta forma, pretende-se evitar a convergência do algoritmo para um único ponto pelo uso de um operador de diversidade, ao mesmo tempo em que se busca favorecer a convergência do algoritmo. Uma desvantagem desta técnica é que se o gerenciamento da diversidade não for realizado com cuidado, o algoritmo pode convergir para uma região pequena da Frente de Pareto ou até mesmo não convergir. Um exemplo desta abordagem é o mecanismo de gerenciamento de diversidade proposto por Adra e Fleming (ADRA; FLEMING, 2011; ADRA, 2007) e que foi denominado de *Diversity Management Operator* (DMO). Outra técnica é a apresentada no algoritmo *Grid-based Evolutionary Algorithm* (GrEA) proposto por Yang *et al.* (YANG *et al.*, 2013) que divide o espaço objetivo em uma malha cujo tamanho de cada malha é ajustada adaptativamente. A partir da construção dessa malha eles definiram três critérios para comparar as soluções, são elas: o ordenamento por malha, o *crowding distance* (DEB *et al.*, 2002) da malha e a distância para o ponto de origem da malha. Uma outra abordagem é o *Shift-based Density Estimation* (SDE) (LI; YANG; LIU, 2014a), que promove o descolamento de soluções afim de favorecer a convergência e a distribuição de soluções.
- Métodos baseados em agregação. Usar funções de agregação dos objetivos é um outro modo de diferenciar as soluções em cenários com muitos objetivos. Segundo (LI *et al.*, 2015a), isso pode ser realizado de duas maneiras diferentes: (1) agregação de informação individual e (2) agregação de informação da solução a partir da sua comparação com outras soluções. Os métodos de agregação de informação individual usam informação individual de cada solução e as agrega para formar um único valor. Ou seja, eles realizam a conversão do vetor objetivo em um valor escalar. A agregação pode ser feita usando os valores objetivos diretamente ou usar o *rank* de cada objetivo dentro da solução. Por outro lado, o método da agregação de comparações entre soluções realiza a agregação da comparação de uma solução com as soluções restantes da população. Alguns exemplos de métodos de agregação são: o *Multiobjective Particle Swarm Optimization with Global Detriment* (MOPSO-GD) (FIGUEIREDO, 2013), o *Multiple Single Objective Pareto Sampling*

(MSOPS) (HUGHES, 2003), o *Multiple Single Objective Pareto Sampling II* (MSOPS-II) (HUGHES, 2007), o *Ranking-dominance* (KUKKONEN; LAMPINEN, 2007), o *Global Detriment Ranking* (GDR) (Garza Fabre; PULIDO; Coello Coello, 2009; GARZA-FABRE; PULIDO; Coello Coello, 2010; LI; WANG; XU, 2017), o *Profit* (Garza Fabre; PULIDO; Coello Coello, 2009), e o *Multiobjective Evolutionary Algorithm based on Decomposition* (MOEA/D) (ZHANG; LI, 2007). Como se pode perceber, esses algoritmos, de um modo geral, transformam o problema multiobjetivo em um problema mono-objetivo, sendo esse o principal fato pelo qual os algoritmos que utilizam esses métodos têm sua diversidade comprometida e acabam convergindo, muitas vezes, para um único ponto. Sendo assim, a principal desvantagem destas técnicas é que, dependendo de como é feita esta agregação, o algoritmo que as utiliza pode ter sua diversidade comprometida. De fato, o algoritmo MOPSO-GD que foi proposto pelo autor desta tese em sua dissertação de mestrado (FIGUEIREDO, 2013), apesar de apresentar uma altíssima convergência, tinha sua diversidade comprometida. Além disso, esses métodos, de modo geral, são sensíveis a forma da Frente de Pareto. O método da *soma ponderada*, por exemplo, que consiste em transformar o problema multiobjetivo em um problema mono-objetivo por meio da combinação linear dos objetivos com fatores de ponderação (pesos), não é capaz de gerar todas as soluções da Frente de Pareto em problemas multiobjetivos cujas Frentes de Pareto são não-convexas (Coello Coello; LAMONT; VELDHUIZEN, 2007; ZITZLER, 1999; ADRA, 2007; FIGUEIREDO, 2013). Finalmente, outra desvantagem desses métodos de agregação é que eles podem envolver a agregação de objetivos em escalas diferentes (ADRA, 2007).

- Abordagens baseadas em indicadores. Como o conjunto aproximação retornado por um algoritmo multiobjetivo é avaliado de acordo com um indicador, uma estratégia interessante consiste em otimizar diretamente esse indicador. Um exemplo de indicador é o hipervolume que consiste em uma medida do tamanho (volume) do espaço dominado pelo conjunto aproximação (ZITZLER et al., 2003). O hipervolume é a única métrica monotônica à dominância de Pareto. Ou seja, sempre que um conjunto aproximação de um algoritmo domina inteiramente um outro conjunto, então o valor do hipervolume do conjunto dominante também será necessariamente maior (BADER; ZITZLER, 2011). Desta forma, um algoritmo pode tentar buscar maximizar o hipervolume (LI et al., 2015a) selecionando as soluções que realizam essa otimização. A desvantagem desta técnica pode ser o alto custo computacional demandado para calcular o indicador. O tempo de cálculo do hipervolume, por exemplo, cresce exponencialmente com o número de objetivos (LI et al., 2015a). Alguns exemplos desta abordagem são o *S Metric Selection Evolutionary Multiobjective Optimization Algorithm* (SMS-EMOA) (BEUME; NAUJOKS; EMMERICH, 2007), o *Hypervolume Estimation Algorithm* (HypE) (BADER; ZITZLER, 2011), e o *Indicator-Based Evolutionary Algorithm* (IBEA) (ZITZLER; KÜNZLI, 2004).

- Abordagem baseada em preferência. Esse método busca contornar o problema de aproximar a Frente de Pareto inteira em um MaOPs. Ao invés de tentar aproximar a Frente de Pareto inteira que exigiria um número enorme de soluções esses métodos concentram a busca em um subconjunto da Frente de Pareto de acordo com a preferência do usuário (LI et al., 2015b). A desvantagem óbvia desta técnica é que o usuário precisa especificar essas preferências, o que nem sempre é possível *a priori*. Alguns exemplos desta abordagem são o *Preference-Inspired Coevolutionary Algorithm* (PICEA) (WANG; PURSHOUSE; FLEMING, 2013c; WANG; PURSHOUSE; FLEMING, 2013b; WANG; PURSHOUSE; FLEMING, 2013a; WANG; PURSHOUSE; FLEMING, 2012), o R-NSGA-II (DEB; SUNDAR, 2006), e o RD-NSGA-II (DEB; KUMAR, 2007).
- Abordagem de redução de dimensionalidade. Esses métodos buscam primeiramente reduzir o número de objetivos preferencialmente para dois objetivos para desta forma resolver o problema original de modo aproximado usando um algoritmo baseado em dominância de Pareto por exemplo (LI et al., 2015b). Ou seja, essas técnicas buscam transformar um problema complexo em uma versão mais simplificada que possa ser resolvida pelos MOEAs e MOPSOs tradicionais (LI et al., 2015b). A desvantagem óbvia desta abordagem é sua aplicabilidade limitada uma vez que nem sempre é possível realizar uma redução de dimensionalidade do problema original. Alguns exemplos desta abordagem são: (1) o C-PCA-NSGAI e o MVU-PCA-NSGA-II propostos por Saxena e Deb (SAXENA; DEB, 2007; SINGH; ISAACS; RAY, 2011), que utilizam técnicas baseadas em correlação para realizar a redução de dimensionalidade; (2) os algoritmos propostos para resolverem os problemas δ -MOSS e k -EMOSS que são problemas que buscam determinar qual conjunto de objetivos que preserva a estrutura de dominância inerente ao problema com um certo erro (BROCKHOFF; ZITZLER, 2009; SINGH; ISAACS; RAY, 2011; JAIMES; COELLO; CHAKRABORTY, 2008); (3) o *Pareto corner search evolutionary algorithm* (PCSEA) (SINGH; ISAACS; RAY, 2011) ; e (4) a técnica de seleção de características não-supervisionadas proposta por (JAIMES; COELLO; CHAKRABORTY, 2008).
- Abordagem baseada em conjunto de pontos de referência. Esses métodos usam um conjunto de pontos de referência para medir a qualidade das soluções. Um dos principais algoritmos representantes dessa abordagem é o NSGA-III (DEB; JAIN, 2014). Nesse algoritmo, os pontos de referência são utilizados para realizar o agrupamento das soluções em torno delas de modo que se possa fazer uma estimativa da densidade das soluções. Dessa forma, NSGA-III é similar ao algoritmo NSGA-II (DEB et al., 2002), em que o *crowding distance* é substituído por esse novo operador de densidade (LI et al., 2015a; DEB; JAIN, 2014; JAIN; DEB, 2014). Essa abordagem tem a vantagem de realizar um bom balanceamento entre convergência e diversidade e é atualmente uma das abordagens mais promissoras (LI et al., 2015a; ASAFUD-

DOULA; RAY; SARKER, 2015). De fato, desde da sua publicação, muitos algoritmos usando essa abordagem foram publicados na literatura. Alguns desses algoritmos são: *clustering-ranking Evolutionary Algorithm* (crEA) (CAI et al., 2015), *Improved Decomposition-based Evolutionary Algorithm* (I-DBEA) (ASAFUDDOULA; RAY; SARKER, 2015), *Many-Objective Evolutionary Algorithm (MaOEA) based on Directional Diversity (DD) and Favorable Convergence (FC)* (MaOEA-DDFC) (CHENG; YEN; ZHANG, 2015), *Multiobjective Evolutionary Algorithm based on Decomposition and Dominance* (MOEA/DD) (LI et al., 2015b), *θ -Dominance-based Evolutionary Algorithm (θ -DEA)* (YUAN et al., 2016), *Reference Vector guided Evolutionary Algorithm* (RVEA) (CHENG et al., 2016a), e *Vector angle-based Evolutionary Algorithm* (VaEA) (XIANG et al., 2017).

2.6 Considerações Finais

Esta seção apresentou os principais conceitos e técnicas necessários para entender o problema abordado por esta tese bem como as propostas de solução. Como foi apresentado, a otimização de muitos objetivos ainda é um problema em aberto e se mostra ainda como um grande desafio para a comunidade de computação inteligente e otimização multiobjetiva (LI et al., 2015b; LI et al., 2015a). A principal questão nesses problemas é a dificuldade de conciliar convergência com diversidade (LI et al., 2015b). Das abordagens propostas na literatura para solução de problemas com muitos objetivos, a utilização de um conjunto de pontos de referência para guiar a busca para um conjunto bem distribuído de pontos próximo a Frente de Pareto tem se mostrado bastante promissora de acordo com a literatura (DEB; JAIN, 2014; LI et al., 2015a; LI et al., 2015b; ASAFUDDOULA; RAY; SARKER, 2015). De fato, desde do surgimento desta abordagem em 2012 (DEB; JAIN, 2012), surgiram muitas propostas usando essa abordagem (CAI et al., 2015; ASAFUDDOULA; RAY; SARKER, 2015; LI et al., 2015b; CHENG; YEN; ZHANG, 2015; CHENG et al., 2016a; YUAN et al., 2016; XIANG et al., 2017). No entanto, esta abordagem ainda não foi avaliada dentro de um algoritmo baseado em enxame de partículas. Sendo assim, esta tese se propõe a desenvolver um algoritmo baseado em enxame de partículas que utiliza um conjunto de pontos de referência e avaliar o desempenho desse algoritmo em problemas com muitos objetivos com diferentes dificuldades. O próximo capítulo apresenta a proposta desenvolvida nesta tese.

3 Otimização por Enxame de Partículas de Muitos Objetivos (MaOPSO)

Neste capítulo, é apresentado o algoritmo proposto nesta tese denominado de Otimização por Enxame de Partículas para Muitos Objetivos. Afim de apresentar o algoritmo de maneira didática, é apresentado primeiramente seu arcabouço geral e, em seguida, cada um dos seus componentes em detalhes.

3.1 Arcabouço do MaOPSO

O arcabouço do algoritmo proposto nesta tese chamado Algoritmo de Otimização por Enxame de Partículas para Muitos Objetivos ou *Many-Objective Particle Swarm Optimization* (MaOPSO) em inglês é resumido no Algoritmo 3. Inicialmente, N partículas são geradas aleatoriamente no espaço de busca usando uma *distribuição uniforme* para formar o enxame inicial S_0 . Em seguida, as partículas são avaliadas usando o MaOP ¹. Como outros MOPSOs, o MaOPSO tem um arquivo externo A_t de tamanho máximo denotado por $|A|_{max}$ (NEBRO; DURILLO; COELLO, 2013; REYES-SIERRA; COELLO, 2006; NEBRO et al., 2009; Coello Coello; LAMONT; VELDHUIZEN, 2007; ENGELBRECHT, 2006), inicialmente vazio, usado para armazenar soluções não-dominadas encontradas pelo algoritmo durante o processo de busca até a iteração t . Esse arquivo é então inicialmente preenchido com as soluções não-dominadas presentes no enxame inicial. Em seguida, um conjunto de pontos de referência Λ é gerado conforme a subseção 3.2.1.1 que será usado no processo de atribuição de aptidão das soluções com o objetivo de discriminar (diferenciar) as soluções do arquivo externo. Finalmente, o MaOPSO mantém um arquivo de soluções extremas Z_0 inicialmente vazio utilizado para armazenar as soluções extremas encontradas ao longo das iterações. A subseção 3.2.1.3 explica como esse arquivo de soluções extremas é atualizado.

O processo de atribuição de aptidão no MaOPSO envolve o cálculo de duas medidas, a saber, a *medida de convergência* e a *medida da densidade*. Essas medidas são calculadas com base nas distâncias ² das soluções do arquivo externo para o hiperplano que é determinado a partir das soluções extremas que são armazenadas no conjunto Z_t e que são identificadas pelo algoritmo no decorrer do processo de busca. A forma como essas soluções

¹ A instância exata do MaOP utilizado depende obviamente de cada problema. No pseudo-código optou-se por conveniência utilizar o termo genérico MaOP para se referir a qualquer problema com muitos objetivos. Essa mesma escolha foi utilizada de modo similar no contexto de MOEAs por Coello Coello *et al.* (Coello Coello; LAMONT; VELDHUIZEN, 2007) ao se utilizar o termo MOP para se referir a uma instância qualquer de problemas multiobjetivos.

² A descrição detalhada de como essas distâncias são calculadas é apresentada nas Seções de 3.2.1.4 a 3.2.1.5.

Algoritmo 3: Arcabouço do MaOPSO.

```

1 Gere um enxame inicial usando uma distribuição aleatória uniforme  $S_0$  de tamanho
  inicial  $N$ ;
2 Avalie o enxame  $S_0$  usando o MaOP;
3 Seja  $A_0 = \emptyset$  o arquivo externo inicialmente vazio;
4 Seja  $Z_0 = \emptyset$  o conjunto inicialmente vazio de pontos extremos encontrado até agora
  (arquivo de soluções extremas.);
5 Gere o conjunto de pontos de referência  $\Lambda$  usando a abordagem proposta por Das e
  Dennis (DAS; DENNIS, 1998);
6 Inicialize  $A_0$  com soluções não-dominadas presentes em  $S_0$ ;
7 para  $t = 1$  a  $t_{max}$  faça
8   Compute o ponto ideal aproximado  $\mathbf{z}^{min}$  a partir de  $A_t$ ;
   /* se o ponto ideal  $\mathbf{z}^*$  do problema não é conhecido pelo usuário */
9   Compute os novos pontos extremos  $Z_t$  a partir de  $Z_{t-1} \cup A_t$ ;
10  Compute o hiperplano a partir dos pontos extremos  $Z_t$ ;
11  Projete os pontos de referência  $\Lambda$  no hiperplano para formar um novo conjunto
   de pontos de referência  $\Lambda^*$ ;
12  Compute a medida da densidade  $\mu_a$  de cada solução  $a \in A_t$  usando o conjunto
   de referência  $\Lambda^*$  (Operador de densidade);
13  Compute a medida convergência  $\rho_a$  de cada solução  $a \in A_t$  usando o conjunto de
   referência  $\Lambda$  (Operador de convergência);
14  Selecione o líder social  $l_i \in A_t$  para cada partícula  $i \in S_t$  usando as medidas de
   densidade e convergência (cf. subseção 3.2.2);
15  Selecione o líder cognitivo para cada partícula  $i \in S_t$  (cf. subseção 3.2.3);
16  Compute a velocidade das partículas em  $S_t$  (cf. Equação 2.8);
17  Atualize a posição das partículas em  $S_t$  (cf. Equação 2.9);
18  Aplique mutação sobre 15% das partículas do enxame  $S_t$ ;
19  Avalie as partículas usando o MaOP;
20  Atualize  $A_t$  a partir de  $A_{t-1} \cup S_{t-1}$ ;
21  Pode o arquivo externo  $A_t$  se necessário;
22 retorna  $A_{t_{max}}$ .
```

extremas são identificadas é explicada na subseção 3.2.1.3. O cálculo dos operadores de densidade e convergência por sua vez são explicados nas subseções 3.2.1.4 e subseção 3.2.1.5, respectivamente. Após o cálculo dessas duas medidas, o MaOPSO aplica iterativamente uma sequência de passos que consiste basicamente em:

1. Calcular a *medida de densidade* das soluções do arquivo externo;
2. Calcular a *medida de convergência* das soluções do arquivo externo;
3. Selecionar os líderes sociais para cada partícula do enxame a partir do arquivo externo por *seleção por torneio* usando as medidas de *densidade* e *convergência*. Essas duas medidas serão responsáveis por direcionar a busca em direção a Frente de Pareto e por distribuir as soluções ao longo da Frente de Pareto de modo uniforme, respectivamente, por meio da pressão de seleção fornecida pela *seleção por torneio*;

4. Atualizar os líderes cognitivos das partículas usando dominância e distância Euclidiana;
5. Mover as partículas no espaço de decisão usando as equações usuais do PSO ³;
6. Aplicar mutação polinomial em 15% das partículas no enxame para melhorar a diversidade e evitar a convergência prematura. A razão pela qual esse valor foi usado é apresentada na subseção 3.2.4. Também na subseção 3.2.4 é apresentado por que esse valor não foi deixado como parâmetro, além de outros detalhes.
7. Atualizar o arquivo externo usando as novas soluções visitadas pelas partículas de modo que o novo arquivo atualizado contenha somente soluções não-dominadas; e
8. Podar o arquivo externo quando seu tamanho exceder o tamanho máximo $|A|_{max}$.

Esses passos são repetidos até o número máximo de iterações t_{max} ser atingido. Os detalhes de cada componente e motivação para sua existência são descritos na seção 3.2.

3.2 Componentes do MaOPSO

Esta seção apresenta em detalhes os componentes do MaOPSO.

3.2.1 Método de Atribuição de Aptidão

O método de atribuição de aptidão usado no MaOPSO é baseado em duas medidas, a saber, a *medida de convergência* para promover a convergência e a *medida da densidade* para promover a diversidade. Para calcular ambas medidas, usa-se um conjunto de pontos de referência. Estes pontos de referência são usados para se realizar o agrupamento das soluções do arquivo externo e com isso estimar a densidade destes pontos no espaço objetivo. Além disso, estes pontos são usados para favorecer soluções mais próximas da Frente de Pareto.

3.2.1.1 Geração de Pontos de Referência

Para gerar os pontos de referência, o MaOPSO usa o método proposto por Das e Dennis (DAS; DENNIS, 1998). Esse método gera um conjunto de pontos uniformemente espalhados sobre um hiperplano no espaço objetivo.

É importante destacar que a ideia de gerar pontos de referência sobre um hiperplano usando o método de Das e Dennis (DAS; DENNIS, 1998) no processo de atribuição de

³ Neste trabalho foi aplicado um esquema de velocidade máxima igual ao algoritmo SMPSO (NEBRO et al., 2009). O leitor deve ler esse trabalho (NEBRO et al., 2009) afim de implementar o MaOPSO corretamente.

aptidão de soluções não-dominadas se deve aos trabalhos de Kalyanmoy Deb e Himanshu Jain, que aplicaram a abordagem de Das e Dennis no desenvolvimento do NSGA-III, um dos algoritmos que é o estado-da-arte em otimização de muitos objetivos (DEB; JAIN, 2014; JAIN; DEB, 2014; DEB; JAIN, 2012).

Outro ponto que vale mencionar é que a distribuição uniforme de pontos de referência sobre o hiperplano é fundamental nesta tese porque é por meio dela que potencialmente pode-se conseguir uma distribuição uniforme sobre a Frente de Pareto. A subseção 3.2.1.5 e a subseção 3.2.1.4 explicam como esses pontos de referência podem ser utilizados para promover a convergência do MaOPSO e também a geração de diversidade, incluindo a cobertura total da Frente de Pareto e também a distribuição de soluções de maneira uniforme sobre ela.

A partir de tudo o que foi exposto, pode-se explicar o método de geração de pontos de referência proposto por Das e Dennis (DAS; DENNIS, 1998) como segue. Basicamente, esse método gera um conjunto de pontos uniformemente espalhados sobre um hiperplano no espaço objetivo, em que este hiperplano encontra-se no primeiro quadrante fazendo ângulos iguais com cada um dos eixos objetivos sendo que o intercepto de cada eixo é um. Em seguida, cada eixo é dividido em p divisões, em que p é um parâmetro fornecido pelo usuário. Fazendo isto, o número total de pontos de referência é dado por $H = C_{m-1}^{m+p-1}$ (CAI et al., 2015). Por exemplo, no caso de um problema com três objetivos ($m = 3$) para usar um exemplo que pode ser explicado visualmente, os pontos de referência são criados sobre um triângulo com eixos em $(1, 0, 0)$, $(0, 1, 0)$ e $(0, 0, 1)$. Considerando quatro divisões ($p = 4$) para cada eixo objetivo, então $C_{3-1}^{3+4-1} = 15$ pontos de referência são criados. O conjunto de pontos de referência usados é denotado por $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_H\}$. A Figura 7 ilustra a distribuição dos pontos de referência com $p = 4$ e três objetivos, totalizando 15 pontos de referência. O algoritmo para gerar esses pontos de referência de modo uniforme e estruturado para um espaço objetivo de qualquer dimensão m usando qualquer número de divisões p fornecido pelo usuário e descrito em detalhes em (DAS; DENNIS, 1998). Grosso modo, esse algoritmo cria cada um dos pontos de referência de modo recursivo a partir de pontos referência parciais com dimensões menores que m , até que finalmente todos pontos de referência atinjam a dimensão completa m . Para mais detalhes, o leitor é convidado a ler o artigo original (DAS; DENNIS, 1998) em que esse algoritmo é descrito.

É importante mencionar que Deb e Jain avaliaram também o uso de pontos de referência gerados de modo aleatório em seus trabalhos com o NSGA-III (DEB; JAIN, 2014; JAIN; DEB, 2014) de modo que a verificação do impacto dessa forma de geração de pontos de referência sobre o desempenho do MaOPSO é deixado como trabalho futuro. Além dessa abordagem de geração aleatória de pontos de referência, outras potenciais abordagens podem ser utilizadas no MaOPSO, de maneira que a avaliação do impacto dessas potenciais abordagens sobre o desempenho do MaOPSO é deixada também como

trabalho futuro. Pode-se, por exemplo, criar um esquema em que os pontos de referência são criados de modo adaptativo ao longo das iterações como nos trabalhos de Yan-yan Tan *et al.* (TAN *et al.*, 2013), ou como nos trabalhos de Rui Wang e colaboradores (WANG; PURSHOUSE; FLEMING, 2013c; WANG; PURSHOUSE; FLEMING, 2013a; WANG; ZHANG; GUO, 2013).

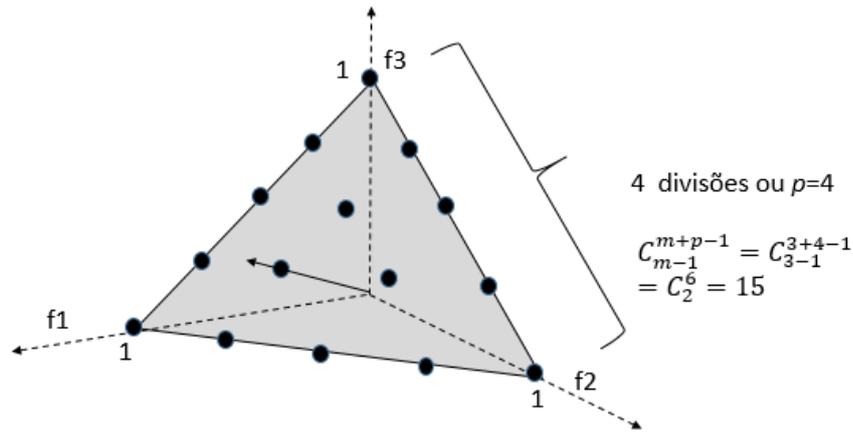


Figura 7 – Ilustração do processo de geração de pontos de referência.

3.2.1.2 Deslocamento das Soluções

Para realizar o deslocamento das soluções, é necessário, primeiro, determinar o ponto ideal $\mathbf{z}^* = (f_1^*, f_2^*, \dots, f_m^*)$ do problema. O ponto ideal de um problema multiobjetivo é o ponto cujos componentes consistem no valor mínimo f_j^* de cada função objetivo j (SINGH; ISAACS; RAY, 2011). Se o ponto ideal \mathbf{z}^* do problema é conhecido, então ele pode ser usado diretamente caso contrário ele precisa ser aproximado (estimado) a partir do arquivo externo. A aproximação (estimação) do ponto ideal $\mathbf{z}^{min} = (f_1^{min}, f_2^{min}, \dots, f_m^{min})$ é calculada como $f_j^{min} = \min_{i \in A_t} f_j(\mathbf{x}_i)$, $j = 1, \dots, m$. Para efeitos de simplificação, a partir daqui o ponto ideal verdadeiro ou sua estimação serão chamados \mathbf{z}^* . Após a determinação do ponto ideal, as soluções do arquivo externo podem ser deslocadas no espaço objetivo subtraindo cada solução i do arquivo externo pelo ponto ideal de maneira que o ponto ideal das soluções deslocadas seja o vetor nulo. Cada objetivo deslocado j de uma solução i é denotado por $f'_j(\mathbf{x}_i) = f_j(\mathbf{x}_i) - f_j^*$.

3.2.1.3 Construção do Hiperplano

Para a construção do hiperplano, é necessário inicialmente construir um *arquivo de soluções extremas* Z_t de m (número de objetivos) soluções extremas a partir das soluções presentes no arquivo externo até a iteração t . No início do algoritmo, o conjunto Z_0 é vazio, ou seja, $Z_0 = \emptyset$. A solução $i \in A_t \cup Z_{t-1}$ é uma solução extrema para o eixo (objetivo)

j se esta solução i minimiza a seguinte função denominada de *Achievement Scalarizing Function* (ASF):

$$ASF(\mathbf{x}_i, \lambda_j) = \max_{k=0}^m f'_k(\mathbf{x}_i)/\lambda_{jk}, \quad \forall i \in A_t \cup Z_{t-1}, \quad (3.1)$$

em que $\lambda_j = (\lambda_{j1}, \lambda_{j2}, \dots, \lambda_{k2}, \dots, \lambda_{jm})$ é um vetor unitário que corresponde a direção do eixo j , de modo que $\lambda_{jk} = 0$ se $k \neq j$ e $\lambda_{jk} = 1$ caso contrário; em que $k \in \{1, 2, \dots, m\}$. O arquivo de soluções extremas Z_{t-1} é o conjunto de soluções extremas da iteração anterior t . É importante destacar que as soluções extremas são determinadas a partir da união das soluções presentes no arquivo externo e das soluções extremas presentes no arquivo de soluções extremas Z_{t-1} para evitar que as soluções extremas que melhor minimizem a função ASF sejam perdidas. Uma vez determinadas as soluções extremas, o conjunto Z_t (arquivo de soluções extremas) assume a seguinte forma $Z_t = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m\}$ em que \mathbf{e}_j é o vetor objetivo que corresponde a uma solução extrema na direção do objetivo j .

A partir das soluções extremas Z_t , é possível calcular o hiperplano que passa por essas soluções extremas no espaço objetivo. O hiperplano que passa pelas soluções extremas Z_t é calculado da seguinte maneira. Como se sabe, a equação de um hiperplano é determinada por um ponto neste hiperplano \mathbf{p}_0 e pelo vetor normal a ele denotado por \mathbf{n} . Então um vetor \mathbf{v} pertence ao hiperplano se a seguinte condição for válida $(\mathbf{v} - \mathbf{p}_0) \bullet \mathbf{n} = 0$. Se uma das soluções extremas for definida como sendo o ponto \mathbf{p}_0 , então determinar o hiperplano significa encontrar o vetor normal \mathbf{n} . Encontrar o vetor \mathbf{n} por sua vez significa encontrar o valor de suas m componentes. Isso consiste em resolver um sistema linear de m incógnitas usando m equações que podem ser estabelecidas substituindo os m pontos extremos na equação $(\mathbf{v} - \mathbf{p}_0) \bullet \mathbf{n} = 0$ no lugar do vetor \mathbf{v} . Esse sistema linear pode ser escrito de forma matricial. Para isso, é necessário primeiro fazer algumas manipulações na equação $(\mathbf{v} - \mathbf{p}_0) \bullet \mathbf{n} = 0$. Essa equação pode ser reescrita da seguinte forma $\mathbf{v} \bullet \mathbf{n} = \mathbf{p}_0 \bullet \mathbf{n}$. Como existem infinitos vetores normais \mathbf{n} possíveis para um mesmo hiperplano, o produto interno $\mathbf{p}_0 \bullet \mathbf{n}$ pode assumir qualquer valor real. Assim, é necessário fixar um valor real de modo a fixar um vetor normal particular. Então, pode-se ter a seguinte equação sem perda de generalidade $\mathbf{v} \bullet \mathbf{n} = 1$. A partir dessa equação linear pode-se construir um sistema linear com m equações substituindo o vetor \mathbf{v} pelos vetores $\mathbf{e}_1 - \mathbf{z}^*$, $\mathbf{e}_2 - \mathbf{z}^*$, \dots , $\mathbf{e}_m - \mathbf{z}^*$, que correspondem as soluções extremas deslocadas pelo vetor ideal. Se esses vetores constituírem uma matriz $\mathbf{E} = [\mathbf{e}_1 - \mathbf{z}^*, \mathbf{e}_2 - \mathbf{z}^*, \dots, \mathbf{e}_m - \mathbf{z}^*]^T$, então o sistema linear pode ser definido pela Equação 3.2:

$$\mathbf{E}\mathbf{n} = \mathbf{u}, \quad (3.2)$$

em que $\mathbf{u} = (1, 1, \dots, 1)^T$ é o vetor unitário.

Resolvendo essa equação, o vetor normal \mathbf{n} é determinado por:

$$\mathbf{n} = \mathbf{E}^{-1}\mathbf{u}. \tag{3.3}$$

A Figura 8 ilustra a criação do hiperplano a partir das soluções extremas para o caso de três objetivos apenas para efeitos de ilustração visual. Nessa figura, o arquivo de soluções extremas Z_t é formado por três soluções extremas $Z_t = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, e, a partir dessas soluções extremas, o hiperplano que passa pelos interceptos $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ é determinado.

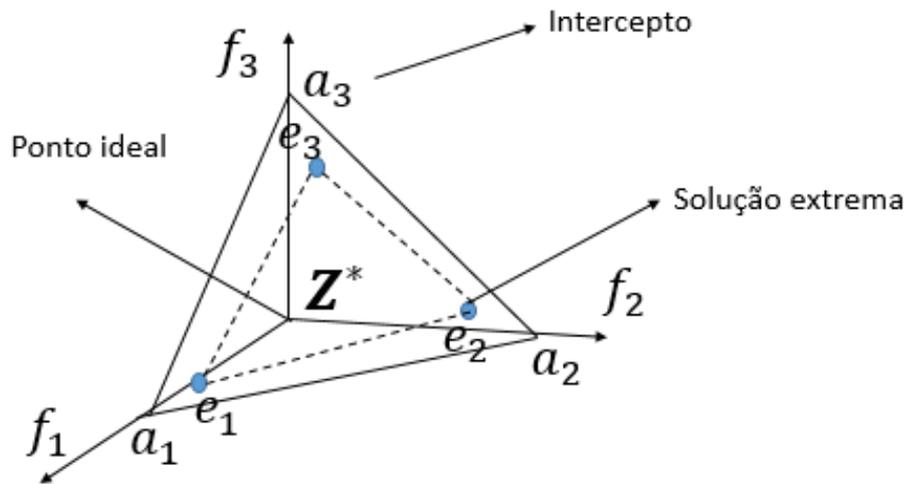


Figura 8 – Construção do hiperplano a partir das soluções extremas do arquivo externo.

Uma vez determinado o hiperplano por meio do cálculo do vetor normal \mathbf{n} , os pontos de referência da subseção 3.2.1.1 denotados por Λ são recalculados e projetados nesse hiperplano e eles são denotados por $\Lambda^* = \{\lambda'_1, \lambda'_2, \dots, \lambda'_m\}$ em que λ'_j é o ponto de referência λ_j projetado no hiperplano. Esses novos pontos de referência são calculados da seguinte forma. Seja λ' o novo ponto de referência a ser calculado e λ o ponto de referência original, então achar λ' consiste em achar o escalar c da equação $\lambda' = c\lambda$ uma vez que λ' e λ são paralelos entre si. Esse escalar é determinado resolvendo-se a seguinte equação $\lambda' \bullet \mathbf{n} = (c\lambda) \bullet \mathbf{n} = 1$. Desta forma, c é calculado usando a equação $c = 1/(\lambda \bullet \mathbf{n})$ e portanto $\lambda' = \lambda/(\lambda \bullet \mathbf{n})$. A Figura 9 ilustra o processo de projeção dos pontos de referência originais no hiperplano para o caso bidimensional. Na Figura 9, os triângulos são os pontos de referência originais denotados pelo conjunto Λ e que nessa figura são os pontos $\Lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\}$. Realizando-se a operação de projeção dos pontos de referência originais Λ sobre o hiperplano de acordo com o procedimento já explicado, obtêm-se os novos pontos de referência $\Lambda^* = \{\lambda'_1, \lambda'_2, \dots, \lambda'_m\}$ que estão sobre o hiperplano determinado a partir das soluções extremas Z_t , e que na Figura 9 são representadas por círculos.

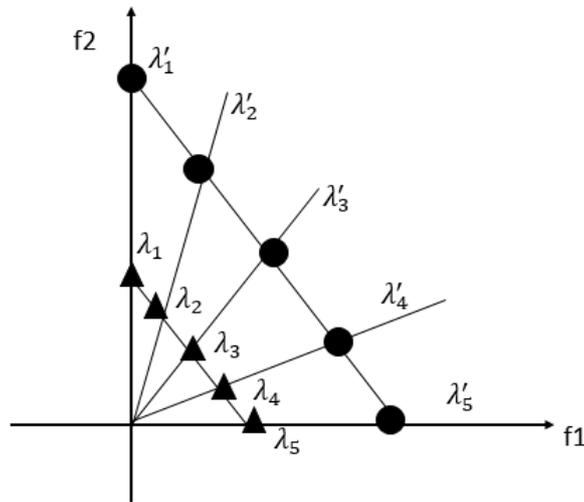


Figura 9 – Projeção dos pontos de referência originais no hiperplano.

3.2.1.4 Operador de Densidade

Um aspecto importante de qualquer algoritmo proposto a resolver um MaOP é promover a diversidade das soluções obtidas. Desta forma, foi desenvolvido um operador de densidade para estimar a densidade baseada em um algoritmo de agrupamento. Este operador funciona como segue. Primeiramente, as soluções do arquivo externo A_t são projetadas sobre o hiperplano criado na subseção 3.2.1.3. Essa projeção é similar a projeção dos pontos de referência.

Suponha um ponto \mathbf{v} a ser projetado, então a projeção deste ponto corresponde a um vetor \mathbf{u} que está sobre o hiperplano. Como \mathbf{u} pertence ao hiperplano então ele pode ser escrito como $\mathbf{u} \bullet \mathbf{n} = 1$. Como \mathbf{u} é paralelo ao vetor \mathbf{v} ele também pode ser escrito como $\mathbf{u} = c\mathbf{v}$. Fazendo a substituição de $\mathbf{u} = c\mathbf{v}$ na equação $\mathbf{u} \bullet \mathbf{n} = 1$, o vetor \mathbf{u} (projeção) pode ser determinado por $\mathbf{u} = \mathbf{v}/(\mathbf{v} \bullet \mathbf{n})$.

Após a projeção das soluções, cada solução projetada é associada com o ponto de referência que lhe é mais próximo sobre o hiperplano. Desta maneira, os pontos de referência no hiperplano formam agrupamentos de soluções que são a eles associados. Seja $\mathbf{f}'(\mathbf{x}_i)$ a projeção sobre o hiperplano do vetor objetivo $\mathbf{f}(\mathbf{x}_i)$ de uma solução $i \in A_t$. Então, define-se o agrupamento Φ_j de um ponto de referência λ'_j sobre o hiperplano como:

$$\Phi_j = \{i \in A_t \mid \|\mathbf{f}'(\mathbf{x}_i) - \lambda'_j\| \leq \|\mathbf{f}'(\mathbf{x}_i) - \lambda'_k\|, \forall \lambda'_k \in \Lambda^*\}. \quad (3.4)$$

Uma vez que é definido o agrupamento de cada ponto de referência, pode-se estimar a densidade ao redor deste ponto de referência contando o número de soluções no arquivo externo associados a ele. Desta maneira, pode-se definir a medida de densidade μ_i de uma solução $i \in A_t$ como o tamanho $\mu_i = |\Phi_k|$ do agrupamento em que ele está associado ($i \in \Phi_k$). Por exemplo, suponha que o agrupamento Φ_j de um ponto de referência λ'_j

seja formado pelas soluções $\Phi_j = \{a, b, c\}$, então a medida de densidade destas soluções é igual a 3. A Figura 10 ilustra o processo de agrupamento das soluções em torno dos pontos de referência para estimação da densidade. De acordo com essa figura, as soluções $\{a, b, c, d\}$ são soluções que estão em um mesmo agrupamento e, portanto, tem uma medida de densidade igual $\Phi_a = \Phi_b = \Phi_c = \Phi_d = 4$. Já as soluções $\{e, f\}$ que estão em outro agrupamento tem medida de densidade igual $\Phi_e = \Phi_f = 2$.

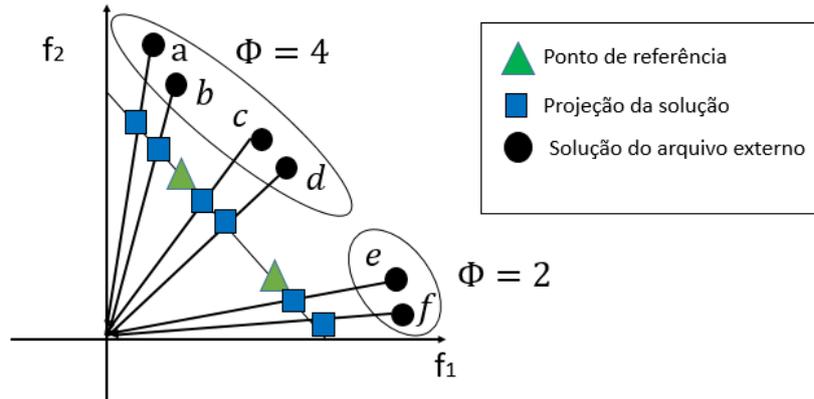


Figura 10 – Funcionamento do operador de densidade.

3.2.1.5 Operador de Convergência

A medida de convergência é usada para promover a convergência para a Frente de Pareto e é definida como segue. Para cada ponto de referência λ_j em Λ , calcula-se o *ASF* (usando a Equação 3.1) das soluções do arquivo externo associadas a este ponto de referência com respeito a ele. O valor do *ASF* para uma solução i no arquivo externo é definida como a *medida de convergência* e é denotado por ρ_i . Matematicamente, a medida de convergência ρ_i de uma solução $i \in A_t$ é definida como $\rho_i = ASF(\mathbf{x}_i, \lambda_j)$ em que $i \in \Phi_j$, e λ_j é um ponto de referência do conjunto Λ ao qual a solução i , com vetor de decisão \mathbf{x}_i , está associada pelo processo de agrupamento realizado na subseção 3.2.1.4. O ponto de referência λ_j , no caso do *ASF*, pode ser pensado também como o vetor que determina a direção ao longo da qual ocorre a sua minimização. Esse vetor pode ser determinado simplesmente conectando-se o ponto de referência λ_j com a origem do espaço objetivo, isto é, cada vetor de referência é determinado pela diferença vetorial entre um ponto de referência λ_j em particular e a origem do espaço objetivo. Neste sentido, cada ponto de referência λ_j produz um vetor λ_j de modo que cada ponto de referência se torna um vetor, de modo que λ_j pode ser tratado tanto como um ponto e como vetor a depender do contexto. Desta forma, no caso do cálculo da função *ASF*, qualquer ponto de referência λ_j do conjunto Λ pode ser pensando como um vetor que determina a direção do processo de minimização da função *ASF*.

A Figura 11 apresenta uma ilustração do cálculo da medida de convergência e consequentemente da função ASF . Nessa figura, as soluções apresentadas estão associadas a um vetor de referência $\lambda = (0, 5; 1, 0)$ em particular. Como se pode perceber, o vetor λ determina uma direção através da qual ocorrerá o processo de minimização da função ASF . As linhas tracejadas indicam os pontos que apresentam o mesmo valor segundo a função ASF , ou seja, são curvas isométricas da função ASF com respeito ao vetor λ em particular. Sendo assim, os pontos $A = (0, 3; 0, 8)$ e $B = (0, 5, 1, 0)$ quando calculados com respeito ao vetor λ , isto é, na direção do vetor λ apresentam o mesmo valor na função ASF . Ou seja, $ASF(A, \lambda) = \max \left\{ \frac{0,3}{0,5}, \frac{0,8}{1,0} \right\} = 0,8$ e $ASF(B, \lambda) = \max \left\{ \frac{0,4}{0,5}, \frac{0,6}{1,0} \right\} = 0,8$.

Vale a pena mencionar que as soluções A e B apresentam um ASF menor que a solução $C = (0, 5; 0, 6)$ cujo valor é $ASF(C, \lambda) = \max \left\{ \frac{0,5}{0,5}, \frac{0,6}{1,0} \right\} = 1,0$. Isso significa que as soluções A e B apresentam uma convergência melhor que a solução C , como se pode observar pela Figura 11. Outro ponto a se destacar é que o ponto de menor ASF corresponde justamente ao ponto da Frente de Pareto que é interceptado por um vetor paralelo ao vetor de referência λ em questão. Com isso, pode-se verificar que a minimização da função ASF corresponde a identificação de um ponto Pareto-ótimo. Como cada vetor de referência corresponde a uma função ASF em particular, a minimização de todas essas funções permite que se encontre várias soluções Pareto-ótimas. Desta forma, caso o conjunto de pontos de referência seja formado por pontos uniformemente distribuídos em torno de um hiperplano, como corre com os pontos de referência Λ da subseção 3.2.1.1, é esperado que a minimização das funções ASF para esses pontos possam levar ao encontro de soluções Pareto-ótimas diferentes para cada vetor de referência, permitindo, com isso, a identificação de soluções Pareto-ótimas aproximadamente uniformes ao longo da Frente de Pareto do problema.

Finalmente, é importante enfatizar essa propriedade do ASF , de encontrar soluções Pareto-ótimas ao longo das direções especificadas pelos vetores de referência a partir de sua minimização, foi uma das razões pela quais o ASF foi utilizado como uma medida de convergência nesta tese. Além disso, caso um conjunto de vetores espalhados de maneira uniforme ao longo do espaço objetivo possa ser definido, a função ASF pode ser usada para identificar soluções Pareto-ótima bem distribuídas de maneira aproximadamente uniforme no espaço objetivo. Sendo esse último fato, uma outra razão pela qual se usou a função ASF nesta tese.

3.2.2 Seleção dos Líderes Sociais

O papel dos líderes sociais é funcionar como uma espécie de memória global do enxame em que soluções não-dominadas encontradas pelo algoritmo até uma dada iteração são utilizadas como guias (referência) para a exploração de boas regiões do espaço de busca de modo a permitir a identificação de novas soluções não-dominadas.

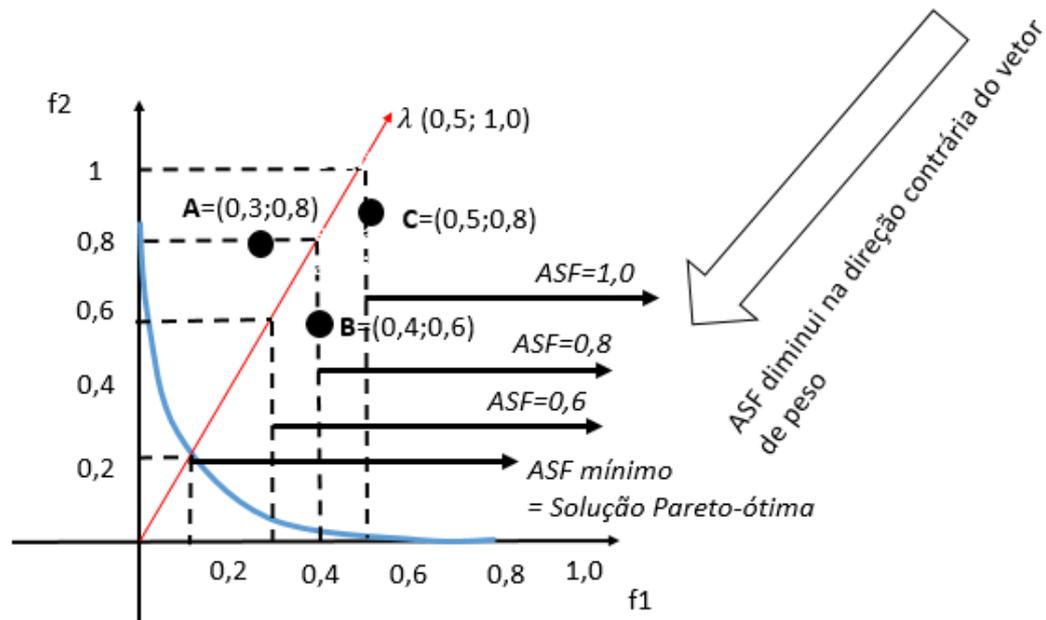


Figura 11 – Ilustração do método ASF.

No MaOPSO, os líderes sociais das partículas são selecionados do arquivo externo e ocorre de duas maneiras diferentes. Assim, o enxame S_t é dividido em dois sub-enxames \bar{S}_1^t e \bar{S}_2^t . Para o primeiro sub-enxame \bar{S}_1^t , os líderes que são atribuídos às partículas são soluções extremas presentes no arquivo externo. Dessa forma, estabelece-se um elitismo sobre as soluções extremas. O objetivo deste processo de seleção é pesquisar por novas soluções extremas mais próximas da Frente de Pareto permitindo a construção de um hiperplano cada vez mais próximo da Frente de Pareto promovendo, desta forma, a maior cobertura da Frente de Pareto e também a convergência do algoritmo uma vez que as partículas se aproximam cada vez mais dos pontos extremos da Frente de Pareto e, com isso, da Frente de Pareto como um todo.

É importante enfatizar e explicar como ocorre o processo de formação desses dois sub-enxames \bar{S}_1^t e \bar{S}_2^t . Os dois sub-enxames, \bar{S}_1^t e \bar{S}_2^t , são inicialmente formados de modo aleatório. Entretanto, uma vez que os sub-enxames são determinados, as partículas não podem mais mudar de sub-enxames. Ou seja, os sub-enxames são mantidos fixos ao longo de todo o processo de busca do MaOPSO. Contudo, o fato dos sub-enxames serem isolados um do outro, de maneira que não é possível uma partícula migrar de um sub-enxame para outro, e também não existir nenhuma intersecção entre os sub-enxames, de modo que uma partícula não pode pertencer a dois sub-enxames, não significa que esses dois sub-enxames do MaOPSO não se comuniquem entre si. Com efeito, as partículas podem compartilhar informações sobre boas regiões do espaço de busca por meio do arquivo

externo. Sendo assim, todas as soluções visitadas pelas partículas de cada sub-enxame são armazenadas no mesmo arquivo externo, de modo que boas regiões descobertas por um determinado sub-enxame pode ser fonte de potenciais líderes para outro sub-enxame e vice-versa numa espécie de processo co-evolucionário. Sendo que cada sub-enxame busca por um determinado aspecto da aproximação da Frente de Pareto. O sub-enxame \bar{S}_1^t busca aumentar a cobertura da Frente de Pareto e, como ele refina as soluções extremas, esse sub-enxame também busca simultaneamente a convergência para os extremos da Frente de Pareto, auxiliando, de certa forma, a convergência para Frente de Pareto como um todo por meio de sua iteração com o sub-enxame \bar{S}_2^t por intermédio do arquivo externo. O sub-enxame \bar{S}_2^t , por sua vez, busca a convergência para toda a Frente de Pareto e também procura distribuir as soluções uniformemente ao longo desta Frente de Pareto. Na literatura, o uso de sub-enxames é um conceito bastante conhecido no contexto da otimização por enxame de partículas (ENGELBRECHT, 2006). E, além disso, já existem algoritmos de enxames de partículas para otimização de muitos objetivos que utilizam essa abordagem, tais como o I-Multi (BRITTO; MOSTAGHIM; POZO, 2013) que é um algoritmo de enxame de partículas puro e sua evolução hibridiza com o algoritmos de estimação de distribuição, o C-Multi (JR.; SANTANA; POZO, 2016; CASTRO; POZO, 2014).

Para o segundo sub-enxame, determina-se os líderes de \bar{S}_2^t usando as medidas de diversidade e convergência para discriminar as soluções do arquivo externo de modo que as melhores soluções de acordo com estas medidas sejam selecionadas. Esse procedimento explicado em detalhes na subseção 3.2.2.2. O tamanho L da população \bar{S}_1^t é definido como aproximadamente 1/3 da população original S_t ($L = \lfloor |S|/3 \rfloor$ em que $\lfloor \cdot \rfloor$ é o operador piso) e este valor foi determinado a partir de estudos empíricos preliminares realizados durante a fase de projeto do algoritmo. De um conjunto de valores verificados pelo autor, o valor de um 1/3 permitia a convergência do algoritmo para o conjunto de problemas avaliados nesta tese e, por isso, ele foi definido como uma constante no algoritmo e não como um parâmetro. Contudo, um estudo mais amplo e profundo sobre o efeito desse parâmetro sobre o desempenho do MaOPSO é deixado como trabalho futuro, uma vez que, por restrições de tempo, isso não pôde ser realizando nesta tese. O primeiro e o segundo estágio da seleção dos líderes sociais são descritos nas próximas subseções.

3.2.2.1 Seleção dos Líderes Sociais para \bar{S}_1

Para o sub-enxame \bar{S}_1^t , os líderes sociais que são atribuídos às partículas deste sub-enxame são soluções extremas presentes no arquivo externo. As soluções extremas são soluções no arquivo externo que minimizam o ASF com respeito aos vetores unitários que são paralelos aos eixos objetivos. Seja $e_1 \in A_t$ uma solução no arquivo externo que corresponde a uma solução extrema no objetivo f_1 , $e_2 \in A_t$ uma solução extrema no objetivo f_2 e assim por diante. Então, o conjunto $\Psi = \{e_1, e_2, \dots, e_m\}$ ($\Psi \subset A_t$) consiste em m soluções extremas, uma para cada objetivo. Assim, o número de soluções

extremas é igual ao número de objetivos. Após a determinação do conjunto Ψ , pode-se realizar o processo de atribuição dos líderes sociais para o sub-enxame \bar{S}_1 . Isso é feito do seguinte modo. Atribui-se as soluções extremas em Ψ às partículas no enxame \bar{S}_1^t de modo que aproximadamente $\lfloor |\bar{S}_1|/m \rfloor$ das partículas em \bar{S}_1^t tenham uma solução extrema particular em Ψ como líder. O efeito disto é dividir o enxame \bar{S}_1^t em m sub-enxames com aproximadamente o mesmo tamanho e cada sub-enxame é especializado em pesquisar ao redor de uma solução extrema em Ψ . Assim, cada um dos m sub-enxames é especializado em pesquisar por novas soluções extremas que sejam cada vez mais próximas da Frente de Pareto. A motivação para escolher as soluções extremas como líderes sociais é promover uma maior cobertura da Frente de Pareto, permitindo consequentemente uma maior geração de diversidade de soluções. Além disso, esse mecanismo também promove a convergência pois ele permite que o enxame possivelmente se aproxime cada vez mais dos pontos extremos da Frente de Pareto, fazendo, com isso, que as partículas se aproximem da Frente de Pareto como um todo. Sendo assim, esse mecanismo realiza a especialização de cada partícula do sub-enxame \bar{S}_1 para um e somente um objetivo, isto é, cada partícula em \bar{S}_1 fica associada a um objetivo em particular. Com isso, a partícula realiza uma busca por uma melhor aproximação da solução extrema do objetivo a qual foi associado. A avaliação de outras abordagens para determinação dos líderes das partículas tais como o uso do *Corner Sort* (WANG; YAO, 2014) foi deixado como trabalho futuro. Além disso, a avaliação de um mecanismo de atribuição das soluções extremas para as partículas de modo aleatório, ao invés da associação fixa de uma partícula para a solução extrema correspondente a um objetivo em particular como feito nesta tese, sobre o desempenho do MaOPSO é deixado como trabalho futuro.

A Figura 12 ilustra o processo de seleção dos líderes sociais para o sub-enxame \bar{S}_1^t . Como se pode perceber por essa figura, as partículas são conduzidas pelas soluções extremas do arquivo externo de modo a encontrar soluções extremas cada vez mais próximas da Frente de Pareto. O processo de seleção do líder social para o sub-enxame \bar{S}_1^t é dado no Algoritmo 4. Segundo esse algoritmo, o índice do líder social da partícula i é definido por l_i e ele é escolhido do conjunto Ψ segundo a ordem do laço **para cada** de maneira circular, isto é, quando a última solução extrema (e_m) é definida como líder de uma partícula i , o líder da partícula $i + 1$ é a solução extrema e_1 , o líder da partícula $i + 2$ é a solução extrema e_2 e assim por diante.

3.2.2.2 Seleção dos líderes sociais para \bar{S}_2

Para selecionar os líderes sociais do sub-enxame \bar{S}_2 , aplica-se a *seleção por torneio* sobre as soluções no arquivo externo considerando o número de membros no torneio igual ao número de objetivos (m). A seleção por torneio foi usada porque ela é um método simples e de baixo custo computacional (MILLER et al., 1995; MILLER; GOLDBERG, 1996; BLICKLE; THIELE, 1996). Além de ser bastante utilizada na literature de otimização multiobjetiva

Algoritmo 4: Processo de seleção do líder social para o sub-enxame \bar{S}_1

```

1  $j = 1$ ;
2 para cada  $i \in \bar{S}_1$  faça
3    $l_i = e_j, e_j \in \Psi$ ;
4    $j = j + 1$ ;
5   se  $j > m$  então
6      $j = 1$ ;
   /*  $m$  é o número de objetivos

```

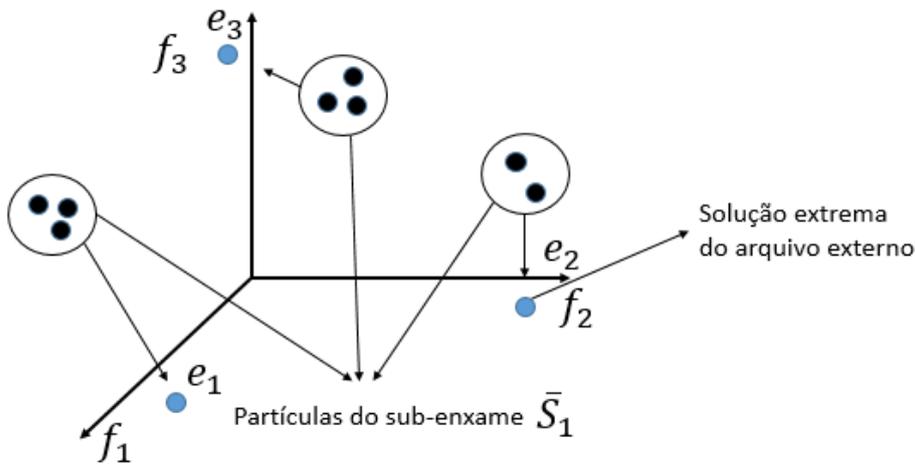


Figura 12 – Processo de seleção do líder social no MaOPSO para o sub-enxame \bar{S}_1 .

e de muitos objetivos tanto para as abordagens evolucionárias (DEB et al., 2002; JAIN; DEB, 2014), como para as abordagens de enxame de partículas (NEBRO et al., 2009; DE CARVALHO; POZO, 2012). Desta forma, esse método de seleção foi preferido aos outros métodos de seleção (BLICKLE; THIELE, 1996).

O procedimento de seleção dos líderes é descrito no Algoritmo 5. Para cada partícula $i \in \bar{S}_2$ (laço **para cada**), executa-se o procedimento por seleção por torneio (laço **para**). Na seleção por torneio, uma solução $j \in A_t$ é considerada melhor que uma solução $k \in A_t$ se a primeira delas tem uma densidade menor que a segunda, isto é, $\mu_j < \mu_k$. Entretanto, se $\mu_j = \mu_k$, então usa-se a medida de convergência para decidir qual solução é a melhor. Assim, no caso de empate, seleciona-se a solução j como líder se $\rho_j < \rho_k$. Caso $\rho_j > \rho_k$, a solução k é selecionada. Finalmente, no caso de haver empate com respeito a medida de convergência, algo raro de acontecer pois os valores de ASF são contínuos, escolhe-se aleatoriamente umas das soluções como líder social.

Algoritmo 5: Processo de seleção do líder social para o sub-enxame \bar{S}_2

```

1 para cada  $i \in \bar{S}_2$  faça
2    $l_i = \text{random}[1, |A_t|]$ ;
3   para  $k = 2$  até  $m$  faça
4      $j = \text{random}[1, |A_t|]$ ;
5     se  $\mu_j < \mu_{l_i}$  então
6        $l_i = j$ ;
7     se  $\mu_j = \mu_{l_i}$  então
8       se  $\rho_j < \rho_{l_i}$  então
9          $l_i = j$ ;

```

3.2.3 Seleção dos líderes cognitivos

O papel do líder cognitivo no MaOPSO é permitir que cada partícula tenha uma solução particular e independente das soluções do arquivo externo e que funcione como guia para a exploração de novas regiões diferentes daquelas exploradas pelas soluções do arquivo externo.

No começo da execução do MaOPSO, o líder cognitivo de cada partícula é sua posição inicial no espaço de busca. Em seguida, o líder cognitivo é atualizado como segue. A solução atual é comparada a solução cognitiva, e a solução atual (partícula) somente substitui seu líder cognitivo se ela domina esse líder cognitivo. Se as soluções são incomparáveis entre si, então as distâncias Euclidianas da solução atual e do líder cognitivo para o ponto ideal são calculadas e a solução com a menor distância é escolhida como o novo líder cognitivo. Assim, o processo de seleção do líder cognitivo é usado para promover a convergência. A Figura 13 ilustra o processo de seleção do líder cognitivo quando a partícula e seu líder cognitivo são incomparáveis. Nesse caso como a partícula está mais próxima do ponto ideal, ela é definida como o novo líder cognitivo.

3.2.4 Perturbação do Enxame

Um dos grandes problemas com técnicas baseadas em enxame de partículas é a perda de rápida diversidade no enxame o leva frequentemente a convergência prematura (COELLO; PULIDO; LECHUGA, 2004; REYES-SIERRA; COELLO, 2006). Para prevenir isso, um mecanismo recorrente é utilizar alguma forma de perturbação no enxame de maneira que algumas partículas sejam levadas para regiões ainda não exploradas do espaço de busca (COELLO; PULIDO; LECHUGA, 2004). Este mecanismo chamado por alguns autores de turbulência (Coello Coello; LAMONT; VELDHUIZEN, 2007; COELLO; PULIDO; LECHUGA, 2004) é similar ao mecanismo de mutação presente nos Algoritmos Evolucionários (ENGELBRECHT, 2007).

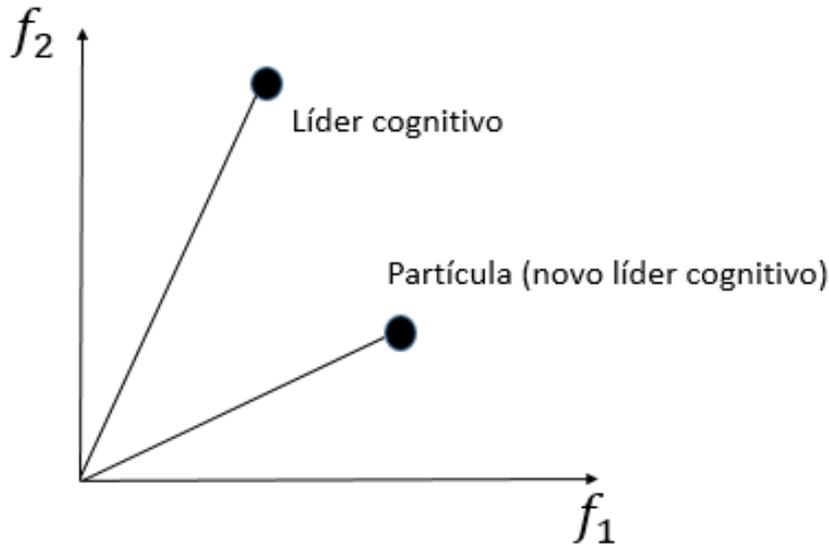


Figura 13 – Processo de seleção do líder cognitivo no MaOPSO.

Com o objetivo de promover a diversidade e remediar a rápida perda de diversidade no MaOPSO, utilizou-se o mesmo mecanismo de turbulência utilizado em um algoritmo denominado *Speed-constrained Multi-objective PSO* (SMPSO) (NEBRO et al., 2009). Esse algoritmo é um exemplo de um MOPSO que é estado-da-arte e que serviu de base para o desenvolvimento do MOPSO-GD (FIGUEIREDO, 2013). O MOPSO-GD foi algoritmo desenvolvido pelo autor desta tese durante o seu mestrado e que é fortemente baseado no SMPSO (FIGUEIREDO, 2013). Em virtude disso, utilizou-se o mesmo mecanismo de turbulência do SMPSO e conseqüentemente do MOPSO-GD no MaOPSO. Sendo assim, o operador de turbulência foi aplicado, como ocorre no SMPSO, em 15% das partículas do enxame. Essa forma de mutação é denominada de *mutação polinomial* (DEB; GOYAL, 1996).

A mutação polinomial foi proposta por Deb e Goyal (DEB; GOYAL, 1996) e consiste em um operador de mutação desenvolvido originalmente para Algoritmos Genéticos com representação real. Esse operador de mutação realiza variações em cada variável de decisão de acordo com uma função de distribuição de probabilidade polinomial. Desta forma, esse operador é aplicado com uma probabilidade p_m sobre as variáveis de decisão das soluções. A probabilidade p_m é chamada probabilidade de mutação. Essa probabilidade é geralmente um valor pequeno para assegurar que boas soluções não sejam perdidas. Assim, esse operador é aplicado probabilisticamente sobre cada variável de decisão de modo que a probabilidade de que uma solução seja afetada por esse operador é dado pela Equação 3.5

$$p(\text{mutação ser aplicada em } \mathbf{x}_i) = 1 - (1 - p_m)^n, \quad (3.5)$$

em que n é a dimensão do espaço de busca.

Para explicar a mutação polinomial, considere o caso unidimensional. Seja x o gene da solução original, y o gene da solução após a aplicação do operador de mutação (gene modificado pela mutação), e Δ_{max} a perturbação máxima permitida sobre o gene x para produzir y . Então, a mutação polinomial funciona como segue:

1. Amostrar um valor de uma função de distribuição uniforme no intervalo $[0, 1]$, isto é, $u \sim U(0, 1)$;
2. Calcule o fator de perturbação $\bar{\delta}$ usando a Equação 3.6,

$$\bar{\delta} = \begin{cases} (2u)^{\frac{1}{\eta_m+1}} - 1 & \text{se } u \leq \frac{1}{2}, \\ (1 - [2(1-u)]^{\frac{1}{\eta_m+1}}) & \text{caso contrário.} \end{cases} \quad (3.6)$$

em que η_m é o índice da distribuição do operador de mutação.

3. Calcule a variável modificada y conforme a Equação 3.7,

$$y = x + \bar{\delta}\Delta_{max}. \quad (3.7)$$

Para o caso multidimensional, esse procedimento pode ser repetido independentemente para cada variável de decisão. Cabe ressaltar que o procedimento apresentado apenas é usado para variáveis em que os limites superiores e inferiores não são especificados, isto é $x \in [-\infty, +\infty]$. Quando uma variável é restrita a um limite superior x_{max} e inferior x_{min} , isto é $x \in [x_{min}, x_{max}]$, a Equação 3.8 é usada no lugar da Equação 3.6,

$$\bar{\delta} = \begin{cases} [2u + (1-2u)(1-\delta)^{\eta_m+1}]^{\frac{1}{(\eta_m+1)}} - 1 & \text{se } u \leq \frac{1}{2}, \\ 1 - [2(1-u) + 2(u-\frac{1}{2})(1-\delta)^{\eta_m+1}]^{\frac{1}{(\eta_m+1)}} & \text{caso contrário,} \end{cases} \quad (3.8)$$

em que δ é dado pela Equação 3.9

$$\delta = \frac{\min[(x - x_{min}), (x_{max} - x)]}{(x_{max} - x_{min})}. \quad (3.9)$$

Essa equação assegura que nenhuma solução será criada fora do intervalo do espaço de busca $[x_{min}, x_{max}]$. É importante mencionar que ao se usar essa equação, deve-se fazer $\Delta_{max} = x_{max} - x_{min}$ quando a Equação 3.7 for aplicada.

Por fim, é importante ressaltar que a avaliação de outros operadores de mutação não foi realizada em virtude de restrições de tempo a esse doutorado. Sendo assim, a avaliação de outros operadores de turbulência é deixada como trabalho futuro. Um possível operador a ser avaliado é o operador de turbulência proposto por Coello Coello *et*

al. (COELLO; PULIDO; LECHUGA, 2004) em seu MOPSO. Além disso, uma análise sobre o efeito dos parâmetros da mutação polinomial sobre o desempenho do MaOPSO é também deixado como trabalho futuro. Finalmente, nesta tese utilizou-se por restrições de tempo a percentagem de 15% de partículas afetadas pelo operador de turbulência como utilizado no SMPSO e no MOPSO-GD. Desta forma, uma investigação sobre a porcentagem adequada de partículas a serem afetadas pelo operador de mutação polinomial (e outros operadores de turbulência também) para cada problema avaliado é uma outra direção de trabalho futuro.

3.2.5 Atualização do Arquivo Externo

Como a seleção dos líderes é executada no arquivo externo (BRITTO; POZO, 2012b; BRITTO; POZO, 2012a; DE CARVALHO; POZO, 2012; REYES-SIERRA; COELLO, 2006), um bom gerenciamento deste arquivo é muito importante e tem um grande impacto sobre o desempenho do algoritmo (REYES-SIERRA; COELLO, 2006; Coello Coello; LAMONT; VELDHUIZEN, 2007; COELLO; PULIDO; LECHUGA, 2004). Em cada iteração, o arquivo externo é atualizado como segue. Primeiramente, tenta-se inserir cada solução localmente não-dominada do enxame S_t no arquivo externo. Se esta solução inserida é dominada por qualquer solução do arquivo, então esta solução é descartada. Caso contrário, esta solução é inserida no arquivo externo e as soluções do arquivo que são dominadas por esta nova solução são removidas. Sendo assim, no final deste processo, o arquivo externo é um conjunto não-dominado.

Após essa etapa, se o tamanho do arquivo externo excede o tamanho máximo permitido, ele é truncado de modo a conter um tamanho máximo $|A|_{max}$ estipulado pelo usuário. Para isto, usa-se a medida de densidade e a medida de convergência para comparar as soluções no arquivo da mesma maneira que elas foram comparadas na seleção do líder social. Em outras palavras, ordena-se as soluções baseadas primeiramente na medida de densidade e quando duas soluções são iguais em termos de densidade, elas são então comparadas usando a medida de convergência. Depois disso, a pior solução é encontrada e descartada. Em seguida, a medida de densidade de todas as soluções que pertencem ao grupo da solução eliminada é diminuída por um. A eliminação de soluções dentro do arquivo externo continua até o arquivo atingir o tamanho máximo permitido.

Finalmente, é importante ressaltar que o tamanho do arquivo externo tem influência sobre o desempenho do MaOPSO e dos MOPSOs em geral. É conhecido que arquivos externos podem impactar na convergência dos algoritmos (Coello Coello; LAMONT; VELDHUIZEN, 2007), de modo que arquivos externos muito grandes podem diminuir a pressão de convergência uma vez que muitas soluções não-dominadas podem ser candidatas a líderes sociais provocando assim a diminuição da pressão de seleção para Frente de Pareto. Além disso, arquivos externos com tamanhos muito grandes ou ilimitados podem tornar os algoritmos muito custosos computacionalmente tanto em termos de tempo

de processamento como de espaço. Sendo assim, uma investigação sobre a influência do tamanho do arquivo externo $|A|_{max}$ sobre a convergência do MaOPSO é deixada como trabalho futuro.

3.2.6 Considerações Finais

Este Capítulo apresentou a abordagem proposta nesta tese denominada MaOPSO. A ideia central da proposta gira em torno de dois pontos chaves: promoção de convergência e geração de diversidade. Esses dois requisitos são promovidos por meio de um processo de atribuição de aptidão que é utilizado para discriminar soluções não-dominadas. Esse processo de atribuição de aptidão por sua vez é utilizado na seleção dos líderes sociais do arquivo externo e na poda do arquivo externo.

O processo de atribuição de aptidão é baseado em duas medidas: a medida de densidade e a medida de convergência, que foram propostas pelo autor desta tese. Para se calcular essas duas medidas, um conjunto de pontos de referência é utilizado. Esses pontos servem para definir sub-regiões no espaço objetivo de modo que se espera que cada solução fique localizada em uma sub-região. Desse modo, a diversidade é assegurada. Esses mesmos pontos de referência servem como vetores de peso que são utilizados em um método denominado ASF que favorece soluções próximas da Frente de Pareto.

Desta forma, as principais contribuições desta tese são:

1. proposição de uma abordagem de pontos de referência em um algoritmo de enxame de partículas;
2. proposição de um esquema inovador para discriminação das soluções do arquivo externo usando duas medidas que são calculadas a partir dos pontos de referência;
3. proposição de dois esquemas distintos de atribuição dos líderes sociais;
4. proposição de sub-enxames para a exploração de soluções extremas do arquivo externo;
5. proposição de seleção por torneio na seleção dos líderes sociais cujo número de concorrentes é igual ao número de objetivos. Ou seja, proposição de seleção por torneio adaptativa.
6. proposição da seleção dos líderes cognitivos usando distância euclidiana.

4 Experimentos e Resultados

Este capítulo apresenta os detalhes do arranjo experimental usado para realização dos experimentos cujo objetivo foi avaliar e comparar o desempenho do MaOPSO em relação a outros algoritmos presentes na literatura. O arranjo experimental é apresentado na seção 4.1. Os resultados obtidos a partir dos experimentos com as funções de teste da família DTLZ são fornecidos e analisados na seção 4.2. Na seção 4.3 é apresentada uma aplicação da proposta desta tese em um problema de otimização com muitos objetivos do mundo real, a saber, o projeto de redes ópticas de alta capacidade. Finalmente, na seção 4.4 o capítulo é encerrado com as considerações finais.

4.1 Arranjo Experimental

Nesta seção, o arranjo experimental será apresentado detalhando os algoritmos usados para comparação (subseção 4.1.1), os problemas de teste comumente utilizados para avaliação de algoritmos meta-heurísticos multiobjetivos pela comunidade científica (subseção 4.1.2), as métricas comumente usadas para avaliação dos algoritmos (subseção 4.1.3) e as configurações dos algoritmos (subseção 4.1.4).

4.1.1 Algoritmos Utilizados Para Comparação

O MaOPSO é comparado com cinco algoritmos, a saber, SMPSO (NEBRO et al., 2009; NEBRO; DURILLO; COELLO, 2013), CDAS-SMPSO (DE CARVALHO; POZO, 2012), CEGA (GARZA-FABRE; PULIDO; Coello Coello, 2010), MDFA (GARZA-FABRE; PULIDO; Coello Coello, 2010) e o NSGA-III (DEB; JAIN, 2014). Esses algoritmos são representativos de algumas abordagens propostas na literatura para lidar com MOPs e MaOPs e por isso foram escolhidos. Com exceção do SMPSO, todos os outros algoritmos são algoritmos propostos originalmente para resolver problemas com muitos objetivos. Contudo, o SMPSO foi incluído para mostrar a superioridade do MaOPSO, que foi criado especialmente para problemas com muitos objetivos, frente a um algoritmo que é baseado em dominância de Pareto em cenários com muitos objetivos.

Antes de descrever cada um desses algoritmos, é necessário fazer alguns comentários de que porque esses algoritmos foram escolhidos, enquanto outros foram deixados de lado. Por exemplo, logo após a defesa da dissertação (FIGUEIREDO, 2013) em 2013, iniciou-se o desenvolvimento do MaOPSO a partir de ideias presentes em (DEB; JAIN, 2012; SINDHYA; MIETTINEN; DEB, 2013) e iniciou-se vários experimentos para avaliá-lo em vários cenários, utilizando-se os algoritmos que eram considerados o estado da arte a época, a saber, o

M DFA, CEGA, e o CDAS-SMPSO. Contudo, durante esse período surgiram paulatinamente outros algoritmos de otimização de muitos objetivos que seguiam a abordagem de pontos de referência proposta inicialmente por Deb e colaboradores em (DEB; JAIN, 2012). O primeiro algoritmo nesse sentido surgiu em 2014 e foi uma contribuição do próprio Deb junto com colaboradores (DEB; JAIN, 2014), e que consistiu numa evolução do algoritmo presente em (DEB; JAIN, 2012). Esse algoritmo foi chamado por Deb *et al.* de NSGA-III (DEB; JAIN, 2014) por se tratar de uma evolução do algoritmo baseado em dominância NSGA-II (DEB *et al.*, 2002). Como o NSGA-III era um algoritmo que partilha das mesmas ideias gerais que o MaOPSO, a comparação do MaOPSO com esse algoritmo era algo de extrema importância para a validação da proposta. De fato, dos algoritmos selecionados, o que mais se aproxima do algoritmo proposto (MaOPSO) é o algoritmo NSGA-III. Isto porque ambas as técnicas são algoritmos baseados em pontos de referência. Ou seja, eles utilizam um conjunto de pontos de referência para estabelecer um pressão de convergência para a Frente de Pareto e para oferecer uma maneira de distribuir os soluções de maneira aproximadamente uniforme ao longo da Frente de Pareto. Contudo, não havia nenhum código do NSGA-III disponível para ser usado e que pudesse ser utilizada como comparação com o MaOPSO. Desta forma, após a publicação do NSGA-III em 2014, o autor desta tese solicitou por *e-mail* o código-fonte do NSGA-III a todos os autores correspondentes do artigo (DEB; JAIN, 2014). Contudo, suas frequentes solicitações não foram respondidas. Em virtude disso, foi necessário desenvolver esse algoritmo por conta própria. Esse desenvolvimento devido a algumas faltas de informação presentes no próprio artigo levou bastante tempo acarretando numa grande perda de tempo de tempo e energia para autor. De fato, estima-se que autor levou seis meses no desenvolvimento deste algoritmo, uma vez que não basta apenas implementar o algoritmo mas é necessário validá-lo experimentalmente de modo que ele atingiram resultados estatisticamente similares aos do artigo (DEB; JAIN, 2014). Isso por fim provocou o atraso no desenvolvimento dos experimentos de comparação realizados nesta tese.

Após a publicação do NSGA-III em 2014, surgiram novos algoritmos evolucionários baseados fortemente no NSGA-III. De fato, a partir de 2015 houve um grande crescimento na publicação de artigos propondo algoritmos evolucionários para *Many-Objective Optimization Problems* (MaOPs) que seguiam o princípio de pontos referência do NSGA-III (DEB; JAIN, 2014). Entre esses algoritmos pode-se mencionar: o *Clustering-Ranking Evolutionary Algorithm* (crEA) (CAI *et al.*, 2015), o *Improved Decomposition-based Evolutionary Algorithm* (I-DBEA) (ASAFUDDOULA; RAY; SARKER, 2015), *Many-Objective Evolutionary Algorithm (MaOEA) based on Directional Diversity (DD) and Favorable Convergence (FC)* (MaOEA-DDFC) (CHENG; YEN; ZHANG, 2015), MOEA/DD (LI *et al.*, 2015b), θ -DEA (YUAN *et al.*, 2016). *Reference Vector guided Evolutionary Algorithm* (RVEA) (CHENG *et al.*, 2016a), *Vector angle-based Evolutionary Algorithm* (VaEA) (XIANG *et al.*, 2017). Além desses algoritmo, surgiram muitos outros que seguiam princípios diferentes, tais como: C-Multi-(JR.;

SANTANA; POZO, 2016), GrEA (YANG et al., 2013; LI et al., 2010), *Shift-based Density Estimation* (SDE) (LI; YANG; LIU, 2014a), *Two-Archive Algorithm* (Two_Arch) (WANG; JIAO; YAO, 2015). O algoritmo Two_Arch é particularmente interessante porque é baseado em ideia inovadora de usar dois tipos de arquivo no processo de seleção dos líderes das partículas, um arquivo para promoção de convergência e outro arquivo para geração de diversidade.

Novamente, devido a limitações de tempo não foi possível implementar esses algoritmos e todas as solicitações para envio de código-fonte dos algoritmos foram negadas ou não-respondidas. Por essa razão, todos esses algoritmos não foram incluídas nos experimentos de comparação com o MaOPSO. Sendo assim, a comparação do MaOPSO com alguns desses algoritmos é deixado como trabalhos futuros.

Um outro algoritmo que poderia ter sido selecionado para a comparação com o MaOPSO é outro algoritmo que foi proposto pelo autor desta tese em sua dissertação de Mestrado (FIGUEIREDO, 2013) e denominado de MOSPO-GD. Essa comparação com o algoritmo MOPSO-GD não foi realizada devido ao fato de que foi demonstrando em (FIGUEIREDO, 2013) que esse algoritmo, apesar de sua alta capacidade de convergência, apresenta grandes desvantagens em termos de diversidade. Sendo assim, ele não foi inserido nessa análise experimental. No entanto, essa comparação pode ser um trabalho futuro juntamente com a inclusão de outros algoritmos de enxames de partículas para otimização de muitos objetivos tais como o algoritmo C-Multi-(JR.; SANTANA; POZO, 2016) que proposto em 2016.

Sendo assim, diante de tudo que foi exposto, os algoritmos utilizados nos experimentos de comparação com o MaOPSO foram: (1) o SMPSO, para comparar o MaOPSO com um algoritmo multiobjetivo de enxame de partículas baseado em dominância de Pareto; (2) o CDAS-SMPSO, para comparar o MaOPSO com um algoritmo de enxame de partículas para otimização de muitos objetivos; (3) o CEGA e (4) o MDFA, que são dois algoritmos evolucionários para otimização de muitos objetivos baseados em agregação; e finalmente (5) o NSGA-III, um algoritmo evolucionário para otimização de muitos objetivos baseado pontos de referência. Todos esses algoritmos são todos descritos brevemente nas próximas subseções. Para mais informações o leitor é convidado a consultar as referências de cada algoritmo que foram colocadas oportunamente em cada uma das subseções que lhes descrevem.

4.1.1.1 Speed-constrained Multi-objective PSO (SMPSO)

O *Speed-constrained Multi-objective* PSO (SMPSO) (NEBRO et al., 2009) é um MOPSO baseado em dominância de Pareto que incorpora um mecanismo de restrição para limitar a velocidade máxima das partículas. A ideia da restrição de velocidade é melhorar a capacidade de convergência do algoritmo. SMPSO é baseado em um algoritmo

chamado OMPSO (REYES-SIERRA; COELLO, 2006) mas difere dele em alguns aspectos tais como a restrição da velocidade, um novo mecanismo de fronteira e no uso do operador de mutação (ver (NEBRO et al., 2009) para mais detalhes). Como mostrado em (NEBRO et al., 2009), SMPSO é superior a outros algoritmos, em termos de convergência e diversidade de soluções, em problemas multiobjetivos com dois ou três objetivos.

4.1.1.2 SMPSO with Control of Dominance Area of Solutions (CDAS)

O *Control of Dominance Area of Solutions* (CDAS) é uma forma de dominância relaxada proposta por Sato *et al.* (SATO; AGUIRRE; TANAKA, 2007) para lidar com problemas com muitos objetivos. Esta técnica pode controlar o grau de expansão ou contração da área de dominância usando o parâmetro S definido pelo usuário. A desvantagem desta técnica é que o valor ótimo de S depende fortemente do número de objetivos, tamanho do espaço de busca, e complexidade do problema (SATO; AGUIRRE; TANAKA, 2007). Em (DE CARVALHO; POZO, 2012), De Carvalho e Pozo em 2012 propuseram usar a técnica CDAS no SMPSO ao invés da dominância de Pareto. O algoritmo resultante foi chamado *SMPSO with Control of Dominance Area of Solutions* (CDAS-SMPSO) e foi aplicado em problemas com muitos objetivos como pode ser constatado em (DE CARVALHO; POZO, 2012). A desvantagem dessa técnica é a determinação do parâmetro S do CDAS. De fato, foi necessário fazer vários experimentos tediosos em (DE CARVALHO; POZO, 2012) para se determinar o valor mais apropriado desse parâmetro.

4.1.1.3 Clustering-based Elitist Genetic Algorithm - CEGA

O *Clustering-based Elitist Genetic Algorithm* (CEGA) (GARZA-FABRE; PULIDO; Coello Coello, 2010) consiste em um MOEA proposto para lidar com problemas com muitos objetivos. CEGA é um algoritmo baseado em agregação que apresenta dois elementos chaves: (i) um método de atribuição de aptidão para promover a convergência, e (ii) um mecanismo explícito para promover a diversidade baseado em um algoritmo de agrupamento. O método de atribuição designa a cada indivíduo uma *aptidão* para concorrer na etapa de *seleção para sobrevivência*. O método de atribuição é chamado *detrimento global* e foi proposto pelos mesmos autores do CEGA em (Garza Fabre, 2009). O *detrimento global* é um método de atribuição de aptidão baseado em distâncias objetivo a objetivo entre o indivíduo e os outros indivíduos da população. A estratégia de agrupamento responsável por guiar a *seleção-para-sobrevivência* é baseado em um procedimento de agrupamento aglomerativo hierárquico. Isto significa que cada indivíduo na população inicialmente representa um *cluster* diferente e, iterativamente, os dois *clusters* mais similares são combinados até um número fixo de grupos ser atingido. O propósito de agrupar as soluções é guiar a seleção para um conjunto bem distribuído de indivíduos para promover a diversidade e melhorar a capacidade de exploração do CEGA. A desvantagem dessa técnica é seu

alto custo computacional em virtude do seu procedimento de agrupamento aglomerativo hierárquico (GARZA-FABRE; PULIDO; Coello Coello, 2010).

4.1.1.4 Multi-Directional Fitness Assignment - MDFA

O *Multi-Directional Fitness Assignment* (MDFA) (GARZA-FABRE; PULIDO; Coello Coello, 2010) é um MOEA proposto para lidar com problemas com muitos objetivos. MDFA é um algoritmo baseado em agregação e ele usa um mecanismo diferente do CEGA para promover a convergência. Este mecanismo tem como finalidade guiar o processo de busca simultaneamente em múltiplas direções usando um conjunto de vetores de peso para orientar a busca em diferentes direções do espaço objetivo. Um vetor de peso é um conjunto de coeficientes que denotam a importância de cada objetivo de modo que diferentes vetores de peso direcionarão a busca para diferentes regiões da Frente de Pareto. Estes vetores de peso são usados no método de atribuição do MDFA como segue. Inicialmente, todos os indivíduos na população são avaliados para cada vetor de peso usando a soma ponderada. Então, o indivíduo com o melhor desempenho para cada vetor de peso é identificado e o valor da aptidão obtida é atribuído a ele. No MDFA, o desempenho ou aptidão de um indivíduo é o valor resultante de avaliar o indivíduo com respeito ao vetor de peso usando a soma ponderada como função de agregação. Como o mesmo indivíduo pode ser o melhor para vários vetores de peso, sua aptidão final será o melhor de todos os valores de aptidão obtido para estes vetores objetivos. Indivíduos que ainda não possuem aptidão usam o pior desempenho para todos os vetores de peso como seu valor de aptidão. Isto permitirá a preferência de soluções que apresentam um melhor desempenho médio para o conjunto de pesos dados. Além disso, a aptidão destes indivíduos é penalizada pela adição da pior aptidão atribuída no primeiro estágio. Isto garante que estes indivíduos serão menos preferidos que o primeiro conjunto de indivíduos. A desvantagem do MDFA é que como ela usa uma soma ponderada em seu método de atribuição de aptidão ela não consegue produzir todas as soluções de um problema com Frente de Pareto não-convexa (Coello Coello; LAMONT; VELDHUIZEN, 2007; ADRA, 2007; FIGUEIREDO, 2013; ZITZLER, 1999).

4.1.1.5 NSGA-III

O *Nondominated Sorting Genetic Algorithm III* (NSGA-III) (DEB; JAIN, 2014) é um algoritmo baseado em conjunto de referência proposto por Deb e Jain (DEB; JAIN, 2014) baseado no *Nondominated Sorting Genetic Algorithm II* (NSGA-II) e similar a outro algoritmo proposto pelos mesmos autores em 2012 chamado *Many-Objective NSGA-II* (MO-NSGA-II) (DEB; JAIN, 2012). NSGA-III fornece um conjunto de pontos de referência bem distribuídos sobre um hiperplano no espaço objetivo com interseptos nos eixos objetivos. Em cada geração, as soluções são normalizadas e associadas com um ponto de referência baseado em sua distância perpendicular para a linha de referência. Portanto, agrupamentos de soluções são formados ao redor de cada ponto de referência e o número

de soluções em cada agrupamento é usado para estimar a densidade. Assim, a informação de densidade é usada na seleção ambiental escolhendo indivíduos em regiões menos densas para a próxima geração.

4.1.2 Problemas de Teste

Para a realização dos experimentos, usa-se um conjunto de seis problemas de teste da família DTLZ, a saber, os problemas DTLZ{1,2,3,4,5,6} (DEB et al., 2005), que pode ser escalado para qualquer número de objetivos m e qualquer número de variáveis de decisão n . O número de variáveis de decisão é definido por $n = M + k - 1$, em que k é um parâmetro que estipula a dificuldade na resolução do problema. O problema DTLZ1 contém $11^k - 1$ Frentes de Pareto locais onde um algoritmo de otimização pode parar antes de atingir a Frente Ótima de Pareto. DTLZ3 é um problema multi-modal difícil que possui $3^k - 1$ Frentes de Pareto locais e uma Frente de Pareto global, e também é um problema não-linear, o que torna esse problema em um dos mais desafiadores e complexos da família DTLZ em termos de convergência (DEB et al., 2005; Coello Coello; LAMONT; VELDHUIZEN, 2007). DTLZ4 é um problema difícil em termos de diversidade e DTLZ6 testa a habilidade do algoritmo de convergir para uma Frente de Pareto uni-dimensional em um espaço m -dimensional. DTLZ{2,5} são versões simples dos problemas DTLZ{3,6}, respectivamente. Nesta tese, assim como em outros trabalhos (DEB et al., 2005; GARZA-FABRE; PULIDO; Coello Coello, 2010; Garza Fabre, 2009), o parâmetro de dificuldade k foi definido como $k = 5$ pra o problema DTLZ1 e $k = 10$ para os problemas restantes. Os valores deste parâmetro são os mesmos que já foram usados extensivamente em outros trabalhos (Coello Coello; LAMONT; VELDHUIZEN, 2007) e foram sugeridos pelos próprios autores dos problemas (DEB et al., 2005).

4.1.3 Métricas de Qualidade

Para avaliar o desempenho dos algoritmos, usa-se três métricas: Distância Geracional, Distância Geracional Invertida, e Hipervolume. A Distância Geracional é uma métrica para avaliar a convergência. A Distância Geracional Invertida e o Hipervolume por sua vez são métricas utilizadas para avaliar tanto a convergência e a diversidade. Como se pode perceber, não foi utilizada nenhuma métrica para avaliar apenas a diversidade. Isso se explica pelo fato de que é muito difícil desenvolver uma métrica que avalie o aspecto de diversidade de modo isolado em espaços multidimensionais. De fato, as poucas métricas que existem trazem muita pouca informação a respeito da cobertura e distribuição das soluções ao longo da Frente de Pareto. Sendo assim, a Distância Geracional, Distância Geracional Invertida, e Hipervolume foram as únicas métricas utilizadas nos experimentos, métricas essas que são recorrentes na literatura da área (GARZA-FABRE; PULIDO; Coello

Coello, 2010; CAI et al., 2015; ASAFUDDOULA; RAY; SARKER, 2015; LI et al., 2015a). Estas métricas são discutidas a seguir:

- A Distância Geracional ou *Generational Distance* (GD) é definida pela Equação 4.1.

$$GD(PF) = \frac{1}{|PF|} \left(\sqrt{\sum_{i=1}^{|PF|} d_i^2} \right), \quad (4.1)$$

em que PF é o conjunto aproximação, isto é, o conjunto de soluções não-dominadas retornado pelo algoritmo. O termo d_i é a distância Euclidiana entre a i -ésima solução em PF e o ponto mais próximo na Frente de Pareto.

Portanto, GD provê uma métrica da proximidade do conjunto aproximação PF e a Frente de Pareto, com um menor valor indicando uma maior proximidade. Os problemas DTLZ{1,2,3,4} têm Frentes de Pareto que podem ser descritas analiticamente de modo que a distância geracional pode ser calculada diretamente. Para os problemas DTLZ{5,6}, foi usado uma amostragem da Frente de Pareto para formar um conjunto de referência P^* como o usado na métrica *Inverted Generational Distance* (IGD) e calcula-se a distância d_i para este conjunto ao invés de calculá-la para a Frente de Pareto.

- A Distância Geracional Invertida ou *Inverted Generational Distance* (IGD) é uma medida que permite avaliar simultaneamente a convergência e a diversidade do conjunto aproximação. O IGD é definido pela Equação 4.2.

$$IGD(PF) = \left(\sqrt{\sum_{i=1}^{|P^*|} d_i^2} \right) / |P^*|, \quad (4.2)$$

em que P^* é um conjunto de referência de pontos na Frente de Pareto e d_i é a distância Euclidiana entre a i -ésima solução em P^* e o ponto mais próximo no conjunto aproximação PF obtido pelo algoritmo.

IGD requer um conjunto de referência P^* de pontos que consiste em uma amostragem aproximadamente uniforme de toda Frente de Pareto. Consequentemente, é necessário criar este conjunto para todos os problemas DTLZ. Isto é feito como segue. A Frente de Pareto de DTLZ1 é um hiperplano e para este problema usa-se o esquema de geração de pontos de referência sobre o hiperplano apresentado na subseção 3.2.1.1. As Frentes de Pareto de DTLZ2, DTLZ3, e DTLZ4 são hiperesferas; portanto os pontos de interseção entre a hiperesfera e as linhas conectando a origem e os pontos sobre o hiperplano da subseção 3.2.1.1 produzem o conjunto de referência para esses problemas. Nesta tese, usa-se um conjunto de referência cujo tamanho é determinado pelo número de pontos de referência H usado nos algoritmos. Para os problemas

DTLZ5 e DTLZ6 foi usado um esquema diferente porque as Frentes de Pareto destes problemas são curvas uni-dimensionais. Desta forma, para esses problemas realizou-se uma amostragem uniforme de 1000 pontos sobre a Frente de Pareto independente do número de objetivos.

- O Hipervolume (HV) calcula o volume do espaço objetivo fechado pelo conjunto aproximação retornado pelo algoritmo e um ponto de referência. Assim, ele é definido por

$$HV(PF) = \mu\left(\bigcup_{z_a \in PF} \{z | z_a \prec z \prec z_{ref}\}\right), \quad (4.3)$$

em que μ é a medida de Lebesgue, PF é o conjunto aproximação, e z_{ref} é o ponto de referência. HV mede tanto convergência e diversidade de um conjunto aproximação (SINGH; ISAACS; RAY, 2011) (LI et al., 2015a), de modo que quanto maior for o HV, maior é convergência e a diversidade do conjunto aproximação. Para calcular o hipervolume, um ponto de referência é necessário. Neste artigo, usa-se a mesma metodologia apresentada em (YANG et al., 2013) para determinar este ponto de referência. Desta maneira, considera-se o ponto de referência como o vetor em que cada componente objetivo corresponde ao pior valor encontrado naquele objetivo considerando todos os conjuntos aproximação obtidos em todas execuções dos algoritmos. Os valores de HV apresentados neste trabalho são normalizados entre $[0, 1]$ usando o melhor e o pior HV obtido entre as execuções dos algoritmos para uma instância particular de cada problema. O método utilizado para o cálculo do hipervolume foi o método exato propositado em (WHILE; BRADSTREET; BARONE, 2012).

4.1.4 Delineamento Experimental e Configurações do Parâmetros dos Algoritmos

Como já foi mencionado, os algoritmos utilizados para comparação foram o CEGA, MDFA, SMPSO, CDAS-SMPSO e NSGA-III, e os experimentos realizados neste trabalho foram conduzidos da seguinte maneira. Todos os algoritmos foram executados 31 vezes produzindo, cada algoritmo, 31 conjuntos aproximações diferentes. Utilizou-se 31 execuções porque nas análises de desempenho do algoritmo a mediana foi bastante utilizada como estatística de comparação de desempenho. Esse valor também foi utilizado em outros trabalhos, tais como (WANG; PURSHOUSE; FLEMING, 2013c; Garza Fabre, 2009; GARZA-FABRE; PULIDO; Coello Coello, 2010; Garza Fabre; PULIDO; Coello Coello, 2009).

Os algoritmos foram avaliados utilizando os problemas da família DTLZ com 2, 3, 5, 7, e 10 objetivos. A Tabela 1 apresenta os números de objetivos m utilizados nos experimentos, o número de divisões por objetivo p para cada número de objetivo, o número

de pontos de referência $|\Lambda|$ criados a partir de p , o tamanho máximo do arquivo externo $|A|_{max}$, e N é o tamanho da população, isto é, o número de indivíduos ou partículas.

Como se pode observar pela Tabela 1, o número de pontos de referência cresce fatorialmente com o número de objetivos. Esse crescimento aumenta o custo computacional do MaOPSO e do NSGA-III, o que poderia impedir a execução dos experimentos em tempo hábil. Desta forma, ao contrário do artigo do NSGA-III (DEB; JAIN, 2014), não se utilizou 15 objetivos nos experimentos devido ao grande número de pontos de referência que esse valor gerava mesmo para um número de divisões p pequeno. Esse fato justificou, desta forma, a exclusão de 15 objetivos dos experimentos.

Tabela 1 – Parâmetros usados nos algoritmos de acordo com número de objetivos.

| m | p | $ \Lambda $ | $ A _{max}$ | N |
|-----|-----|-------------|-------------|-----|
| 2 | 200 | 200 | 200 | 100 |
| 3 | 20 | 231 | 231 | 100 |
| 5 | 8 | 495 | 495 | 100 |
| 7 | 6 | 924 | 924 | 200 |
| 10 | 5 | 2002 | 500 | 300 |

Todos os algoritmos utilizaram a mesma quantidade de avaliações da função de aptidão de maneira a permitir uma comparação justa entre os algoritmos. A única exceção foi o algoritmo CEGA em que um número muito menor de avaliações foi utilizado. Isto porque o CEGA emprega um técnica de agrupamento em cada iteração o que faz com que o uso de muitas iterações em cenários com muitos objetivos se torne inviável. Sendo assim, os algoritmos MaOPSO, MDFA, SMP SO, CDAS-SMPSO e NSGA-III foram executados usando 100 000 iterações, enquanto que o CEGA foi executado sempre com 300 iterações. Esse número de iterações foi utilizado no MaOPSO porque ele apresentou um comportamento de convergência até esse número de iterações (como pode ser verificado na subseção 4.2.1). A Tabela 2 apresenta o número de iterações e o número de avaliações de aptidão para cada algoritmo de acordo com o número m de objetivos do problema. Nesta tabela, m é o número de objetivos utilizados, NI significa o número de iterações, N é o tamanho da população usado em cada algoritmo, e NAA é o número de avaliações de aptidão. Como se pode perceber pela tabela, todos os algoritmos com exceção do CEGA usaram o mesmo número de avaliações de aptidão.

O tamanho da população utilizado em todos os algoritmos também foi o mesmo dependendo do número de objetivos e novamente o CEGA foi a única exceção pela mesma razão explicada anteriormente, ou seja, devido ao seu alto custo computacional. O tamanho da população usado nos algoritmos varia dependendo do número de objetivos do problema e é apresentado na Tabela 1. A Tabela 1 apresenta em cada coluna, da esquerda para a direita, o número de objetivos (Obj), o número de divisões (p) usados

em cada objetivo para construir os pontos de referência, o tamanho do arquivo externo ($|A|_{max}$) usado para cada número de objetivo, e o tamanho da população usado para cada número de objetivo. Como se pode perceber pela tabela, o CEGA foi executado usando 100 indivíduos independentemente do número de objetivos por razões já mencionadas. Os valores utilizados como tamanho da população, número de pontos de referência, e tamanho do arquivo externo foram definidos empiricamente de maneira a fazer com que o MaOPSO atingisse a convergência. Os valores escolhidos para o tamanho da população não poderiam ser os mesmos que os utilizados no NSGA-III como realizado em (DEB; JAIN, 2014) porque isso poderia demandar um alto custo computacional em termos de tempo de processamento, o que poderia atrasar os experimentos e conseqüentemente o desenvolvimento deste doutorado. Sendo assim, valores razoáveis de tamanho de população, como na Tabela 1, e que permitiam a execução dos experimentos em tempo hábil foram utilizados.

Com respeito ao ambiente de execução dos algoritmos, todos os algoritmos foram executados utilizando um computador com sistema operacional Windows 10 Home Single Language (Sistema Operacional de 64 bits) Versão 1607 com memória RAM 8,00 GB, com processador Intel(R) Core(TM) i7-4500U CPU @ 1,80 GHz 2,40 GHz. Os algoritmos CEGA, MDFA, NSGA-III foram implementados pelo autor e validados de acordo com os resultados dos seus respectivos artigos. Para o algoritmo SMPSO foi utilizado a implementação do *framework* JMetal (DURILLO; NEBRO, 2011), que é um *framework* de MOEAs implementando em Java e bem conhecido na comunidade científica. A implementação do CDAS-SMPSO foi enviada pelo próprio autor do algoritmo e também estava escrita em Java. Sendo assim, todos os algoritmos utilizados para comparação foram implementados em Java permitindo assim uma maior uniformização na comparação dos algoritmos.

Finalmente, os demais valores dos parâmetros que são particulares de cada algoritmo são apresentados nas próximas subseções.

4.1.4.1 Algoritmo Proposto: MaOPSO

Os parâmetros usados no MaOPSO são apresentados a seguir. Os coeficiente c_1 e c_2 são iguais a 1,49618 e ω é igual a 0,729844. Os parâmetros foram escolhidos baseados na análise realizada por Clerc e Kennedy (CLERC; KENNEDY, 2002). A Tabela 1 apresenta os valores de p utilizados no MaOPSO, que é o número de divisões, e o respectivo número de pontos de referência gerado por esse número de divisões. Como se pode observar, o valor de p variou dependendo do número de objetivos e para objetivos maiores ele apresentou valores menores em virtude da natureza combinatorial do número de pontos de referência gerados. A Tabela 1 também apresenta o tamanho do arquivo externo $|A|_{max}$. Conforme se pode verificar na Tabela 1, o tamanho do arquivo externo do MaOPSO varia de acordo com o número de objetivos e seu tamanho é igual ao número de pontos de referência. No

Tabela 2 – Número de iterações e avaliações de aptidão por algoritmo e para cada objetivo.

| Algoritmo | m | NI | N | NAA | Algoritmo | m | NI | N | NAA |
|-----------|-----|---------|-----|------------------|------------|-----|---------|-----|------------------|
| MaOPSO | 2 | 100 000 | 100 | 10×10^6 | SMPSO | 2 | 100 000 | 100 | 10×10^6 |
| | 3 | 100 000 | 100 | 10×10^6 | | 3 | 100 000 | 100 | 10×10^6 |
| | 5 | 100 000 | 100 | 10×10^6 | | 5 | 100 000 | 100 | 10×10^6 |
| | 7 | 100 000 | 200 | 20×10^6 | | 7 | 100 000 | 200 | 20×10^6 |
| | 10 | 100 000 | 300 | 30×10^6 | | 10 | 100 000 | 300 | 30×10^6 |
| NSGA-III | 2 | 100 000 | 100 | 10×10^6 | CDAS-SMPSO | 2 | 100 000 | 100 | 10×10^6 |
| | 3 | 100 000 | 100 | 10×10^6 | | 3 | 100 000 | 100 | 10×10^6 |
| | 5 | 100 000 | 100 | 10×10^6 | | 5 | 100 000 | 100 | 10×10^6 |
| | 7 | 100 000 | 200 | 20×10^6 | | 7 | 100 000 | 200 | 20×10^6 |
| | 10 | 100 000 | 300 | 30×10^6 | | 10 | 100 000 | 300 | 30×10^6 |
| MDFA | 2 | 100 000 | 100 | 10×10^6 | CEGA | 2 | 300 | 100 | 30 000 |
| | 3 | 100 000 | 100 | 10×10^6 | | 3 | 300 | 100 | 30 000 |
| | 5 | 100 000 | 100 | 10×10^6 | | 5 | 300 | 100 | 30 000 |
| | 7 | 100 000 | 200 | 20×10^6 | | 7 | 300 | 100 | 30 000 |
| | 10 | 100 000 | 300 | 30×10^6 | | 10 | 300 | 100 | 30 000 |

caso de 10 objetivos, devido a limitações computacionais de tempo, o tamanho do arquivo externo foi limitado a 500 soluções. O MaOPSO usou uma mutação polinomial com um índice de distribuição de mutação $\eta_m = 20$ com probabilidade $1/n$. Estes valores para o operador de mutação foram estabelecidos tomando o SMPSO como referência (NEBRO et al., 2009). Devido às restrições de tempo deste doutorado e também devido à complexidade que envolveu o desenvolvimento do MaOPSO, um estudo mais aprofundado sobre o efeito do índice de distribuição de mutação η_m sobre o desempenho do MaOPSO e dos outros parâmetros do operador de mutação não foi realizado. Esse estudo é deixado como trabalho futuro.

4.1.4.2 SMPSO & CDAS-SMPSO

Para SMPSO e CDAS-SMPSO, usa-se os mesmos parâmetros usados no artigo original (ver (NEBRO et al., 2009) e (DE CARVALHO; POZO, 2012), respectivamente, para mais detalhes). Nestes algoritmos, o tamanho do arquivo externo é igual a 100, conforme (NEBRO et al., 2009) e (DE CARVALHO; POZO, 2012). O parâmetro S usado no CDAS-SMPSO é definido como 0,25 porque nos experimentos realizados neste trabalho, foi esse valor de S que produziu o melhor desempenho em termos de convergência. Convergência é um requisito que vem antes da diversidade em qualquer algoritmo multiobjetivo (Coello Coello; LAMONT; VELDHUIZEN, 2007) (inclusive no MaOPSO), e ele foi usado para definir o valor de S para as análises comparativas.

4.1.4.3 CEGA & MDFA

No CEGA e MDFA, usa-se os mesmos parâmetros usados no artigo original (ver (GARZA-FABRE; PULIDO; Coello Coello, 2010) para mais detalhes). No CEGA, foi usado uma população com 100 indivíduos e 300 iterações conforme (GARZA-FABRE; PULIDO; Coello Coello, 2010) devido a limitações de tempo (CEGA usa um algoritmo de agrupamento computacionalmente custoso).

4.1.4.4 NSGA-III

No NSGA-III, usa-se os mesmos parâmetros usados em seu artigo original para os operadores de cruzamento e mutação (ver (DEB; JAIN, 2014) para mais detalhes). Para uma comparação justa com o MaOPSO, o número de pontos de referência usado no NSGA-III é o mesmo que o MaOPSO e é dado pela Tabela 1.

4.2 Resultados nas Funções de Teste da Família DTLZ

Esta seção apresenta os resultados dos experimentos. A subseção 4.2.1 apresenta uma comparação entre o MaOPSO e o SMPSO em termos de convergência. Esta subseção tem como objetivo analisar o processo de convergência do MaOPSO. Na subseção 4.2.2, os resultados para as métricas de qualidade são apresentados na forma de *box-plots* com o objetivo de comparar os algoritmos em termos de convergência e diversidade, quando esses algoritmos são executados usando 100 000 iterações. A subseção 4.2.3 realiza o mesmo procedimento, mas a análise da convergência e diversidade dos algoritmos é feita usando um número limitado e reduzido de 1 000 iterações. A subseção 4.2.4 apresenta uma análise estatística sobre os resultados das métricas dos algoritmos para salientar a significância dos resultados obtidos pelo MaOPSO. Finalmente, a subseção 4.2.5 apresenta alguns exemplos de conjuntos aproximação produzidos pelos algoritmos.

4.2.1 Análise de Convergência

Esta seção apresenta uma análise da convergência do MaOPSO ao longo das iterações. A métrica GD foi usada para medir a convergência. Esta seção apresenta apenas os resultados para os problemas $DTLZ\{1,2,3,4,5,6\}$ com 10 objetivos. Desta forma, buscou-se avaliar a convergência do MaOPSO no cenário com o maior número de objetivos.

As Figuras de 14 a 19 mostram o GD mediado sobre 31 execuções para os algoritmos MaOPSO e SMPSO ao longo das iterações. Em outras palavras, essas figuras mostram a média do GD sobre 31 execuções iteração a iteração. Optou-se por colocar o número iterações ao invés do número de avaliações de aptidão pois os dois algoritmos utilizam

uma população de igual tamanho e que, por isso, utilizam o mesmo número de avaliações de aptidão por iteração.

O SMPSO foi usado como um algoritmo de referência para efeitos de comparação porque ele é um dos melhores *Multiobjective Particle Swarm Optimizations* (MOPSOs) baseados em dominância de Pareto (DE CARVALHO; POZO, 2012). O objetivo desta análise foi apenas demonstrar a efetividade do mecanismo de atribuição de aptidão em termos de convergência do MaOPSO sobre um MOPSO baseado em dominância de Pareto. Por isso, os outros algoritmos tais como NSGA-III, CDAS-SMPSO, CEGA e o MDFA não foram consideradas nesta análise. Nas próximas seções, uma comparação da convergência e da diversidade entre todos os algoritmos é realizada.

No problema DTLZ1 (Figura 14), é notório o comportamento convergente do MaOPSO. Nas primeiras 30 000 iterações ele converge rapidamente para uma região muito próxima da Frente de Pareto para logo em seguida refinar a busca e finalmente atingir a convergência. O SMPSO, por sua vez, tem uma tendência de convergência mas esse processo ocorre bastante lentamente.

A análise do problema DTLZ2 (Figura 15) mostra a principal diferença entre o método de atribuição de aptidão do MaOPSO e o método baseado em dominância de Pareto do SMPSO. Como se pode observar nesta figura, o MaOPSO converge suavemente enquanto que o SMPSO permanece estagnado em uma região longe da Frente de Pareto.

No problema DTLZ3 (Figura 16), o MaOPSO converge rapidamente nas primeiras 30000 iterações, provavelmente para uma região próxima da Frente de Pareto. Em seguida, o GD aumenta um pouco para logo em seguida diminuir e estagnar em um valor constante indicando a convergência do algoritmo. O aumento do GD provavelmente ocorre devido ao aumento da diversidade de soluções logo após uma fase de convergência inicial. Este aumento da diversidade, por sua vez, é oriundo provavelmente do mecanismo de geração de diversidade do MaOPSO, que é a mutação polinomial, produzindo um aumento de soluções não-dominadas no arquivo externo. Por outro lado, a etapa seguinte consiste em um refinamento da convergência dessas soluções para Frente de Pareto. O valor pequeno do GD indica que provavelmente o MaOPSO convergiu para a Frente de Pareto. Por outro lado, o progresso de convergência do SMPSO é muito lento devido ao mecanismo de seleção do líder que é baseado na relação de dominância de Pareto. Como as soluções produzidas são quase sempre não-dominadas, o SMPSO não consegue impor uma pressão de seleção para a Frente de Pareto. Portanto, este resultado mostra que o método de atribuição de aptidão usado no MaOPSO é mais efetivo que a método usado no SMPSO para problemas com muitos objetivos. É importante ressaltar que o problema DTLZ3 é o problema mais difícil em termos de convergência da família DTLZ (DEB et al., 2005), o que torna os resultados obtidos pelo MaOPSO nesse problema muito promissores.

A análise do problema DTLZ4 (Figura 17) apresenta um fato interessante: o SMPSO

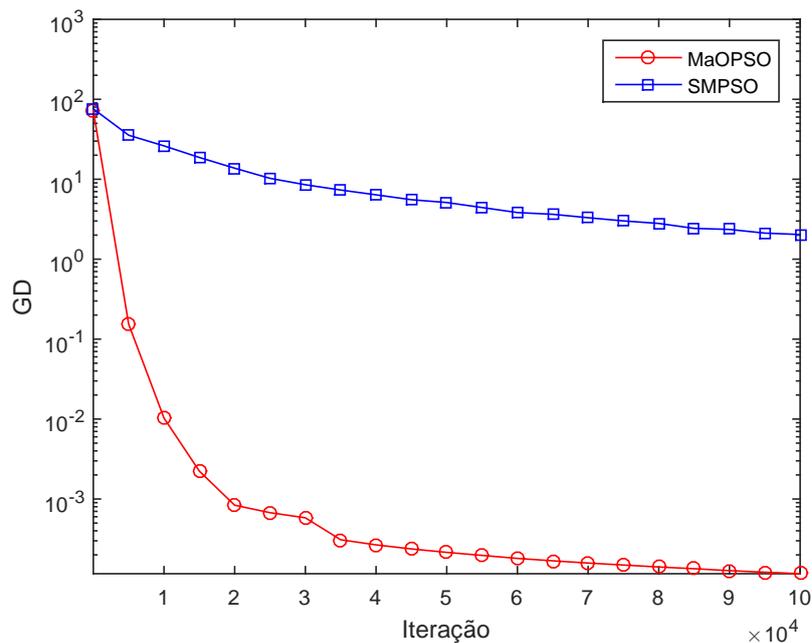


Figura 14 – Resultados de GD em função do número de iterações para o problema DTLZ1 com 10 objetivos.

apresenta um comportamento divergente ao invés de convergente. Isso ocorre devido ao fato de o SMPSO utilizar o operador *crowding distance* como operador de densidade. Devido a forma como o *crowding distance* é calculado, soluções não-dominadas, mas afastadas de suas soluções vizinhas, são privilegiadas. Desta forma, o algoritmo acaba favorecendo soluções afastadas da Frente de Pareto o que explica o comportamento divergente do algoritmo. O MaOPSO, por sua vez, converge suavemente nesse problema.

No problema DTLZ5 (Figura 18), o SMPSO também apresenta inicialmente um comportamento divergente e logo em seguida estagna em alguma região do espaço objetivo. O MaOPSO tem uma rápida convergência inicial nas primeiras 5000 iterações e em seguida ele estagna em uma região próxima a Frente de Pareto.

No problema DTLZ6 (Figura 19), o SMPSO também apresenta inicialmente um comportamento um pouco divergente e logo em seguida estagna em alguma região do espaço objetivo. O MaOPSO tem uma rápida convergência inicial nas primeiras 5000 iterações e em seguida ele estagna em uma região próxima a Frente de Pareto.

4.2.1.1 Considerações Finais Sobre Análise de Convergência

A partir das análises feitas sobre cada problema, pode-se concluir que o MaOPSO conseguiu convergir para a Frente de Pareto ou para uma região próxima a ela em todos os problemas considerados. Além disso, os resultados indicam que o número de iterações usado (100 000) é um número suficiente para promover a convergência em muitas das execuções do algoritmo em todos os problemas avaliados. Nas próximas seções, é realizada

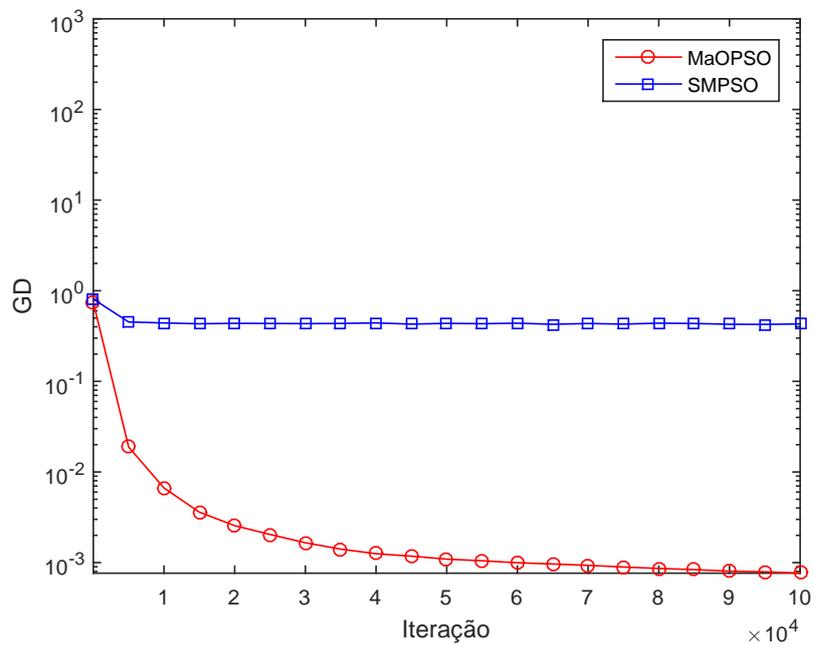


Figura 15 – Resultados de GD em função do número de iterações para o problema DTLZ2 com 10 objetivos.

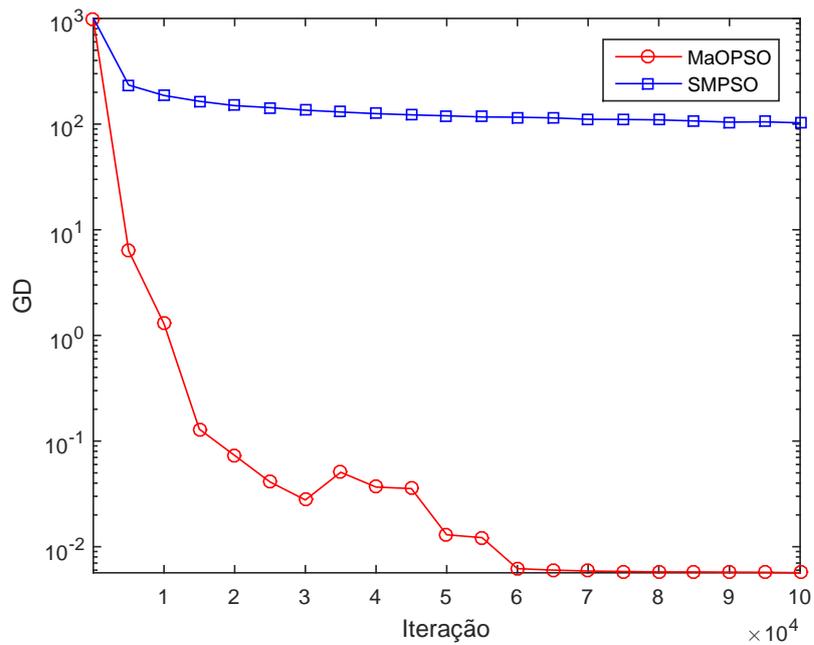


Figura 16 – Resultados de GD em função do número de iterações para o problema DTLZ3 com 10 objetivos.

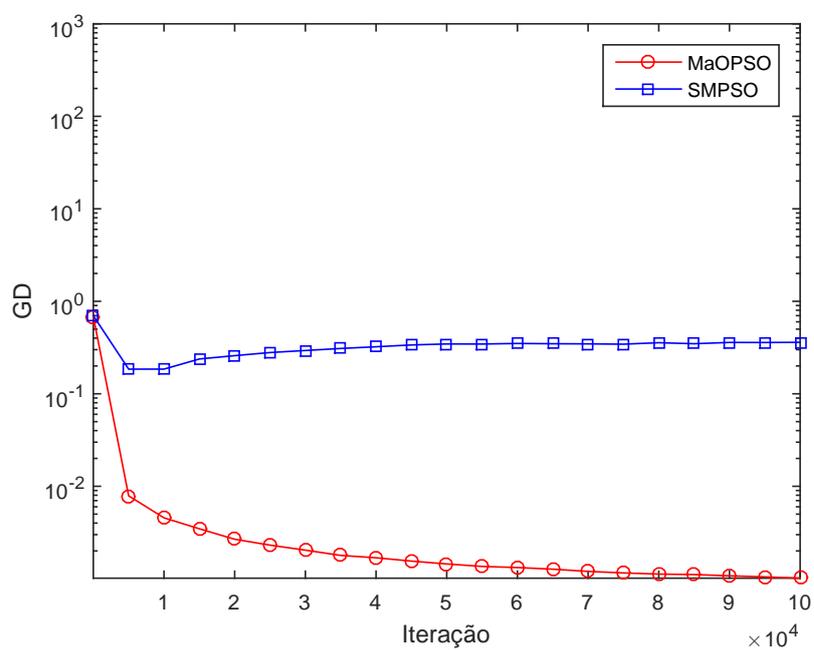


Figura 17 – Resultados de GD em função do número de iterações para o problema DTLZ4 com 10 objetivos.

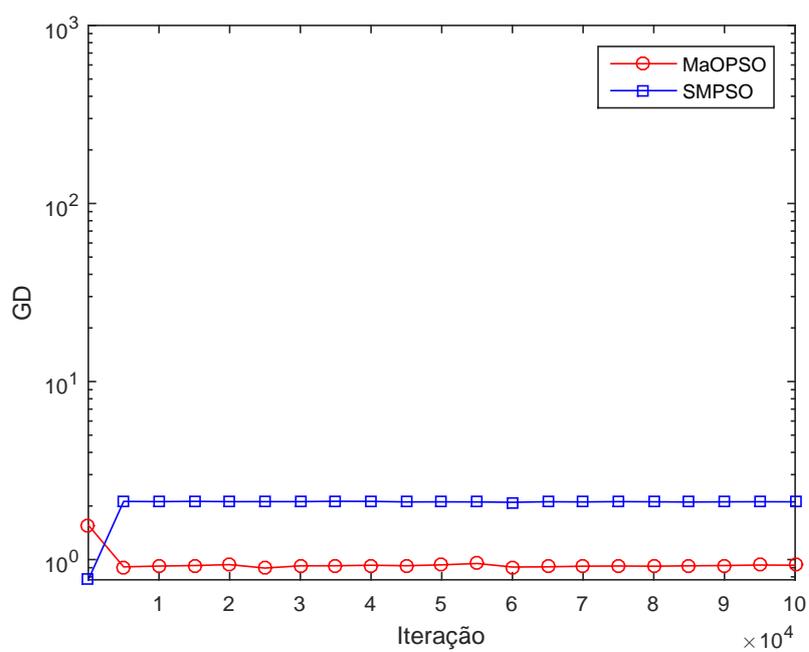


Figura 18 – Resultados de GD em função do número de iterações para o problema DTLZ5 com 10 objetivos.

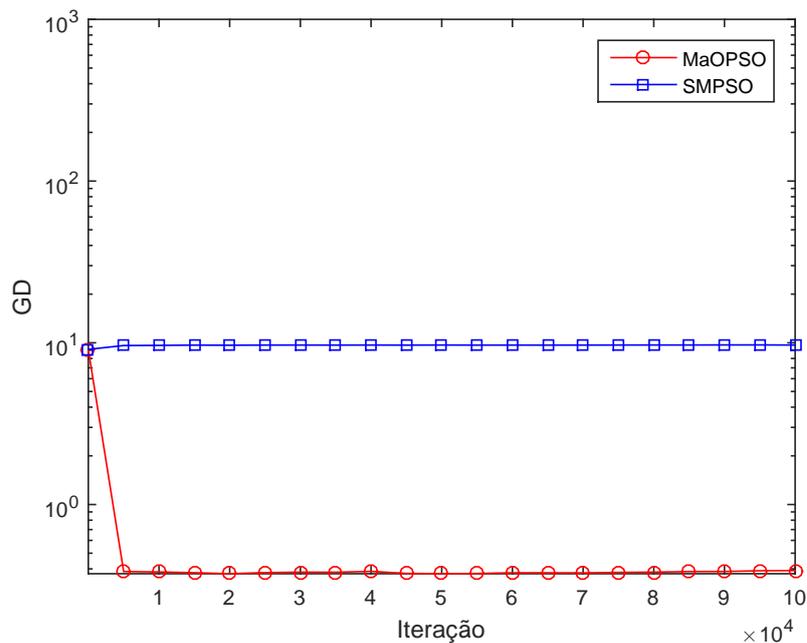


Figura 19 – Resultados de GD em função do número de iterações para o problema DTLZ6 com 10 objetivos.

a análise da convergência final (GD) e da diversidade final (IGD e HV) usando esse número de iterações, isto é, 100 000 iterações (subseção 4.2.2). Contudo, 100 000 iterações é um valor muito alto de iterações o que poderia tornar a aplicação do MaOPSO ineficaz em problemas práticos. Desta forma, é necessário avaliar o desempenho do MaOPSO usando um número limitado e muito menor de iterações. Sendo assim, na subseção 4.2.3 realizou-se a mesma análise da subseção 4.2.3, mas usando apenas 1 000 iterações. Isto é, efetuou-se a análise da convergência final (GD) e da diversidade final (IGD e HV) de todos os algoritmos em todos os problemas avaliados usando somente 1 000 iterações. Em suma, na subseção 4.2.2, realiza-se uma análise do desempenho dos algoritmos após um número de iterações (100 000) que permite a convergência do MaOPSO (100 000 iterações). E na subseção 4.2.3, realiza-se a análise do desempenho dos algoritmos após um número pequeno e razoável de iterações (1 000 iterações), de maneira que se possa estabelecer quais algoritmos são melhores utilizando um número limitado e reduzido de iterações e de avaliações de aptidão.

4.2.2 Resultados para GD, IGD, e Hipervolume para 100 000 iterações

Nesta seção, os algoritmos são comparados em termos de GD, IGD e HV usando *box-plots*. As Figuras 20 a 25 mostram o comportamento de convergência dos algoritmos, e Figuras 26 a 31 mostram os resultados relativos a métrica IGD. Todos os resultados para GD e IGD são apresentados como *box-plots* com escala logarítmica sobre o eixo das ordenadas. Figuras 32 a 37 mostram os resultados para a métrica HV.

DTLZ1 (Figuras 20, 26 & 32): Comparando-se apenas a mediana dos algoritmos, pode-se concluir que o comportamento de convergência do MaOPSO é melhor que todos os outros algoritmos para 2, 3 e 5 objetivos. Para 7 objetivos, o MaOPSO é melhor em convergência que o SMPSO, e o NSGA-III é equivalente aos algoritmos MDFA, CEGA e CDAS-SMPSO. Para 10 objetivos, MaOPSO é superior em convergência ao NSGAIII, SMPSO e CDAS-SMPSO, e equivalente aos algoritmos MDFA e CEGA. Por fim, para efeitos de esclarecimentos, é necessário explicar a razão da alta variância nos resultados do MDFA. Isso se explica porque em algumas execuções ele atingia um GD muito baixo, quase próximo de zero.

Para a métrica IGD, pode-se chegar as seguintes observações sobre os algoritmos levando-se em conta apenas o valor da mediana. O MaOPSO foi melhor que todos os outros algoritmos para 2, 3, 5 e 10 objetivos. É importante notar, porém, que à medida que o número de objetivos aumenta, a diferença de desempenho se torna cada vez menor. Para 7 objetivos, o MaOPSO é melhor que todos os outros algoritmos com exceção do NSGA-III. Para essa instância do problema, o NSGA-III é ligeiramente superior, de maneira que se pode dizer que os algoritmos apresentam desempenhos equivalentes.

Considerando o HV, pode-se chegar às seguintes alegações considerando a mediana dos algoritmos. Os algoritmos MaOPSO, NSGA-III, SMPSO e MDFA apresentaram resultados equivalentes em todas as instâncias dos problemas. E estes algoritmos, por sua vez, foram todos superiores ao CEGA em todas as instâncias. O CDAS-SMPSO apresenta um desempenho inferior ao MaOPSO para 2 e 5 objetivos, mas apresenta um desempenho equivalente a partir de 5 objetivos.

Assim, o MaOPSO apresentou, considerando a convergência e a diversidade em conjunto, um desempenho no mínimo equivalente aos outros algoritmos considerando todas as instâncias do problema.

DTLZ2 (Figuras 21, 27 & 33): Neste problema, pode-se afirmar as seguintes observações considerando a mediana do GD. O MaOPSO e o NSGA-III obtiveram desempenho praticamente equivalentes em todas as instâncias do problema, e o desempenho do SMPSO foi equivalente a eles para 2 e 3 objetivos e foi ligeiramente inferior para mais do que 5 objetivos. Todos esses algoritmos, por sua vez, foram inferiores aos algoritmos MDFA, CEGA e CDAS-SMPSO. Esse fato, se analisado isoladamente, pode indicar que esses algoritmos são melhores que o MaOPSO. No entanto, a análise do IGD e do HV indicaram o contrário.

A métrica IGD revela resultados mais significativos a respeito da superioridade do MaOPSO frente aos outros algoritmos e sobretudo sobre seu principal concorrente: o NSGA-III. Considerando apenas o valor da mediana desta métrica, pode-se fazer as seguintes alegações. O MaOPSO é superior a todos os outros algoritmos para virtualmente

todas as instâncias do problema. A única exceção a essa afirmação é a instância com 7 objetivos. Neste caso, MaOPSO e SMPSO apresentaram resultados equivalentes. Desta forma, esta métrica confirma a superioridade do MaOPSO neste problema relativamente simples mesmo em comparação ao NSGA-III.

Finalmente, a análise do HV, que é uma métrica que condensa informações de convergência e diversidade conjuntamente, confirma as alegações anteriores a respeito do superioridade do MaOPSO com respeito aos outros algoritmos neste problema. Como se pode notar, observado-se o comportamento da mediana desta métrica neste problema, o MaOPSO foi superior a todos os outros algoritmos sobretudo nos cenários com muitos objetivos.

Desta forma, o MaOPSO apresentou, considerando a convergência e a diversidade em conjunto, um desempenho superior aos outros algoritmos principalmente em cenários com muitos objetivos, demonstrando inclusive a sua superioridade frente a seu principal concorrente neste problema.

DTLZ3 (Figuras. 22, 28 & 34): Este problema traz a convergência como seu principal desafio. Com isso, pode-se colocar as seguintes observações a respeito da mediana do GD. O MaOPSO é sempre superior ao CDAS-SMPSO e ao CEGA para qualquer número de objetivos. O MaOPSO apresenta um desempenho equivalente ou ligeiramente superior ao NSGA-III em termos de convergência. Contudo, ele tem uma vantagem sobre este algoritmo que consiste na consistência dos seus resultados de convergência ao contrário do NSGA-III que apresenta uma grande variabilidade, o que indica que em algumas de suas execuções ele não teve uma convergência tão boa. Outro resultado interessante está relacionado com o SMPSO, cuja convergência foi se degradando à medida que o número de objetivos foi aumentando. Este resultado está de acordo com a literatura de muitos objetivos uma vez que o SMPSO é um algoritmo baseado em dominância de Pareto e desta forma apresenta problemas de convergência em cenários com muitos objetivos por razões já explicadas. Finalmente, o MDFA, em virtude de sua diversidade precária, por convergir para poucos pontos da Frente de Pareto, apresenta a melhor convergência entre todos os algoritmos para todas as instâncias deste problema.

A análise da mediana da métrica IGD neste problema (DTLZ3) sugere de forma clara a eficácia do MaOPSO em conciliar convergência e diversidade em problemas com muitos objetivos frente a todos os outros algoritmos. A mediana do IGD é muito significativa sobretudo neste problema cujo principal mérito é avaliar a capacidade de convergência dos algoritmos. Tendo isso em vista, fica claro que o MaOPSO é superior a todos os algoritmos em todas as instâncias. Vale destacar que ele é superior ao seu principal concorrente neste problema: o NSGA-III. Mais uma vez aqui, importa evidenciar a inferioridade do SMPSO frente a todos os algoritmos em cenários com muitos objetivos.

Finalmente, a análise do HV também confirma a boa capacidade de unir convergência e diversidade do MaOPSO além de sua grande estabilidade. Apesar de não se poder declarar nada de modo seguro a respeito da superioridade do MaOPSO em relação a todos os algoritmos, pode-se afirmar que ele junto com o NSGA-III foram os algoritmos que obtiveram os resultados mais consistentes uma vez que apresentaram uma variabilidade pequena em virtualmente todas as instâncias do problema.

Com isso, o MaOPSO alcançou o melhor compromisso entre convergência e diversidade entre todos os algoritmos para este problema que é um dos mais difíceis em termos de convergência da família DTLZ, tendo inclusive alta consistência nos resultados das métricas avaliadas.

DTLZ4 (Figuras 23, 29 & 35): O principal quesito que este problema busca explorar é a diversidade. O ponto mais importante a se notar com respeito a essa métrica neste problema é a degradação da convergência (mesma que pequena) do SMPSO. Neste problema, o MaOPSO apresentou certa superioridade de convergência com relação ao NSGA-III em cenários com muitos objetivos.

Considerando a mediana da métrica IGD, pode-se observar que o MaOPSO teve um desempenho no mínimo equivalente aos outros algoritmos.

Finalmente, a medida do HV evidencia a superioridade do MaOPSO em conciliar convergência e diversidade frente a todos os algoritmos em todas as instâncias do problema.

DTLZ5 (Figuras 24, 30 & 36): Este problema impõe uma dificuldade interessante aos algoritmos, pois se trata de uma Frente de Pareto unidimensional em um espaço com m dimensões. Mesmo nessas condições, fica clara, pela análise da mediana do GD, a boa convergência do MaOPSO em todas as instâncias do problema. É necessário ressaltar que a convergência do MaOPSO é similar ao SMPSO e ao NSGA-III para 2 e 3 objetivos. Mais ainda, é necessário mencionar a superioridade do MaOPSO com respeito ao SMPSO e ao NSGA-III (seu principal competidor) em cenários com muitos objetivos. Por fim, é importante destacar que a superioridade dos algoritmos MDFA, CEGA e CDAS-SMPSO com relação ao MaOPSO vem ao custo da diversidade como será apresentado a seguir.

A análise da mediana do IGD oferece um indicativo sobre a diversidade neste problema. Neste sentido, cumpre notar que os algoritmos MDFA, CEGA e CDAS-SMPSO apresentam um IGD inferior ao do MaOPSO em todas as instâncias do problema. Outro ponto a destacar é que o MaOPSO é superior ou tem um IGD comparável ao NSGA-III em cenários com muitos objetivos.

Por fim, a análise da mediana da métrica HV pode fornecer uma ideia do compromisso entre convergência e diversidade obtido pelos algoritmos. Nesse sentido, os algoritmos MDFA, CEGA e CDAS-SMPSO apresentaram um desempenho inferior com relação ao MaOPSO em todas as instâncias do problema. Contudo, o MaOPSO obteve

um desempenho inferior ou comparável ao algoritmos NSGA-III e SMPSO em todas as instâncias do problema.

Desta forma, apesar da boa convergência, o MaOPSO apresenta um diversidade inferior se comparado ao NSGA-III seu principal concorrente; muito embora, o MaOPSO apresente uma diversidade muito melhor que os algoritmos MDFA, CEGA e CDAS-SMPSO.

DTLZ6 (Figuras 25, 31 & 37): Este problema, além de ser um problema degenerado como o anterior, apresenta a dificuldade de convergência (enorme quantidade de Frente de Pareto locais) como um dos seus aspectos. Sendo assim, pode-se mencionar as seguintes alegações sobre os algoritmos a partir dos resultados obtidos. A primeira e mais importante é a superioridade de convergência do MaOPSO sobre seu principal corrente, o NSGA-III. O segundo ponto a se mencionar é a equivalência de convergência do MaOPSO com respeito ao SMPSO para dois e três objetivo. Além disso, o MaOPSO supera o SMPSO em cenários com muitos objetivos. De fato, a habilidade de convergência do SMPSO se deteriora com o aumento do número de objetivos.

A análise da mediana do IGD permite fazer as seguintes observações. O MaOPSO é superior aos algoritmos NSGA-III, MDFA, CEGA e CDAS-SMPSO em todas as instâncias do problema com exceção da instância com três objetivos. Nesse caso, o MaOPSO é apenas superior aos algoritmos MDFA, CEGA e CDAS-SMPSO e superado pelos algoritmos NSGA-III e SMPSO. Em todos os casos, o MaOPSO obteve um desempenho no mínimo equivalente ao SMPSO.

A mediana do HV permite a seguinte análise. O MaOPSO foi superior a seu principal concorrente (NSGA-III) em todas as instâncias com exceção do caso com três objetivos. Com três objetivos, o NSGA-III é superior ao MaOPSO. Em todas as instâncias, o MaOPSO é superior ao CEGA e ao CDAS-SMPSO. Com respeito ao MDFA, o MaOPSO é inferior e equivalente a este algoritmo nas instâncias com três e cinco objetivos, respectivamente. Por fim, o MaOPSO é inferior ao SMPSO em todas as instâncias do problema. Com isso, embora o MaOPSO tenha obtido um bom compromisso, de acordo com as métricas acima, em termos de convergência e diversidade comparado ao NSGA-III, MDFA, CEGA e CDAS-SMPSO na maioria das instâncias do problema, ele não foi melhor que o SMPSO neste problema. Este fato pode estar relacionado com a natureza deste problema que apresenta uma Frente de Pareto degenerada. Isso faz com que o MaOPSO tenha dificuldades em obter as soluções extremas e com isso não consiga construir um hiperplano que comporte toda a Frente de Pareto.

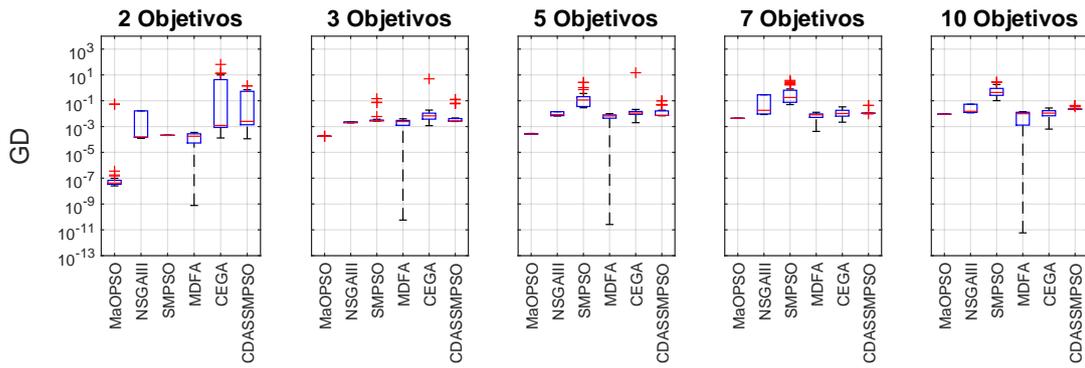


Figura 20 – Resultados da métrica GD para o problema DTLZ1 (depois de 100 000 iterações).

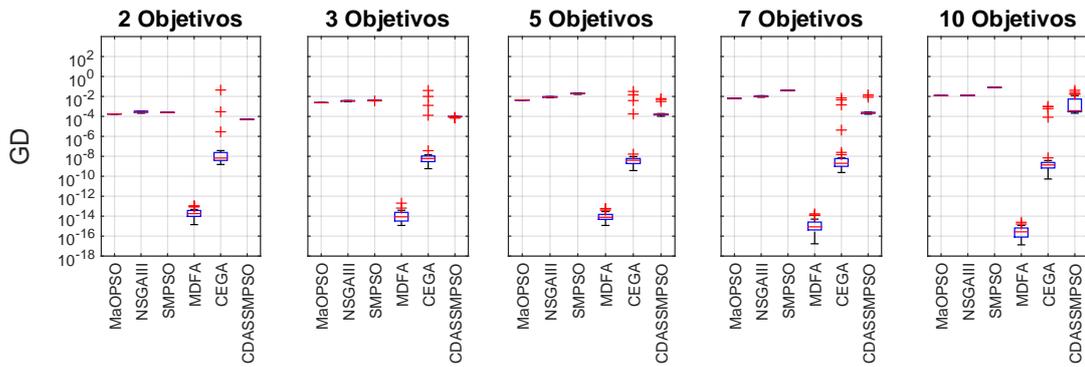


Figura 21 – Resultados da métrica GD para o problema DTLZ2 (depois de 100 000 iterações).

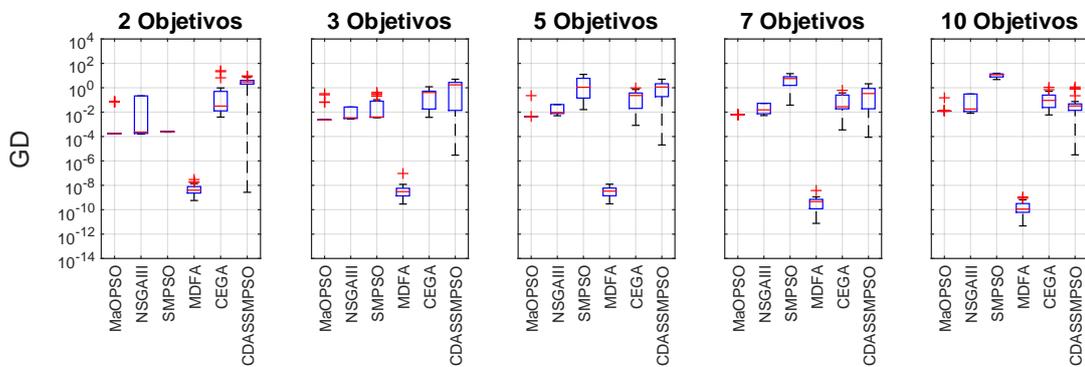


Figura 22 – Resultados da métrica GD para o problema DTLZ3 (depois de 100 000 iterações).

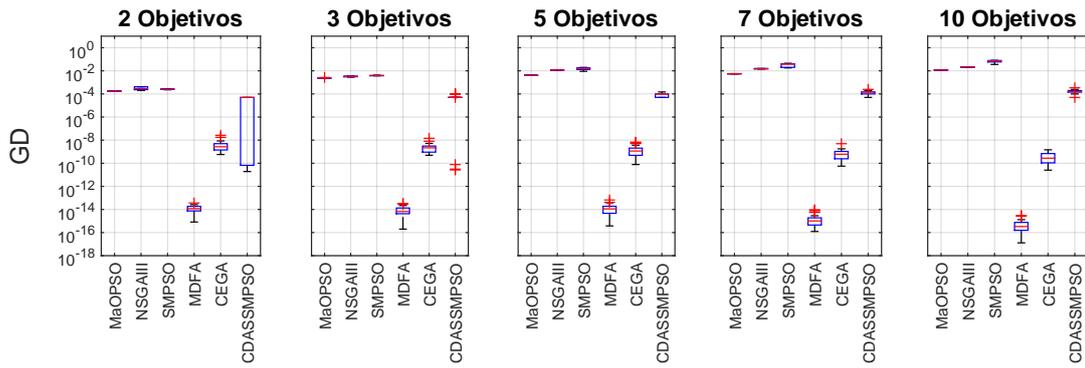


Figura 23 – Resultados da métrica GD para o problema DTLZ4 (depois de 100 000 iterações).

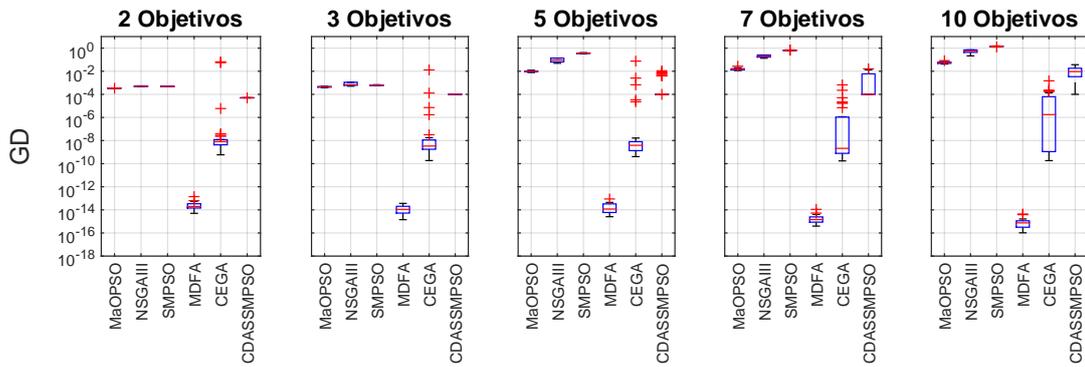


Figura 24 – Resultados da métrica GD para o problema DTLZ5 (depois de 100 000 iterações).

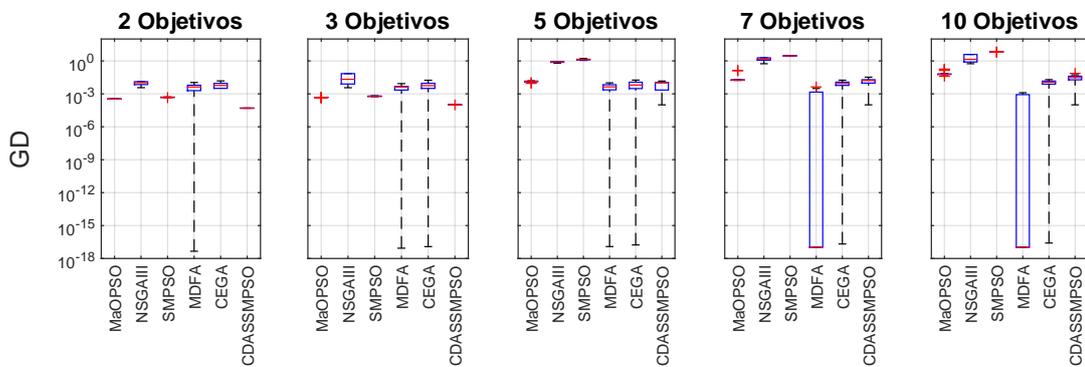


Figura 25 – Resultados da métrica GD para o problema DTLZ6 (depois de 100 000 iterações).

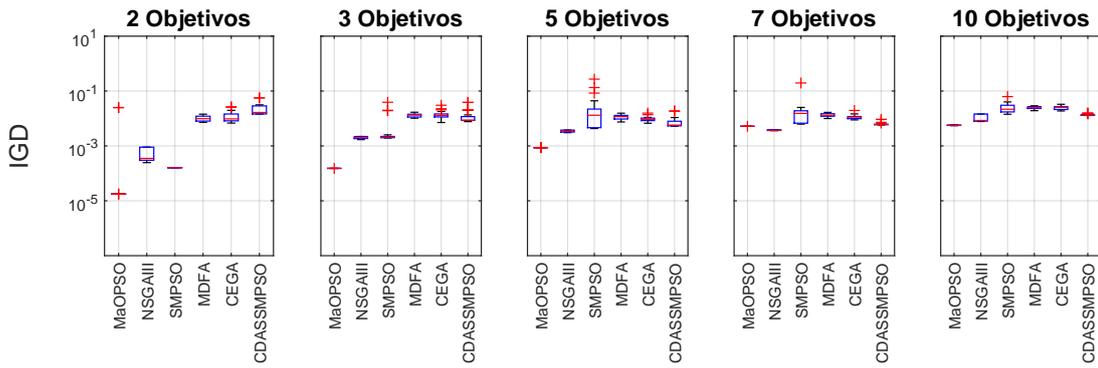


Figura 26 – Resultados da métrica IGD para o problema DTLZ1 (depois de 100 000 iterações).

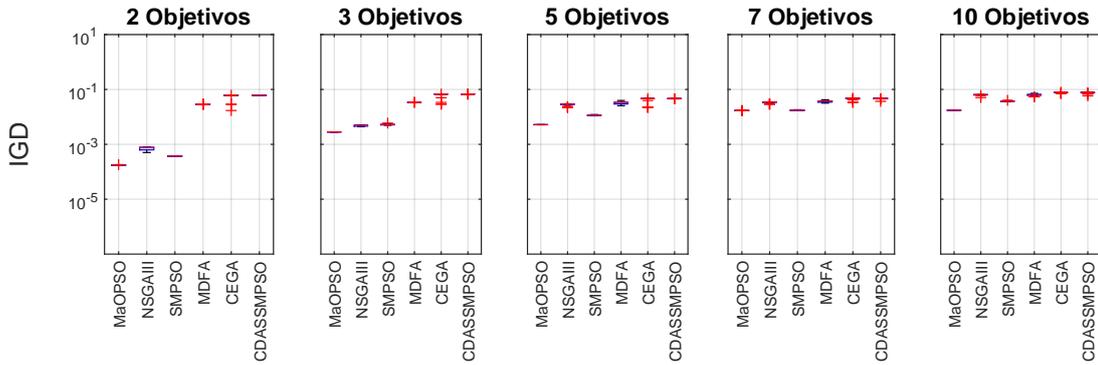


Figura 27 – Resultados da métrica IGD para o problema DTLZ2 (depois de 100 000 iterações).

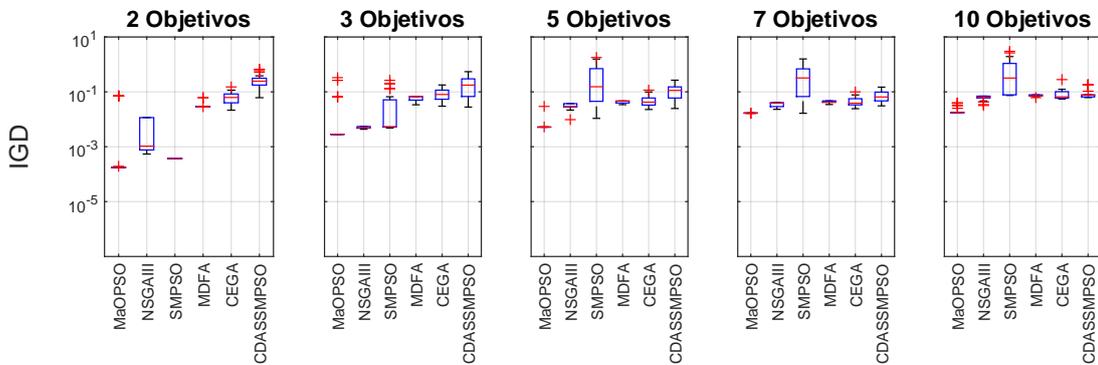


Figura 28 – Resultados da métrica IGD para o problema DTLZ3 (depois de 100 000 iterações).

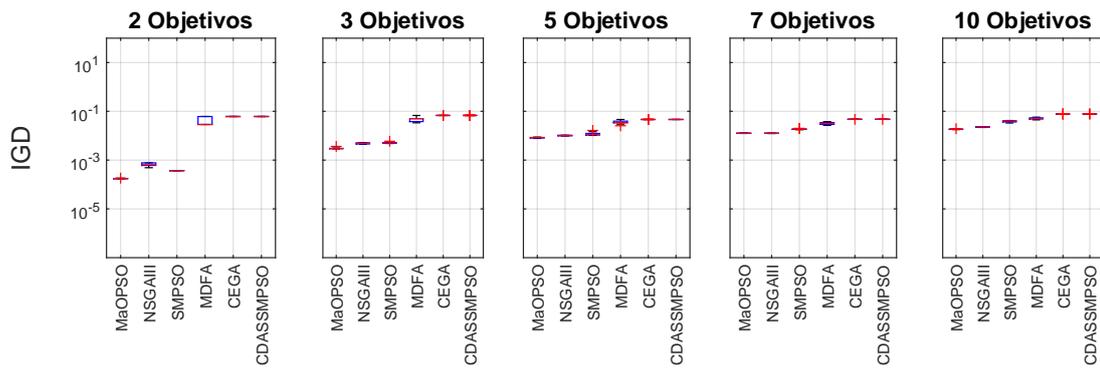


Figura 29 – Resultados da métrica IGD para o problema DTLZ4 (depois de 100 000 iterações).

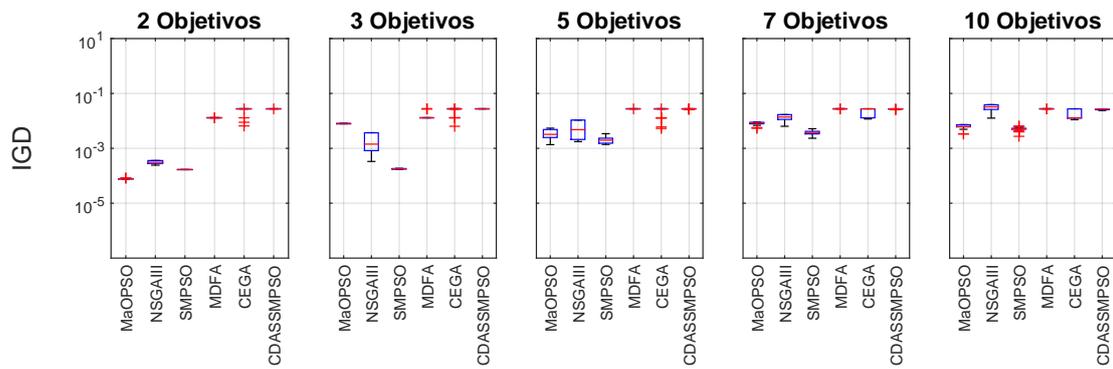


Figura 30 – Resultados da métrica IGD para o problema DTLZ5 (depois de 100 000 iterações).

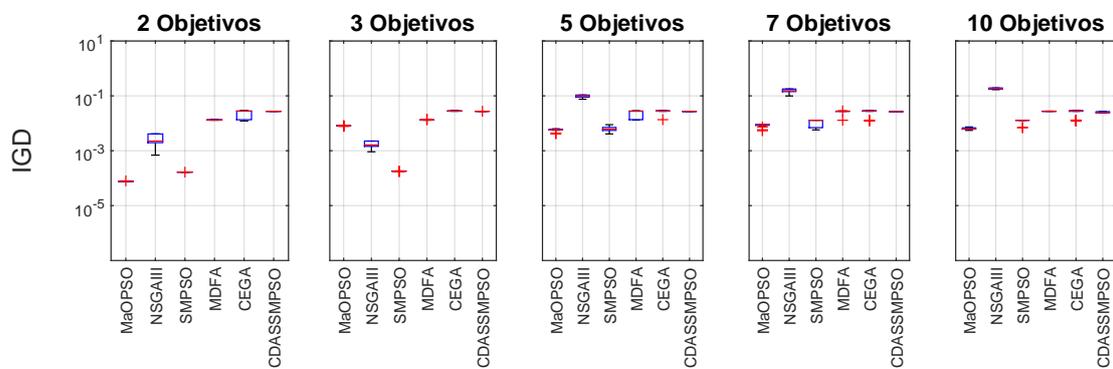


Figura 31 – Resultados da métrica IGD para o problema DTLZ6 (depois de 100 000 iterações).

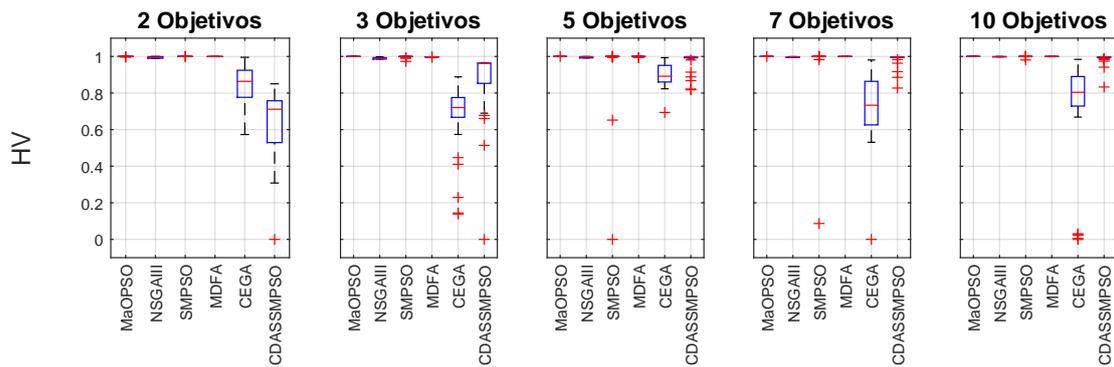


Figura 32 – Resultados da métrica HV para o problema DTLZ1 (depois de 100 000 iterações).

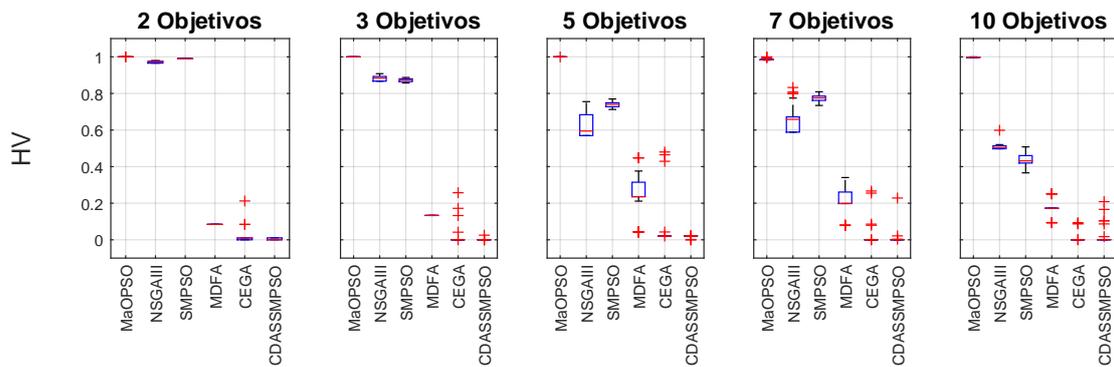


Figura 33 – Resultados da métrica HV para o problema DTLZ2 (depois de 100 000 iterações).

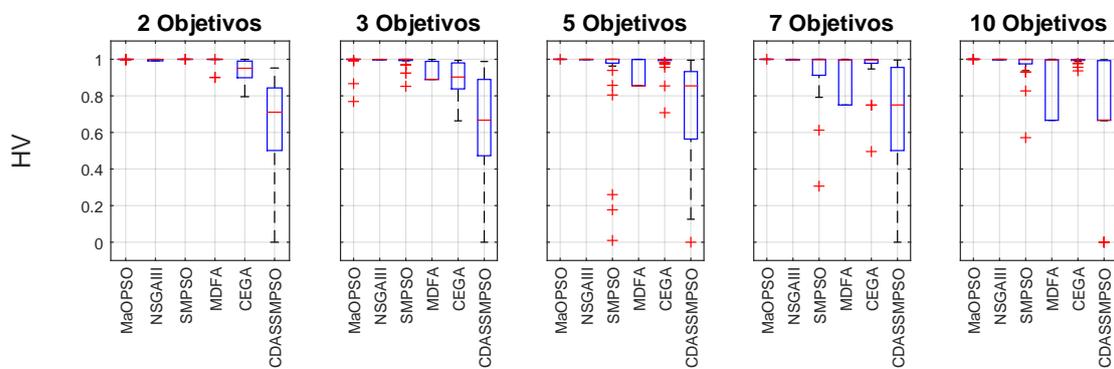


Figura 34 – Resultados da métrica HV para o problema DTLZ3 (depois de 100 000 iterações).

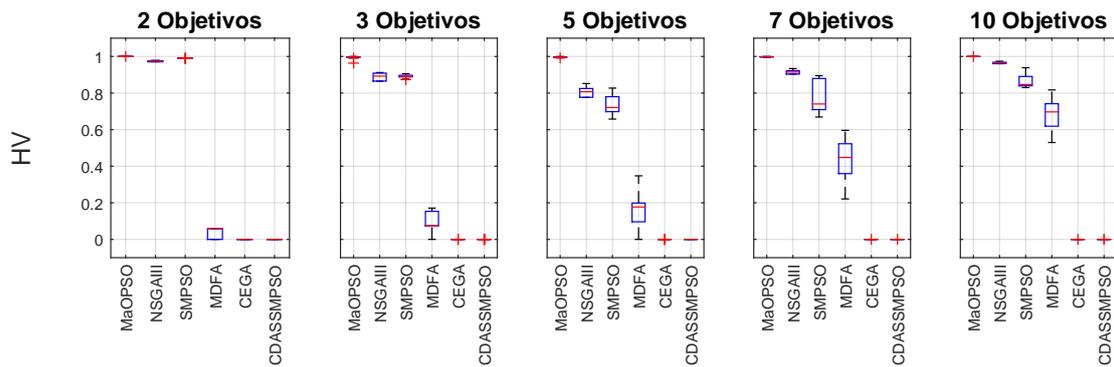


Figura 35 – Resultados da métrica HV para o problema DTLZ4 (depois de 100 000 iterações).

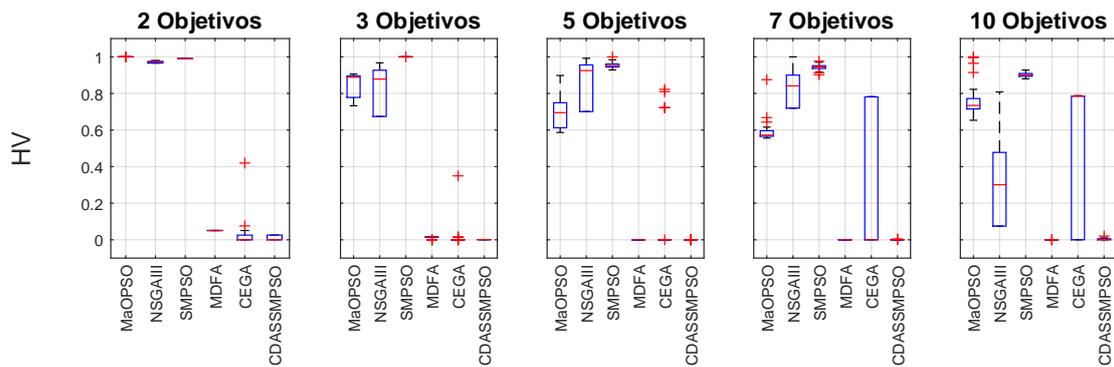


Figura 36 – Resultados da métrica HV para o problema DTLZ5 (depois de 100 000 iterações).

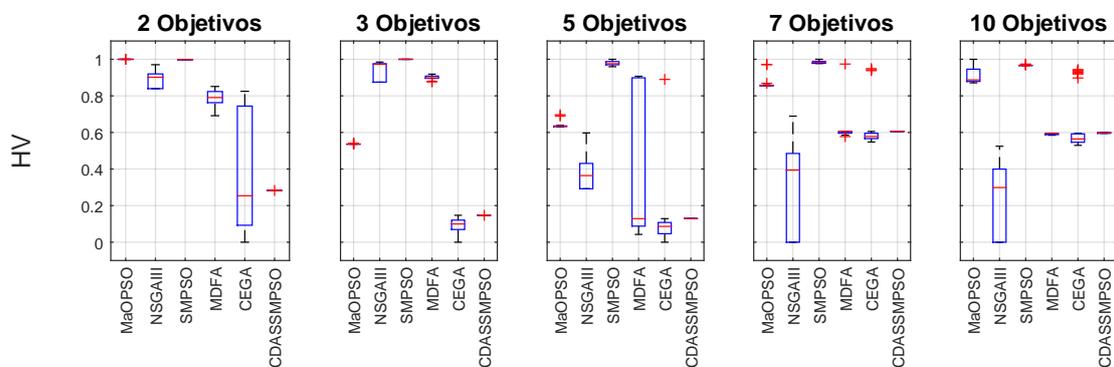


Figura 37 – Resultados da métrica HV para o problema DTLZ6 (depois de 100 000 iterações).

4.2.3 Resultados para GD, IGD, e Hipervolume para 1 000 iterações

Esta seção têm o seguinte objetivo: mostrar como o MaOPSO se comporta usando um número limitado e reduzido de iterações com relação aos outros algoritmos. Sendo assim, os algoritmos são comparados em termos de GD, IGD e HV usando apenas 1 000 iterações ao invés de 100 000 iterações como na subseção 4.2.2 enquanto todas as outras configurações do arranjo experimental anterior foram mantidas. Além disso, é importante enfatizar que neste experimento, o CEGA foi executado também com 1 000 iterações, diferentemente de como aconteceu na subseção 4.2.2. Assim, todos os algoritmos usaram a mesma quantidade de iterações entretanto o CEGA continua usando menos avaliações de aptidão pois sua população tem que se manter pequena em virtude do seu alto custo computacional. Ou seja, o CEGA usou 10^5 avaliações de aptidão, enquanto os outros algoritmos usaram 1×10^5 avaliações de aptidão para 2, 3 e 5 objetivos, 2×10^5 avaliações para sete objetivos e 3×10^5 avaliações de aptidão para 10 objetivos. Apesar de não ser o ideal, esse novo arranjo experimental permite uma melhor avaliação dos algoritmos em condições um pouco mais justas. Além disso, esse arranjo experimental permite avaliar o desempenho dos algoritmos usando um número reduzido de avaliações de aptidão em comparação com o arranjo experimental da subseção 4.2.2. As Figuras 38 a 43 mostram o comportamento de convergência dos algoritmos utilizando a métrica GD, e Figuras 44 a 49 mostram os resultados relativos a métrica IGD. Todos os resultados para GD e IGD são apresentados como *box-plots* com escala logarítmica sobre o eixo das ordenadas. Figuras 50 a 55 mostram os resultados para a métrica HV também usando *box-plots*.

DTLZ1 (Figuras 38, 44 & 50): Como se pode perceber por essas figuras, o SMPSO teve o pior desempenho em todas as instâncias. Esse comportamento já era esperado devido ao fato dele ser baseado em dominância de Pareto. MaOPSO foi o algoritmo com melhor convergência para dois objetivos, contudo ele foi superado pelo CEGA e o MDFA nas outras instâncias. De fato, MDFA e CEGA foram os algoritmos que apresentaram melhores valores de convergência. O IGD releva resultados mais interessantes. Considerando essa métrica, MaOPSO e NSGA-III são os melhores algoritmos considerando todas as instâncias do problema. O SMPSO apresenta um alto IGD para as instâncias com sete e dez objetivos pois ele não consegue convergir nessas instâncias do problema. O algoritmo CDAS-SMPSO é o algoritmo que apresentou o pior valor de IGD para as instâncias com dois, três e cinco objetivos, e teve o segundo pior IGD para as instâncias com sete e dez objetivos. É importante mencionar que CEGA e o MDFA apresentam um IGD pior que o MaOPSO e o NSGA-III. Ou seja, apesar do CEGA e MDFA apresentarem a melhor convergência que os outros algoritmos para a maioria das instâncias deste problema, eles apresentaram uma pior diversidade segundo a métrica IGD em todas as instâncias do problema. Considerando o HV, os algoritmos MaOPSO, MDFA, SMPSO, MDFA apresentam o hipervolume máximo relativamente aos outros algoritmos, indicando que esses algoritmos provavelmente tem o

melhor compromisso entre convergência e diversidade.

DTLZ2 (Figuras. 39, 45 & 51): Neste problema, pode-se afirmar os seguintes pontos considerando a mediana do GD. Os algoritmos CEGA e MDFA apresentaram os melhores valores de convergência. O MaOPSO apresentou um valor ligeiramente maior de convergência que o algoritmo NSGA-III para instâncias com até sete objetivos. O NSGA-III superou ligeiramente o MaOPSO em termos de convergência na instância com dez objetivos. O SMPSO apresentou o pior valor em termos de convergência em instâncias com muitos objetivos. A métrica IGD é mais esclarecedora a respeito do comportamento dos algoritmos. De acordo com essa métrica, os algoritmos CEGA, MDFA, CDAS-SMPSO são algoritmos que apresentam a pior diversidade em todas as instâncias do problema. O algoritmo SMPSO como esperado apresenta uma boa diversidade para as instâncias com dois e três objetivos, mas sua diversidade é deteriorada à medida que o número de objetivos aumenta. Contudo, o ponto que merece destaque considerando o IGD, é que o MaOPSO foi superior a todos os algoritmos em todas as instâncias do problema, superando inclusive o seu principal concorrente que o NSGA-III. Outro ponto que também surpreende e que é importante frisar é a degradação da diversidade do NSHA-III com o aumento do número de objetivos, fenômeno que não acontece com a proposta desta tese, o MaOPSO. Finalmente, para encerrar a análise é necessário averiguar o comportamento da métrica HV, afim de ser verificar quais algoritmos apresentam uma maior capacidade de equilibrar de modo adequado convergência e diversidade. Da análise do HV, pode-se verificar que ele revela um resultado similar ao do IGD, mostrando a superioridade do MaOPSO frente aos algoritmos mesmo usando um número limitado de iterações como é o caso deste estudo que usa somente 1,000 iterações. Desta forma, o MaOPSO apresentou, considerando a convergência e a diversidade em conjunto, um desempenho superior aos outros algoritmos mesmo usando um número reduzido de iterações principalmente em cenários com muitos objetivos, demonstrando inclusive a sua superioridade frente a seu principal concorrente neste problema, o NSGA-III.

DTLZ3 (Figuras. 40, 46 & 52): A análise da convergência mostra que o MaOPSO não apresentou bons valores de convergência. De fato, o MaOPSO e o SMPSO foram os algoritmos com pior valor de convergência. Novamente, o CEGA e o MDFA foram os algoritmos que apresentaram os melhores valores de convergência. Os algoritmos NESGA-III e CDAS-SMPSO também não exibiram uma boa convergência, mas apresentaram valores melhores que o MaOPSO e o SMPSO. A incapacidade do MaOPSO e do NSGA-III atingirem a convergência neste problema pode ser explicada pela grande quantidade de Frente de Pareto locais que esse problema possui, sendo que apenas 1 000 iterações é um número de iterações insuficientes para garantir a convergência dessas algoritmos neste problema desafiador. Os únicos algoritmos que atingiram a convergência foram os algoritmos CEGA e MDFA.

O fato dos algoritmos NSGA-III, MaOPSO, SMPSO, e CDAS-SMPSO não atingirem a convergência prejudica a análise do IGD e por conseguinte da diversidade uma vez que o IGD é uma métrica que apenas tem a capacidade de passar alguma informação sobre a diversidade dos algoritmos se esses mesmos algoritmos tiverem alcançado a convergência. Tendo isso em vista, pode-se fazer as seguintes alegações. O NSGA-III foi o algoritmo que apresentou melhor a diversidade apenas nas instâncias com dois e três objetivos. Contudo, para instâncias com cinco ou mais objetivos, os valores de IGD do NSGA-III se degradam rapidamente. Entretanto, não se pode dizer que isso ocorre porque a diversidade do NSGA-III é ruim, mas pelo fato do conjunto aproximação produzidos por este algoritmo estarem longe da Frente de Pareto. Fato este que evidencia o cuidado que se deve ao se analisar o IGD uma vez que é uma métrica que mede convergência e diversidade (cobertura e espalhamento) simultaneamente. Um ponto que merece grande destaque são os altos valores de IGD obtidos pelo SMPSO sobretudo para instâncias com muitos objetivos. De fato, para dez objetivos, os valores de IGD atingem valores com duas casas decimais. É interessante também observar que o NSGA-III apresentou valores inferiores de IGD para as instâncias com sete e dez objetivos. Por outro lado, o NSGA-III foi equivalente ou superior ao MaOPSO em termos de IGD para as instâncias com dois, três, e cinco objetivos. O CDAS-SMPSO apresentou também muito altos de IGD em todas as instâncias, apresentando valores de IGD superiores os valores do MaOPSO em todas as instâncias deste problema.

A análise do HV é bastante reveladora, pois mostra que apenas o MaOPSO e o NSGA-III foram os que alcançaram o maior valor nessa métrica em todas as instâncias. De fato, esses algoritmos foram os únicos que atingiram o valor máximo relativo na instância com dez objetivos. Tirando, essas exceção todos os algoritmos virtualmente atingiram o valor máximo do HV, a única exceção sendo CDAS-SMPSO que apresentou sempre um desempenho inferior.

DTLZ4 (Figuras. 41, 47 e 53): Nesse problema, merece destaque a superioridade de convergência do MaOPSO comparado ao NSGA-III e ao SMPSO, sobretudo em cenários com muitos objetivos. Outro ponto que merece ser mencionado é a degradação da convergência (mesma que pequena) do SMPSO à medida que o número de objetivos aumenta. Novamente, os algoritmos CEGA, MDFA e CDAS-SMPSO apresentaram altos valores de convergência neste problema. Considerando a métrica IGD, pode-se observar que o MaOPSO apresentou um desempenho equivalente ou superior ao NSGA-III em todas as instâncias do problema. E ambos os algoritmos foram superiores aos outros algoritmos para instâncias com três ou mais objetivos. Por outro lado, os algoritmos que se destacaram na convergência, ou seja, CEGA, MDFA, e CDAS-SMPSO. apresentaram valores inferiores de IGD o que mostra que esses algoritmos apresentaram uma diversidade pobre apesar de sua alta capacidade de convergência. Finalmente, o HV evidencia a capacidade do MaOPSO

em conciliar convergência e diversidade. De fato, o MaOPSO, NSGA-III, e SMPSO foram os algoritmos que apresentaram valor máximo de HV relativamente a todos os algoritmos.

DTLZ5 (Figuras 42, 48 e 54): Como já foi mencionado, este problema é um problema degenerado e que portanto impõe uma dificuldade a mais sobre os algoritmos utilizados em sua solução. Neste problema, o MaOPSO é superior em termos de convergência ao NSGA-III em todas as instâncias deste problema. É importante também ressaltar a degradação da convergência do SMPSO à medida que o número de objetivos cresce. O ponto que merece ser comentado é a grande capacidade de convergência do algoritmo CEGA.

A análise do GD apenas diz muito pouco a respeito do comportamento dos algoritmos. Ela apenas informa a respeito da convergência dos algoritmos, mas não esclarece nada a respeito da diversidade (distribuição e cobertura) das soluções. Sendo assim, a métrica IGD sendo uma métrica complementar ao GD uma vez que ela avalia a diversidade das soluções precisa ser avaliada em conjunto com ela. A observação dos *box-plots* do IGD permite emitir as seguintes conclusões. MaOPSO não foi superado por nenhum algoritmo em termos de IGD em todas as instâncias, com exceção da instância com três objetivos. O CEGA, MDFA, CDAS-SMPSO foram sempre os piores algoritmos em todas as instâncias. Esse último resultado precisa ser um pouco mais pois revela que até mesmo com um número limitado de iterações, esses algoritmos convergiram para provavelmente para um ponto ou número pequeno de pontos. Fato que explica os altos valores de convergências associados a valores muito baixos de diversidade (IGD).

Por fim, a última métrica que deve ser analisada é o HV. Essa métrica é muito reveladora, pois mostra claramente a alta capacidade de conciliar convergência e diversidade nesse problema que impõe uma grande dificuldade na geração de diversidade tanto para instâncias com dois ou três objetivos mas sobretudo para instâncias com muitos objetivos mesmo usando um número limitado de avaliações de aptidão. É interessante notar que o NSGA-III foi o algoritmos que produziu is piores valores de hipervolume em instâncias com muitos objetivos.

Desta forma, o MaOPSO foi o único algoritmo que conseguiu conciliar convergência e diversidade neste problema particularmente difícil por se tratar de um problema degenerado com um Frente de Pareto com uma dimensão muito menor do que a dimensão natural caso todos os objetivos fossem de fato conflitantes.

DTLZ6 (Figuras 43, 49 e 55): Este problema, vale repetir, é também um problema degenerado exatamente como o problema DTLZ5, mas com a diferença de que ele apresenta muitas Frontes de Pareto locais, o torna esse problema ideal para se avaliar tanto a capacidade de convergência dos algoritmos como também a capacidade desses algoritmos de gerar diversidade e cobertura em um problema degenerado.

A análise da convergência dos algoritmos neste problema oferece as seguintes

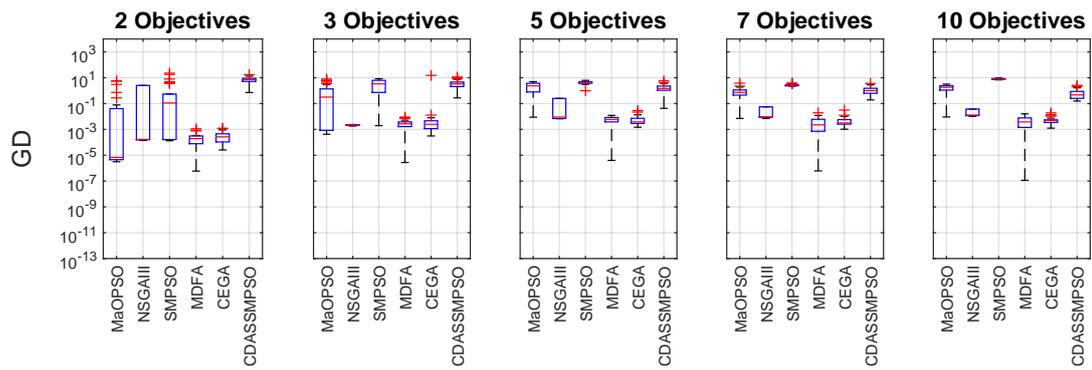


Figura 38 – Resultados da métrica GD para o problema DTLZ1 (depois de 1 000 iterações).

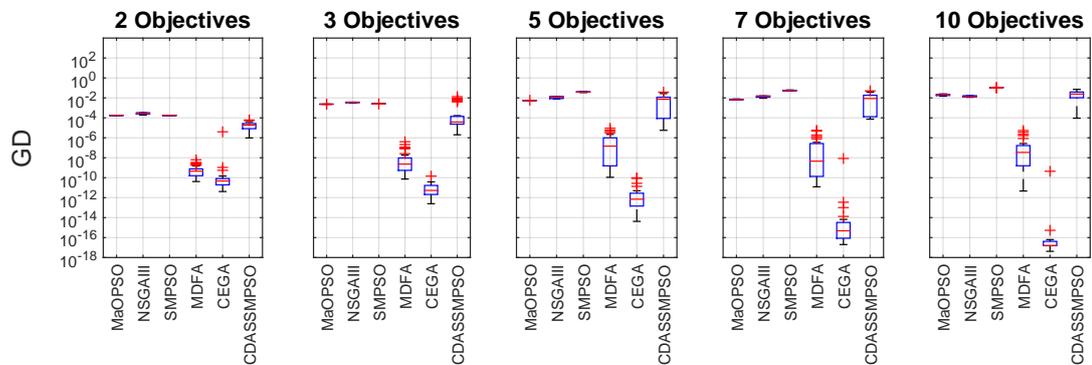


Figura 39 – Resultados da métrica GD para o problema DTLZ2 (depois de 1 000 iterações).

conclusões. O MaOPSO obteve melhores valores de convergência que o NSGA-III. Os algoritmos CEGA, MDFA, e CDAS-SMPSO foram os algoritmos que apresentaram melhor convergência. Por outro lado, a análise do IGD indica uma outra face dos resultados uma vez que considerando essa métrica os algoritmos CEGA, MDFA, CDAS-SMPSO apresentaram valores inferiores aos obtidos pelo NSGA-III. Isto significa que apesar desses algoritmos apresentarem uma melhor convergência que o MaOPSO, eles convergem apenas para alguns pontos da Frente de Pareto. Ainda segundo o IGD, o MaOPSO foi o algoritmo que obteve melhor diversidade entre todos os algoritmos em todas as instâncias do problema com exceção da instância com três objetivos, em que é superado apenas pelo SMPSO.

A métrica HV ratifica a superioridade do MaOPSO frente aos outros algoritmos em cenários com muitos objetivos. Contudo, para a instância com três objetivos, em que o MaOPSO foi superado pelos algoritmos SMPSO e MDFA.

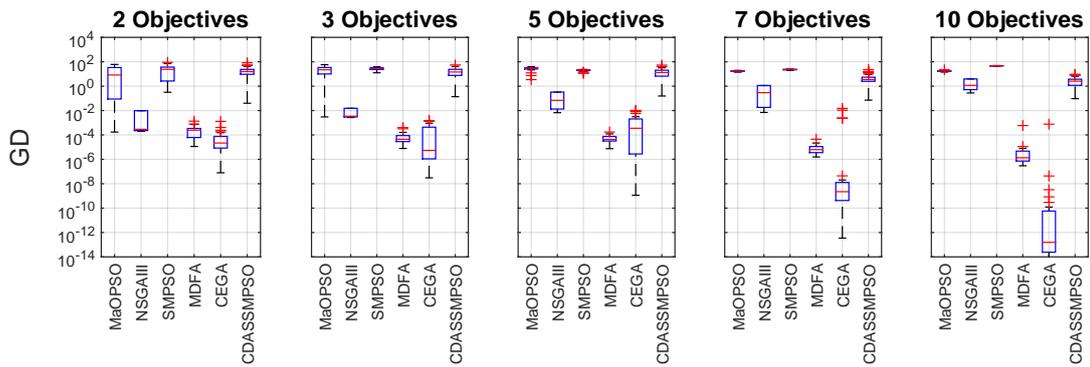


Figura 40 – Resultados da métrica GD para o problema DTLZ3 (depois de 1 000 iterações).

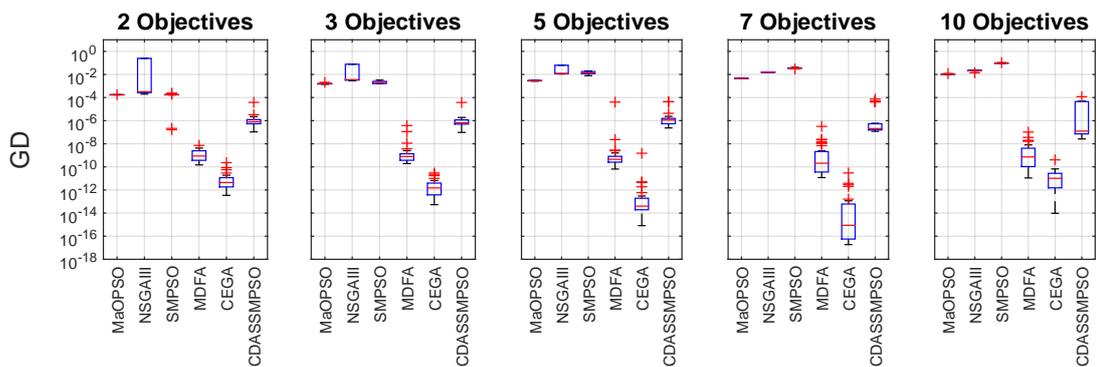


Figura 41 – Resultados da métrica GD para o problema DTLZ4 (depois de 1 000 iterações).

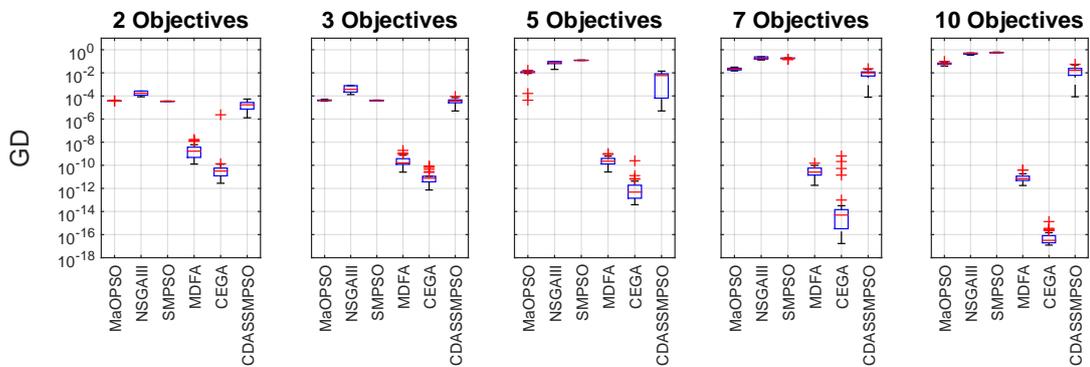


Figura 42 – Resultados da métrica GD para o problema DTLZ5 (depois de 1 000 iterações).

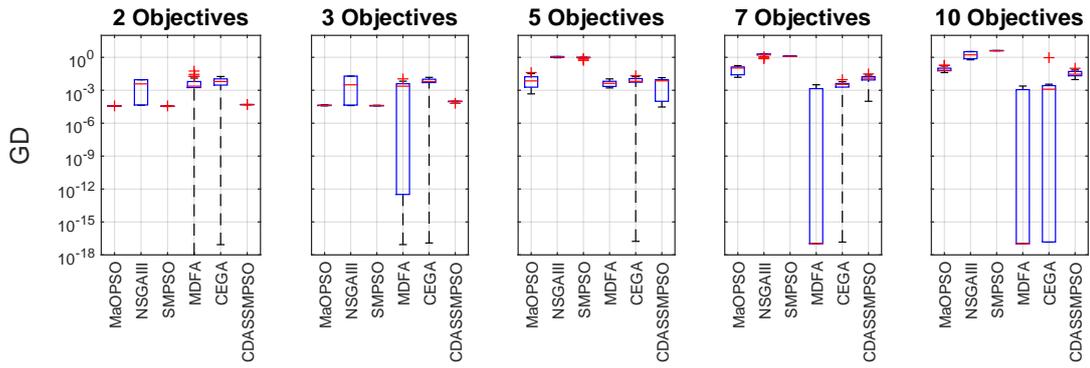


Figura 43 – Resultados da métrica GD para o problema DTLZ6 (depois de 1 000 iterações).

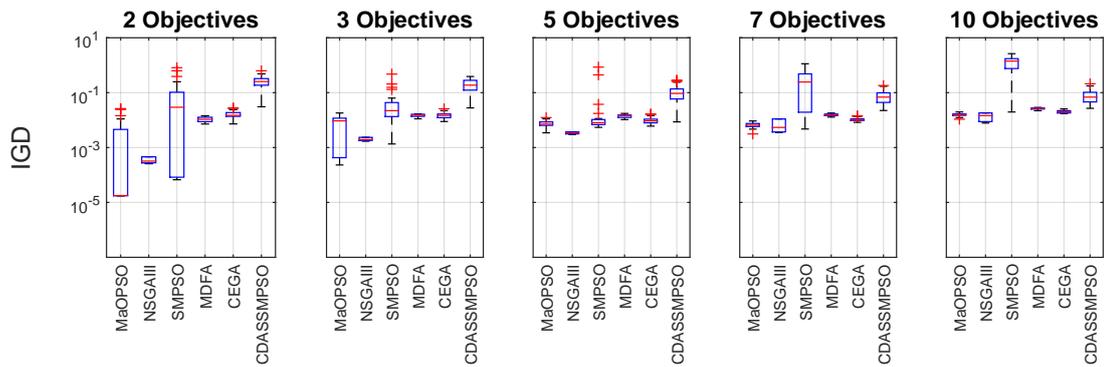


Figura 44 – Resultados da métrica IGD para o problema DTLZ1 (depois de 1 000 iterações).

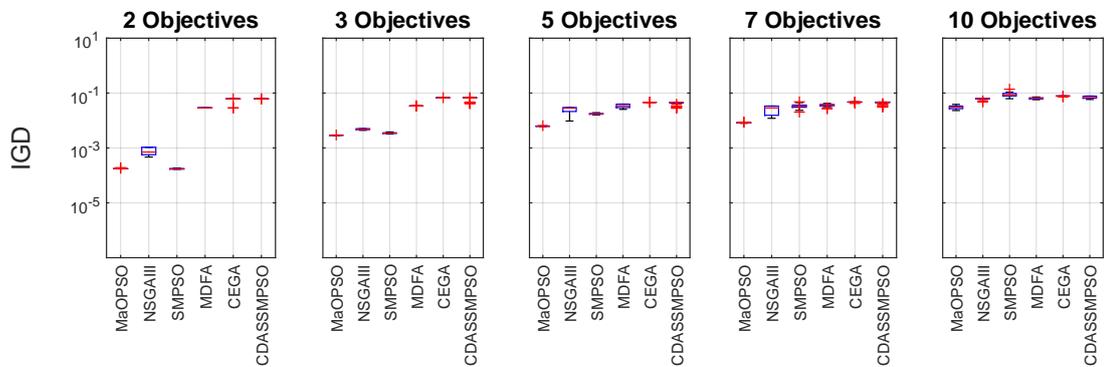


Figura 45 – Resultados da métrica IGD para o problema DTLZ2 (depois de 1 000 iterações).

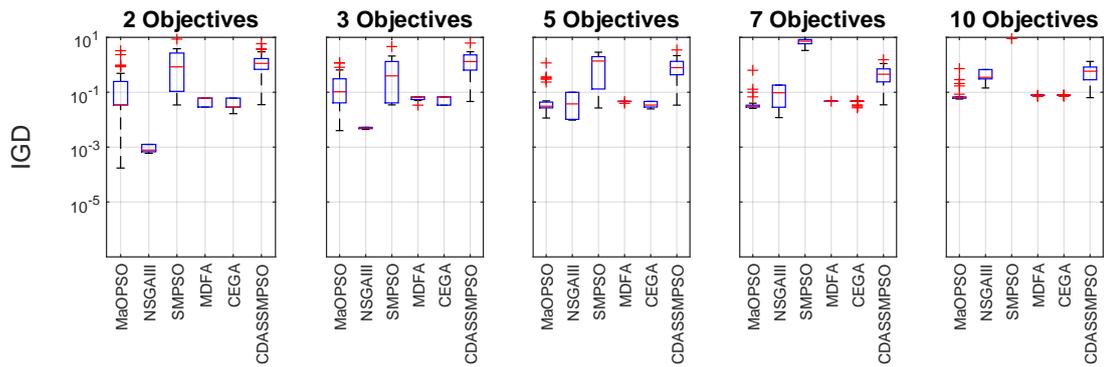


Figura 46 – Resultados da métrica IGD para o problema DTLZ3 (depois de 1 000 iterações).

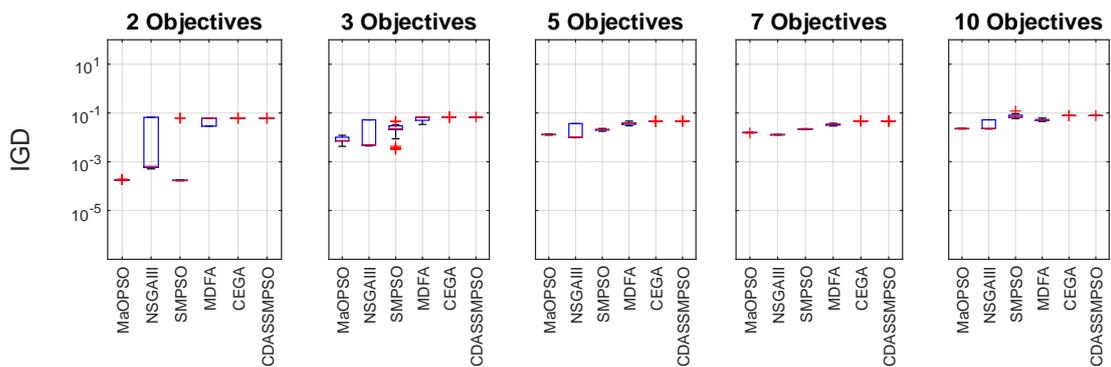


Figura 47 – Resultados da métrica IGD para o problema DTLZ4 (depois de 1 000 iterações).

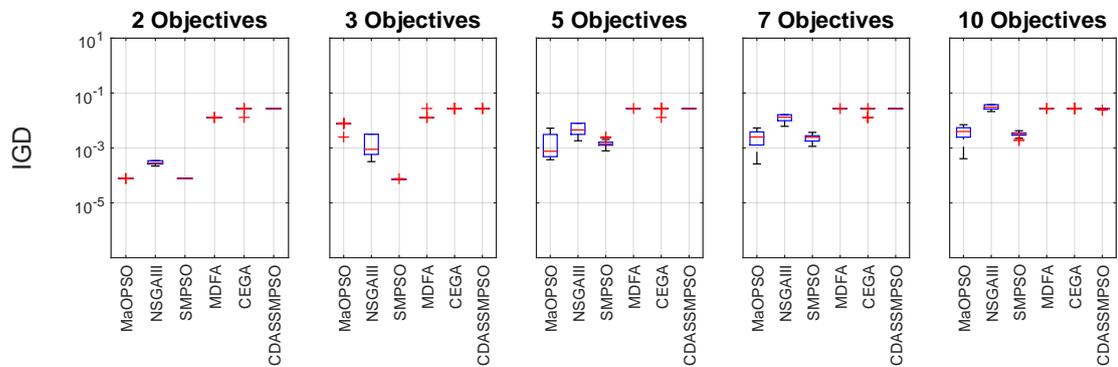


Figura 48 – Resultados da métrica IGD para o problema DTLZ5 (depois de 1 000 iterações).

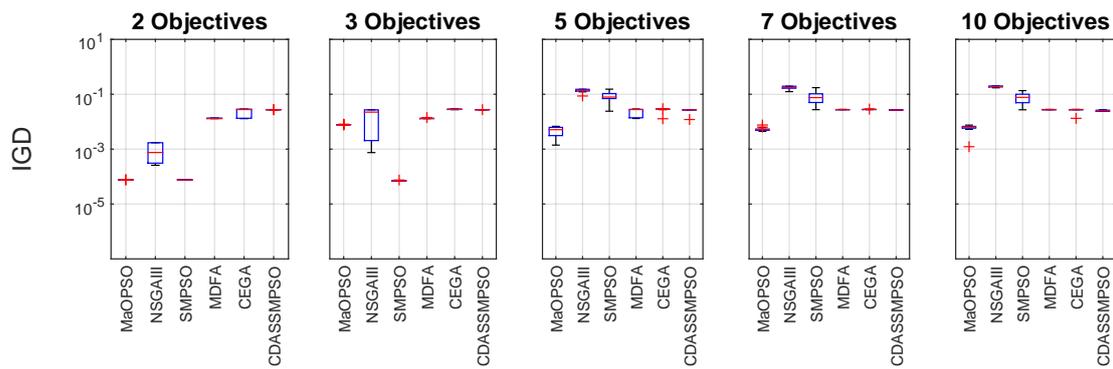


Figura 49 – Resultados da métrica IGD para o problema DTLZ6 (depois de 1 000 iterações).

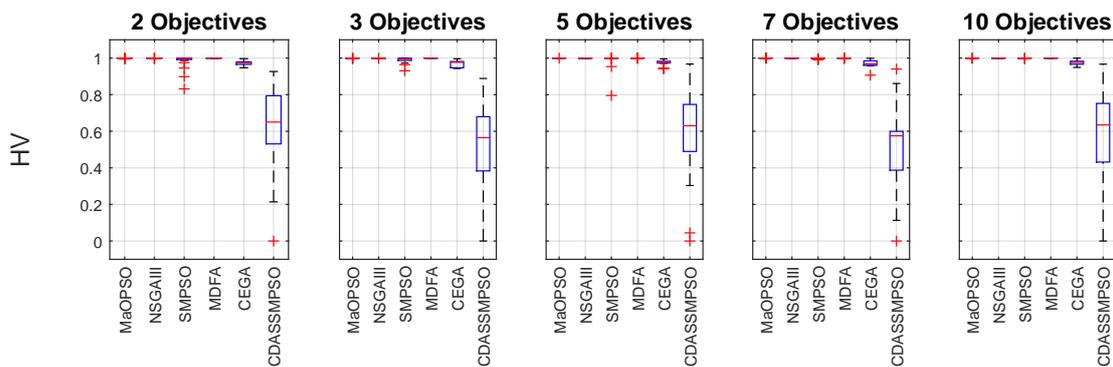


Figura 50 – Resultados da métrica HV para o problema DTLZ1 (depois de 1 000 iterações).

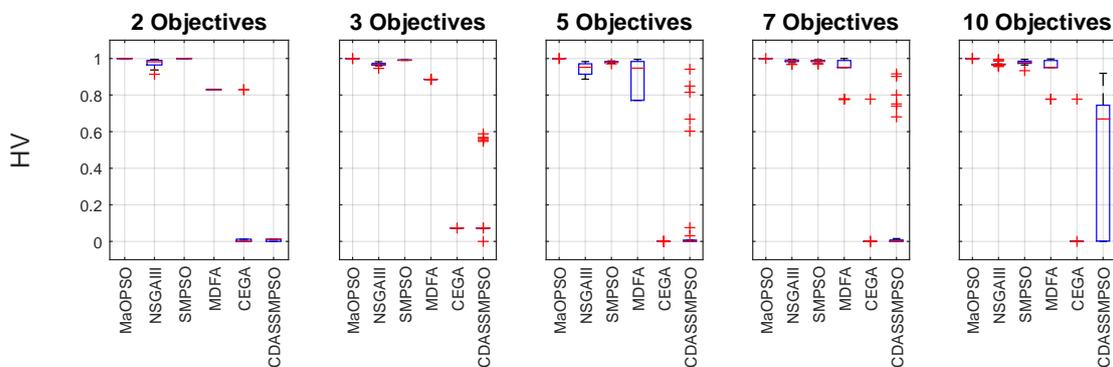


Figura 51 – Resultados da métrica HV para o problema DTLZ2 (depois de 1 000 iterações).

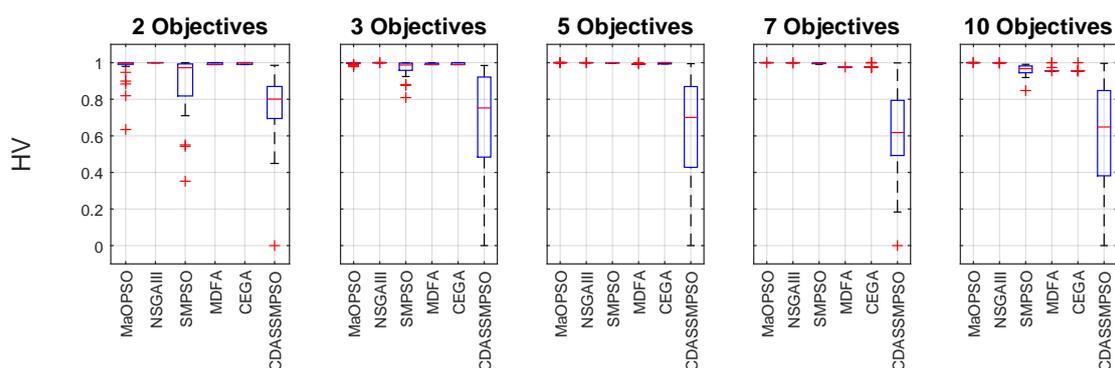


Figura 52 – Resultados da métrica HV para o problema DTLZ3 (depois de 1 000 iterações).

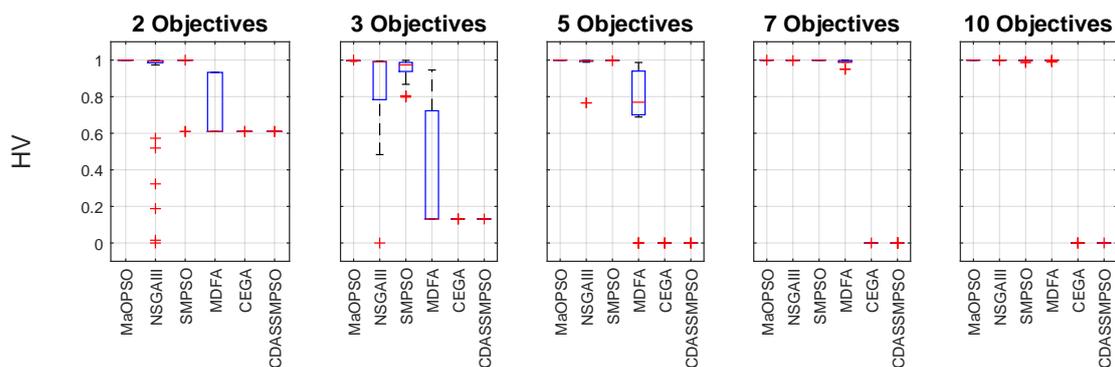


Figura 53 – Resultados da métrica HV para o problema DTLZ4 (depois de 1 000 iterações).

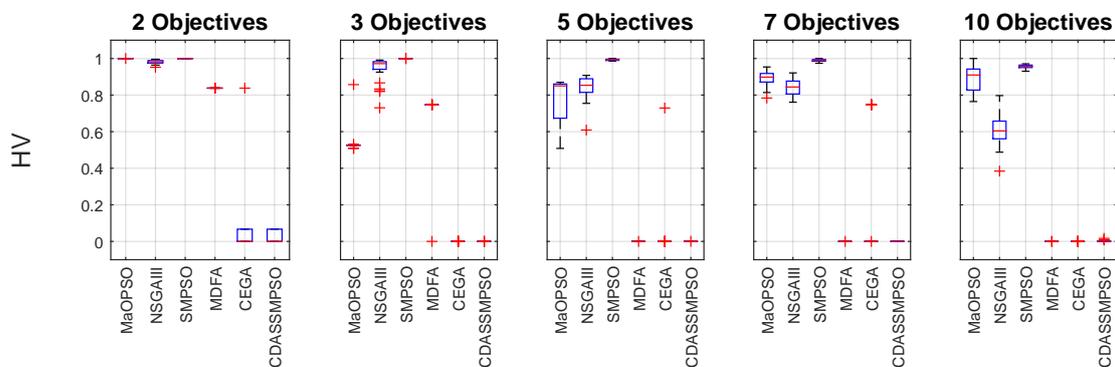


Figura 54 – Resultados da métrica HV para o problema DTLZ5 (depois de 1 000 iterações).

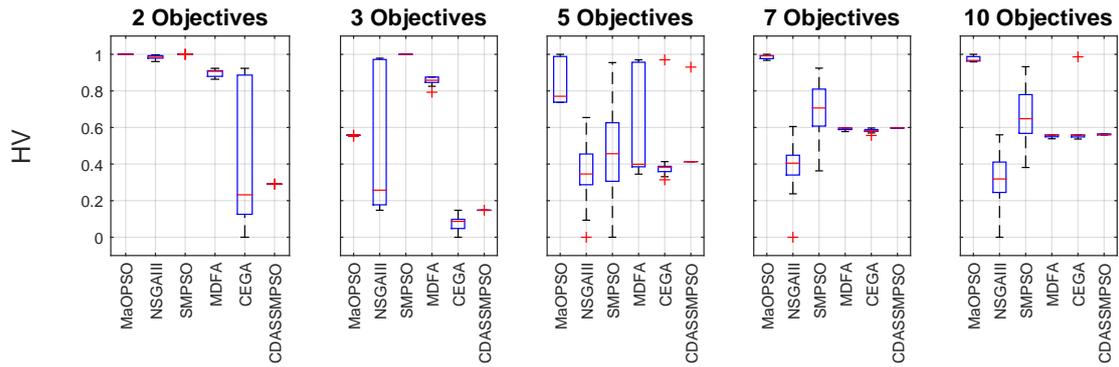


Figura 55 – Resultados da métrica HV para o problema DTLZ6 (depois de 1 000 iterações).

4.2.4 Análise de Significância Estatística Dos Resultados

Nesta seção, os resultados são apresentados de uma maneira compacta usando uma análise estatística. O objetivo desta análise é primeiramente fornecer uma comprovação de que os resultados obtidos pelo MaOPSO são estatisticamente diferentes e possivelmente relevantes comparado aos resultados obtidos pelos outros algoritmos. Para isso, utilizou-se o teste estatístico não-paramétrico de Wilcoxon baseado na *mediana* com um *p-value* de 5%. Esse teste foi aplicado de modo a comparar o MaOPSO com o cada um dos algoritmos em cada uma das métricas de avaliação de desempenho, a saber, o HV, GD e IGD. Os resultados são apresentados na Tabela 3. Essa tabela mostra o número de problemas em que um algoritmo foi estatisticamente superior (B), inferior (W) ou teve um desempenho similar (E) aos outros algoritmos. Como o número total de problemas (instâncias de problemas) é 30 (seis problemas vezes cinco possibilidades de número de objetivos), a soma do número de instâncias em que o MaOPSO foi superior, inferior ou similar deve ser 30. Pela Tabela 3, pode-se observar que o MaOPSO obteve um resultado global bastante expressivo. Se por uma lado, ele venceu (obteve melhores resultados) o NSGA-III e o SMPSO em um maior número de instâncias em termos de convergência (GD) e perdeu para o MDFA, CEGA e CDAS-SMPSO. Por outro lado, o MaOPSO ganhou em termos de diversidade (IGD e HV) em comparação a todos os seus concorrentes. Dessa forma, o MaOPSO foi superior a todos os algoritmos em termos de diversidade e teve um desempenho pior em termos de convergência justamente quando ele é comparado com os algoritmos que convergem para um pequena região da Frente de Pareto (às vezes um ponto) como será exposto na subseção 4.2.5.

A Tabela 4 mostra de forma compacta os resultados para as métricas GD, IGD, e HV comparando o MaOPSO e outros algoritmos com respeito a 1 000 iterações. Desta forma, utilizou-se a mesma metodologia estatística já apresentada. Contudo, nesta nova análise o que se pretende é avaliar o desempenho global do MaOPSO usando um número

Tabela 3 – Número de vezes em que o MaOPSO foi estatisticamente superior (B), igual (E) ou inferior (W) aos outros algoritmos considerando 100 000 iterações.

| | | Algoritmos com 100,000 iterações. | | | | | |
|-----|--------|-----------------------------------|---------|-------|------|------|------------|
| | | | NSGAIII | SMPSO | MDFA | CEGA | CDAS-SMPSO |
| GD | MaOPSO | B | 30 | 30 | 7 | 12 | 10 |
| | | E | 0 | 0 | 0 | 0 | 0 |
| | | W | 0 | 0 | 23 | 18 | 20 |
| IGD | MaOPSO | B | 27 | 25 | 30 | 30 | 30 |
| | | E | 0 | 0 | 0 | 0 | 0 |
| | | W | 3 | 5 | 0 | 0 | 0 |
| HV | MaOPSO | B | 26 | 22 | 29 | 30 | 30 |
| | | E | 0 | 0 | 0 | 0 | 0 |
| | | W | 4 | 8 | 1 | 0 | 0 |

número baixo de iterações e conseqüentemente um número limitado de avaliações de aptidão. É importante enfatizar novamente que todos os algoritmos utilizaram exatamente o mesmo número de avaliação de aptidão. Sendo assim, a partir da Tabela 4, pode-se concluir que o MaOPSO apresenta o melhor compromisso entre convergência e diversidade entre todos os algoritmos utilizados na comparação. Pode-se também concluir que o CEGA apresenta o pior desempenho para poucas iterações.

Tabela 4 – Número de vezes em que o MaOPSO foi estatisticamente superior (B), igual (E) ou inferior (W) aos outros algoritmos considerando 1 000 iterações.

| | | Algoritmos com 1,000 iterações. | | | | | |
|-----|--------|---------------------------------|---------|-------|------|------|------------|
| | | | NSGAIII | SMPSO | MDFA | CEGA | CDAS-SMPSO |
| GD | MaOPSO | B | 19 | 21 | 2 | 2 | 4 |
| | | E | 1 | 4 | 1 | 2 | 10 |
| | | W | 10 | 5 | 27 | 26 | 16 |
| IGD | MaOPSO | B | 16 | 22 | 27 | 27 | 30 |
| | | E | 5 | 4 | 2 | 2 | 0 |
| | | W | 9 | 4 | 1 | 1 | 0 |
| HV | MaOPSO | B | 24 | 22 | 24 | 27 | 30 |
| | | E | 2 | 2 | 3 | 3 | 0 |
| | | W | 4 | 6 | 3 | 0 | 0 |

4.2.5 Visualização das Frentes de Pareto

Esta seção apresenta alguns exemplos de conjuntos aproximação dos algoritmos MaOPSO, CEGA, MDFA, SMPSO, CDAS-SMPSO e NSGA-III. O objetivo é fornecer uma visualização gráfica da convergência e da diversidade dos conjuntos aproximação dos algoritmos. As Figuras de 56 até 61 apresentam os conjuntos aproximação produzidos pelos algoritmos após 100 000 iterações em uma única execução para os problemas DTLZ com três objetivos. Esses conjuntos aproximação correspondem ao caso da mediana do GD para cada problema. Nessas figuras, os conjuntos aproximação são apresentadas juntamente com as Frentes de Pareto correspondentes aos problemas para que se possa realizar uma análise visual da convergência, cobertura e distribuição das soluções.

As Figuras de 63 até 68 apresentam os conjuntos aproximação produzidos pelos algoritmos em um única execução para os problemas DTLZ com dez objetivos. Novamente, esses conjuntos aproximação correspondem ao caso da mediana do GD para cada problema. As Figuras de 63 até 68 mostram os conjuntos aproximação usando coordenadas paralelas. Coordenadas paralelas são gráficos utilizados para análise de dados multidimensionais. Nesse tipo de gráfico, os objetivos estão dispostos no eixo horizontal em uma sequência qualquer. O eixo vertical do gráfico é uma escala numérica para indicar o valor de cada objetivo. Sendo assim, o gráfico é construído do seguinte modo. Para cada ponto multidimensional, o valor de cada componente do ponto é marcado no eixo paralelo ao eixo vertical exatamente na altura do valor correspondente. Feito isso, os pontos marcados em cada eixo vertical de cada objetivo são conectados formando uma linha poligonal que é a representação do ponto multidimensional em coordenadas paralelas. Para que se possa fazer uma análise visual entre os conjuntos aproximação e as Frentes de Pareto dos problemas em termos de convergência, cobertura e distribuição, uma amostragem das Frentes de Pareto de cada problema é plotada na Figura 62. Essa amostragem consiste exatamente no conjunto de referência de cada problema usado no cálculo do IGD. Como o conjunto referência P^* é um subconjunto bem distribuído da Frente de Pareto, ele ilustra bem a geometria da Frente de Pareto do problema quando plotado em coordenadas paralelas.

Como se pode observar a partir das figuras de Figura 56 a Figura 59 para três objetivos e das figuras de Figura 63 a Figura 66 para o 10 objetivos, o MaOPSO foi capaz de obter um conjunto aproximação bem distribuído e próximo da Frente de Pareto para os problemas DTLZ $\{1,2,3,4\}$ tanto para poucos objetivos (três objetivos) como para muitos objetivos (dez objetivos). Para três objetivos, esta visualização é direta: basta confrontar a Frente de Pareto representada pelas superfícies geométricas de cada problema e o conjunto aproximação (pontos sobre essas superfícies geométricas). É importante destacar que os resultados obtidos pelo NSGA-III são diferentes dos resultados presentes em (DEB; JAIN, 2014). Isso se explica pelo fato de que não se foi utilizado o tamanho da população de indivíduos recomendado em (DEB; JAIN, 2014), pois isso implicaria em uma comparação

não justa em termos do número de indivíduos utilizados entre o MaOPSO e o NSGA-III.

Para dez objetivos, por outro lado, é necessário fazer um comparativo entre o aspecto dos conjuntos aproximação dos algoritmos com as representações da Frente de Pareto na Figura 62. Por outro lado, os algoritmos NSGA-III, CEGA, MDFA e CDAS-SMPSO apresentaram, para esses mesmos problemas, um conjunto aproximação com problemas de convergência, cobertura e espalhamento dependendo do problema. Por exemplo, o problema da convergência fica claro ao se observar os conjuntos aproximação dos algoritmos SMPSO, NSGA-III e CEGA no problema DTLZ3 na Figura 65. Nesta mesma figura, fica clara também a boa convergência do algoritmo MaOPSO.

Os problemas DTLZ5 e DTLZ6, devido à sua natureza, impõem mais dificuldades que os problemas anteriores, pois eles apresentam uma Frente de Pareto unidimensional imersa em um espaço multidimensional. Para o problema DTLZ5 com três objetivos (Figura 60), o MAOPSO converge para a Frente de Pareto mas a sua cobertura é limitada pois as soluções se concentram na parte superior da Frente de Pareto. O único algoritmo que consegue uma boa cobertura juntamente com uma boa distribuição é o SMPSO. O NSGA-III consegue convergir para a Frente de Pareto mas as soluções são mal distribuídas. Para o problema DTLZ5 com dez objetivos (Figura 67), todos os algoritmos falharam em encontrar a Frente de Pareto. Essa afirmação é derivada a partir do confronto entre a visualização do conjunto de referência P^* em coordenadas paralelas na Figura 62 e os conjuntos aproximação. Para o problema DTLZ6 com três objetivos (Figura 61), o MaOPSO novamente não conseguiu uma boa cobertura da Frente de Pareto apesar de ter convergido para ele. O SMPSO novamente conseguiu convergir e ter uma boa cobertura e distribuição de soluções. Por outro lado, o NSGA-III convergiu para uma Frente de Pareto local. Finalmente, para o problema DTLZ6 com dez objetivos (Figura 68), todos os algoritmos falharam em convergir para a Frente de Pareto. Como se pode perceber, o desempenho do MaOPSO não é tão bom nestes problemas como nos demais. Isso se deve a natureza desses problemas, que são problemas degenerados pois a Frente de Pareto é unidimensional enquanto a Frente de Pareto esperada deveria ter $m - 1$ dimensões. Esse fato faz com que algumas soluções extremas da Frente de Pareto não fiquem nos eixos dos objetivos, e com isso o MaOPSO não consegue localizar todos os extremos e consequentemente ele não consegue determinar o hiperplano e portanto o algoritmo deixa de funcionar como esperado.

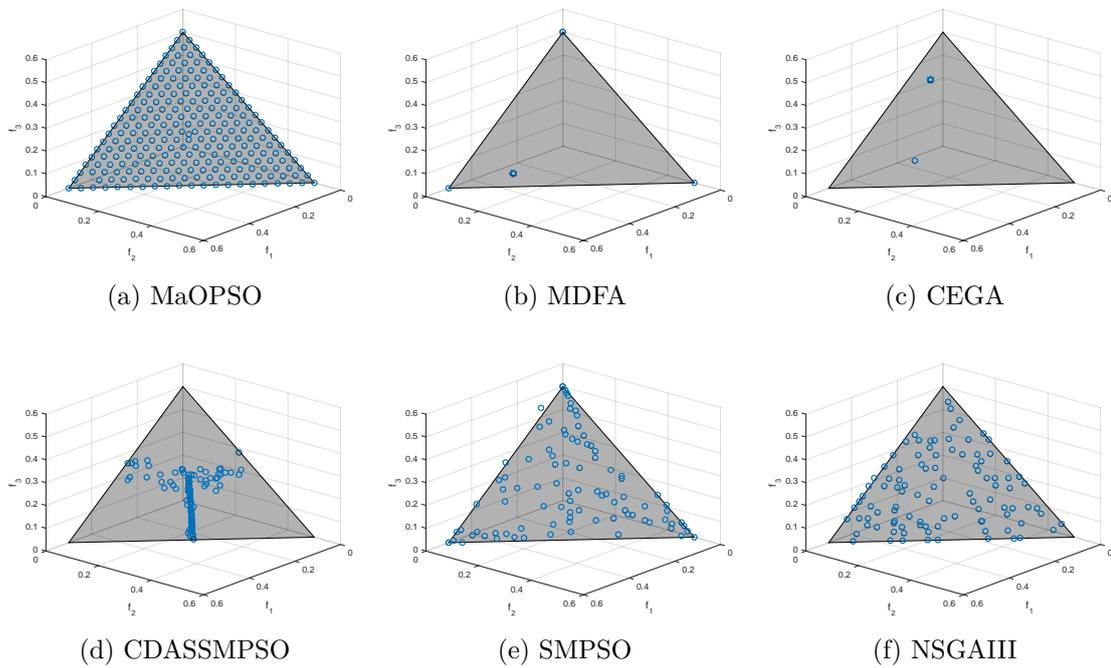


Figura 56 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ1 com três objetivos.

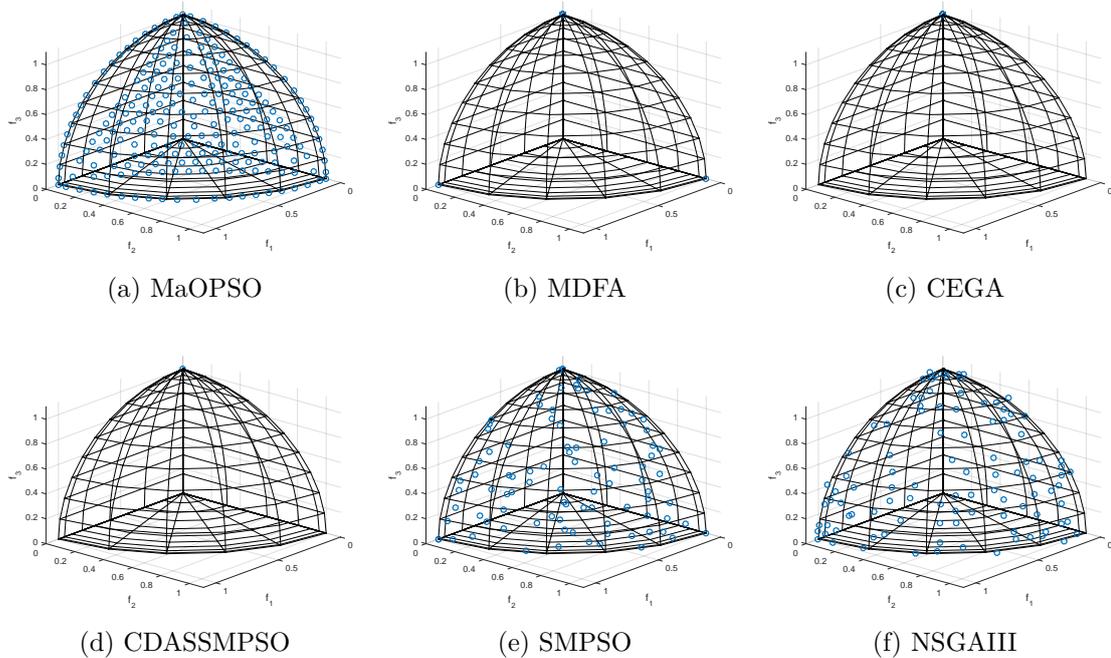


Figura 57 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ2 com três objetivos.

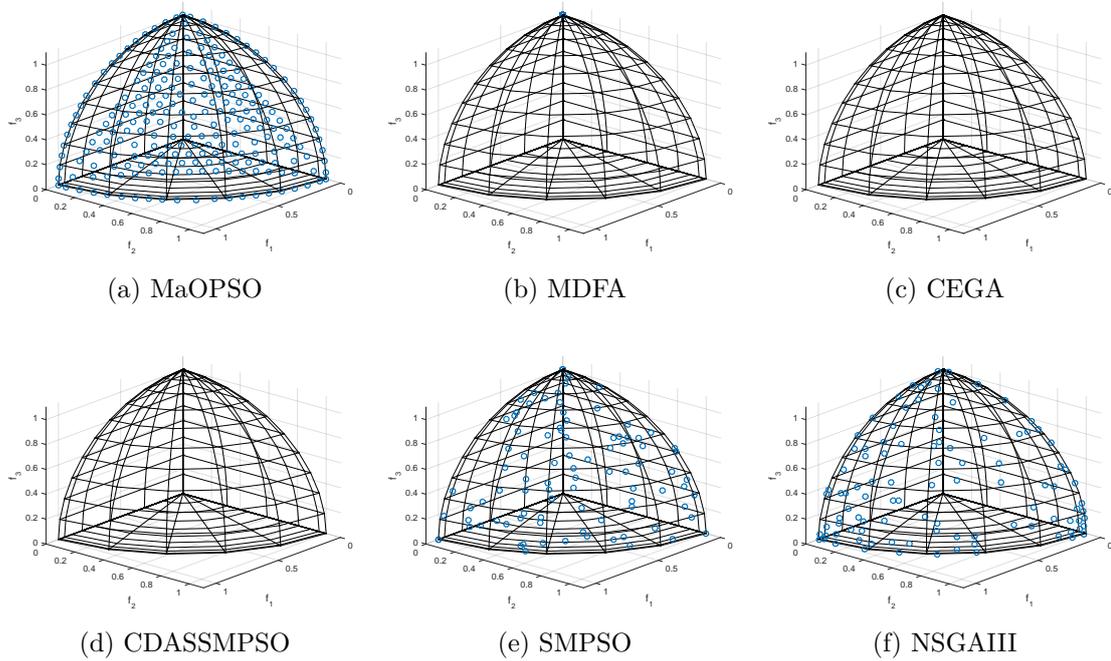


Figura 58 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ3 com três objetivos.

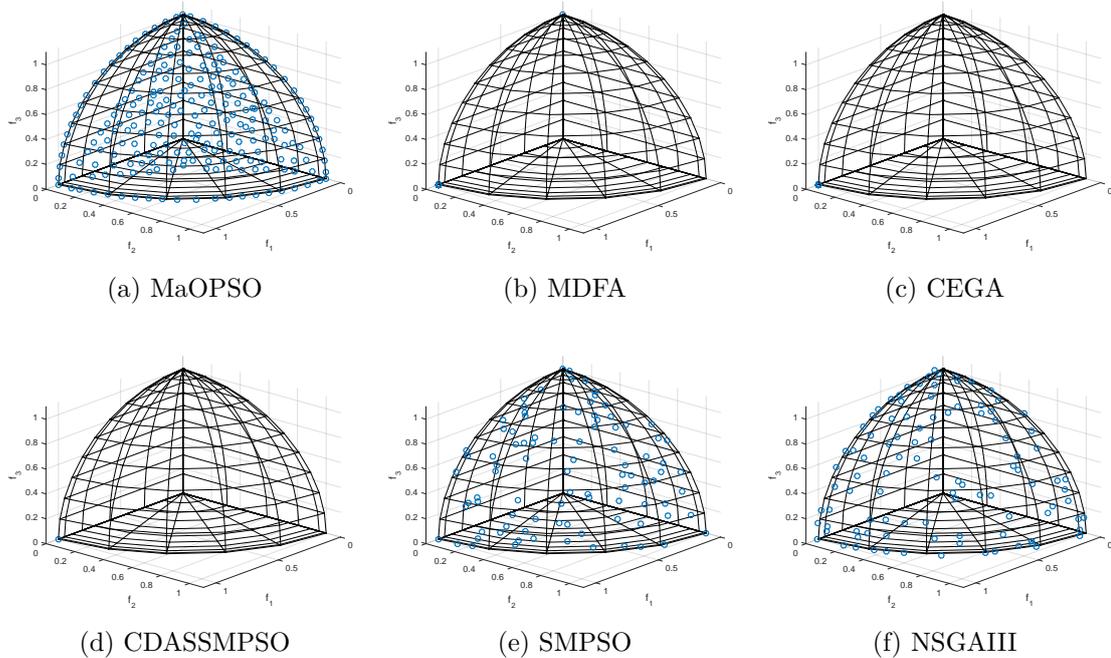


Figura 59 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ4 com três objetivos.

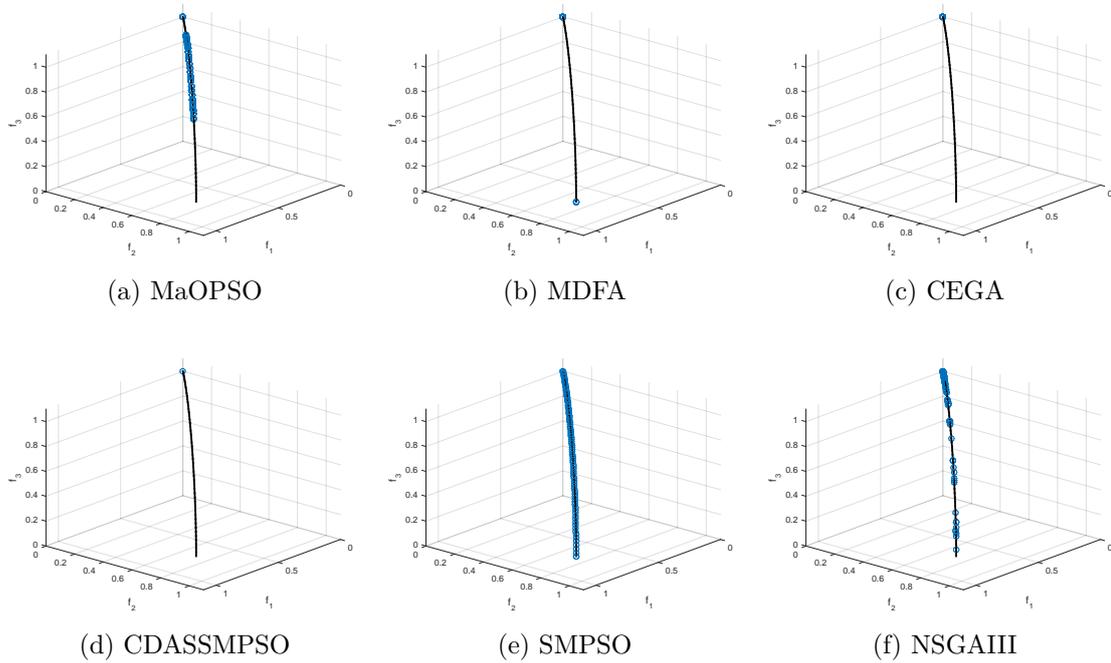


Figura 60 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ5 com três objetivos.

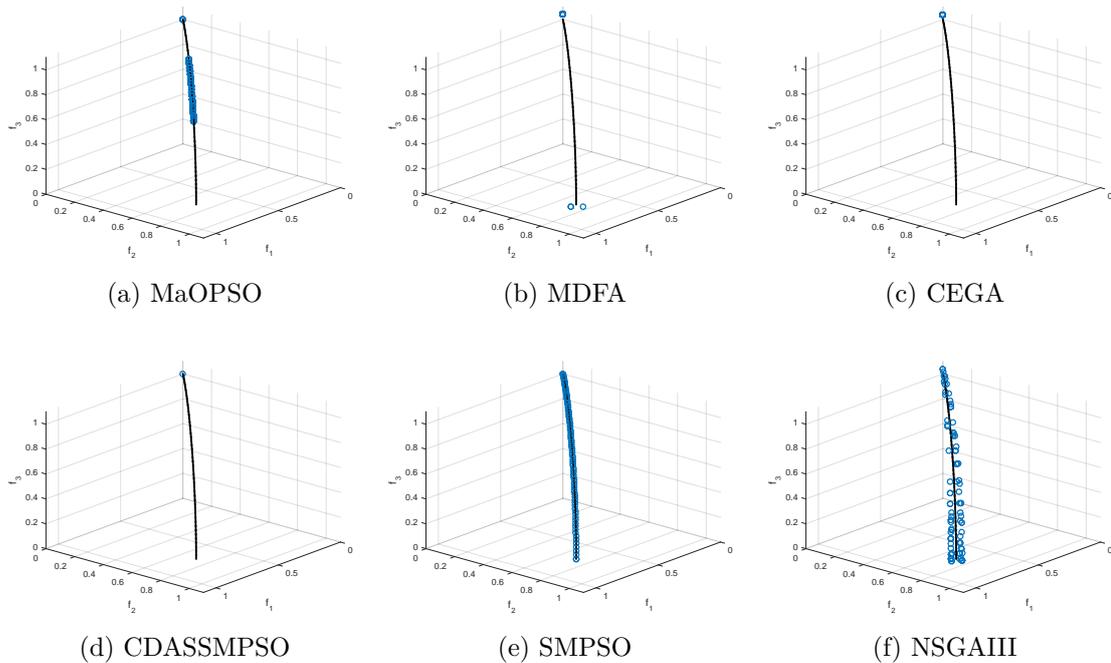


Figura 61 – Conjunto aproximação dos algoritmos após 100 000 iterações para o problema DTLZ6 com três objetivos.

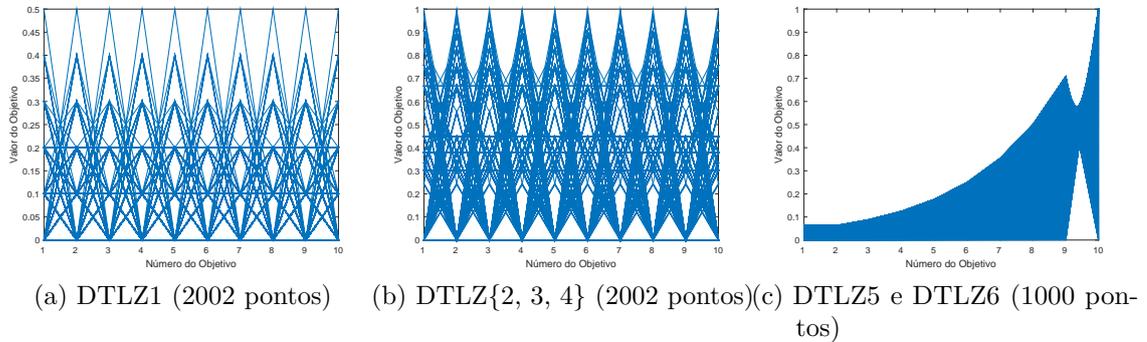


Figura 62 – Coordenadas paralelas dos conjuntos referências dos algoritmos para os problemas de teste com dez objetivos.

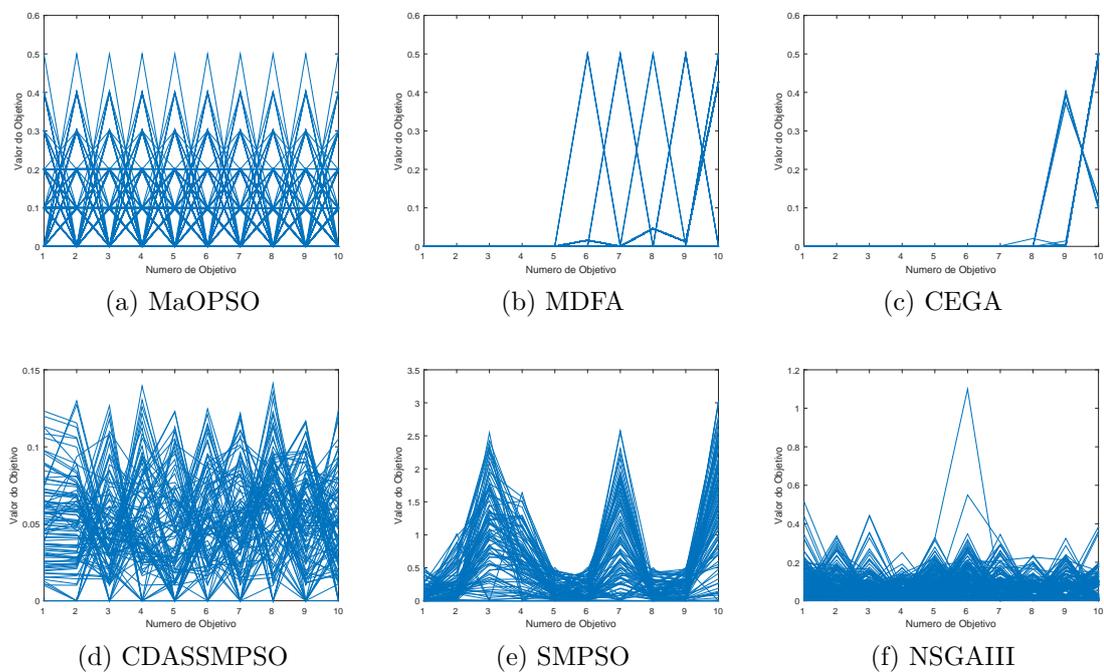


Figura 63 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ1 com dez objetivos.

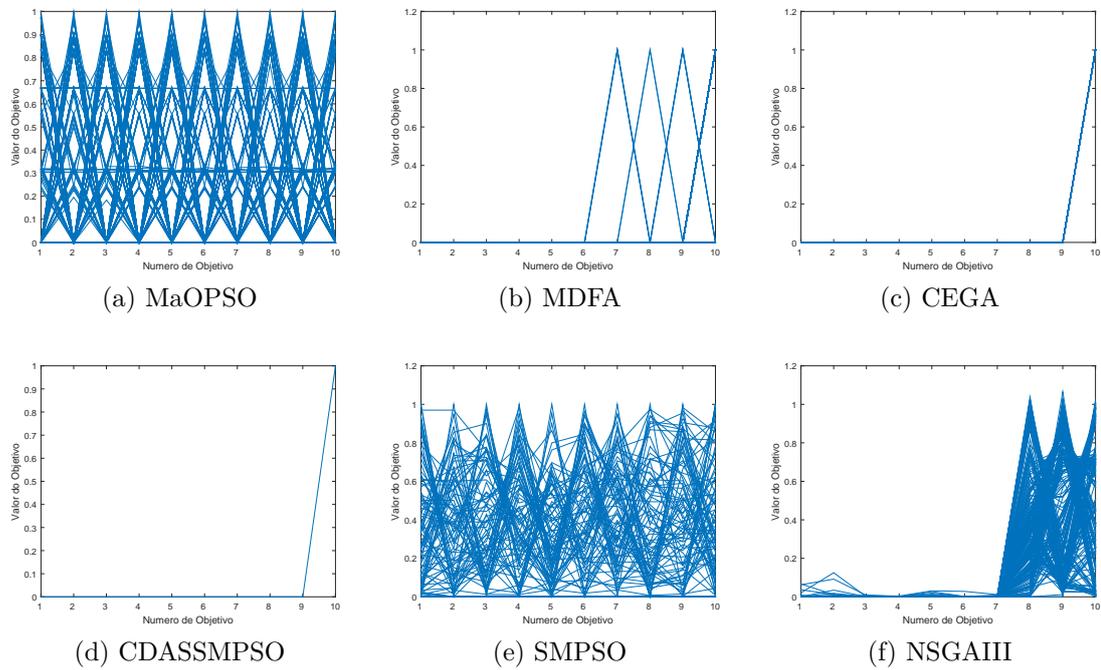


Figura 64 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ2 com dez objetivos.

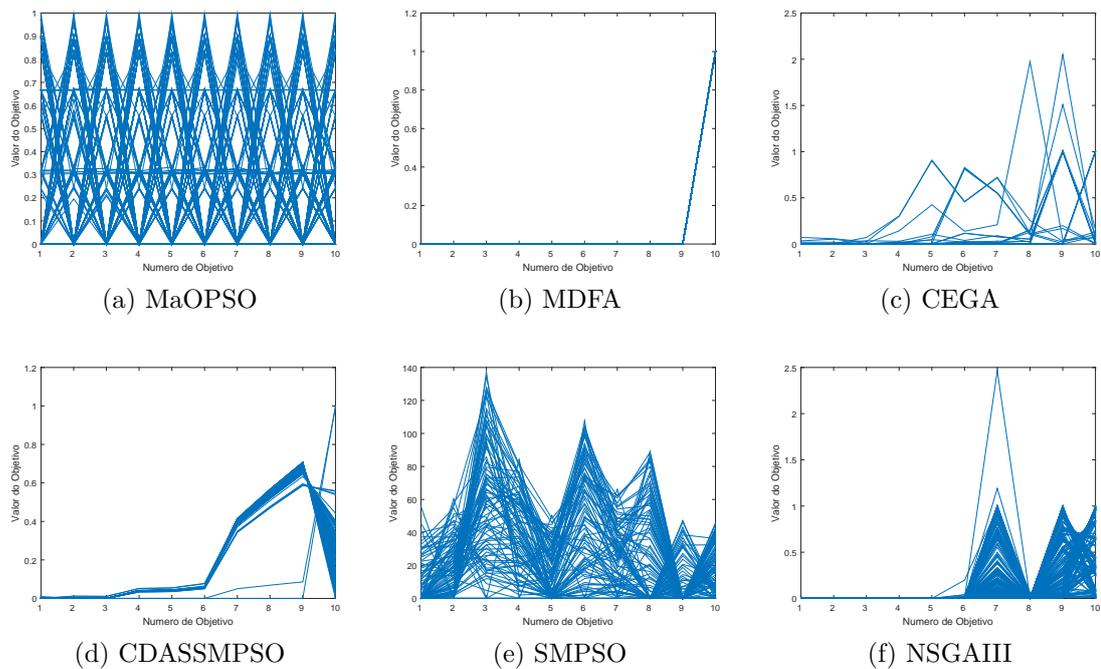


Figura 65 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ3 com dez objetivos.

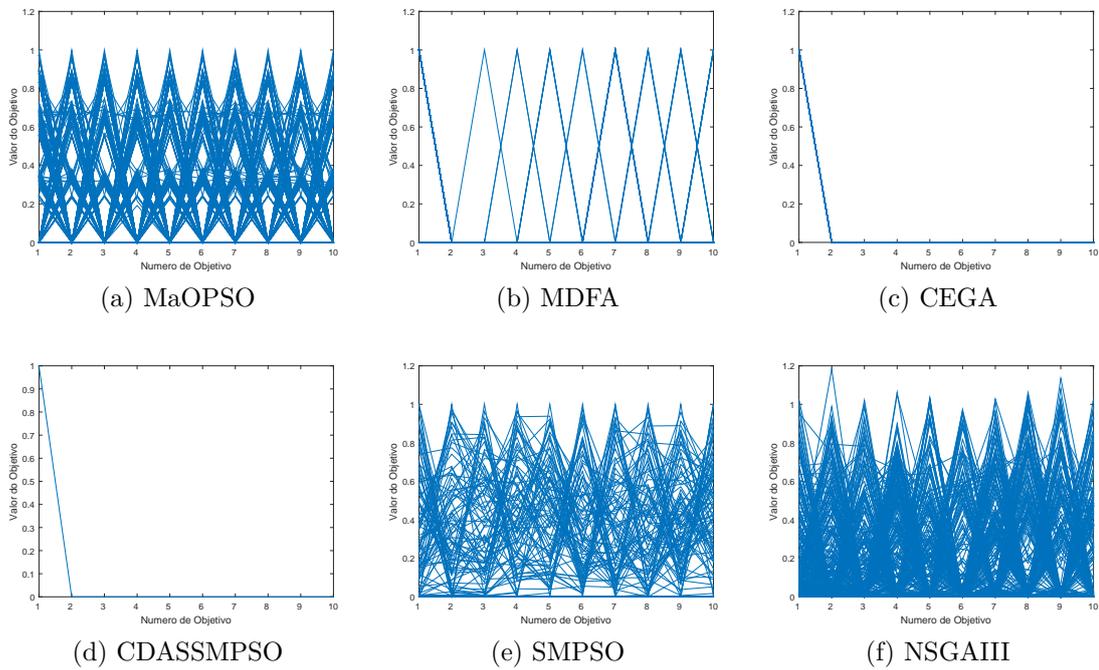


Figura 66 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ4 com dez objetivos.

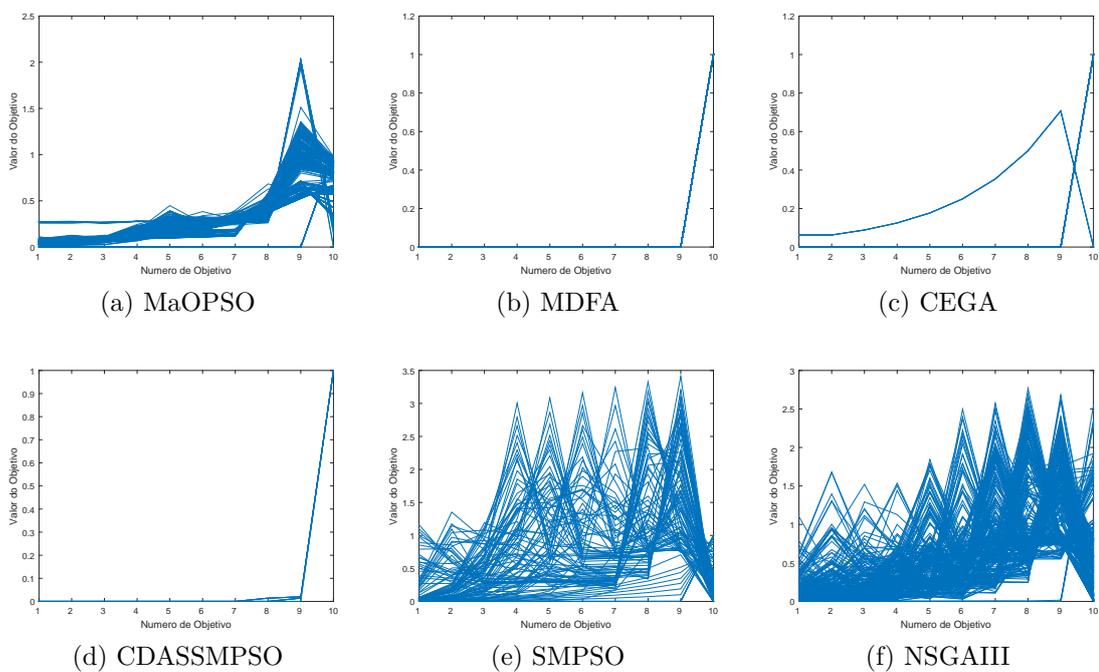


Figura 67 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ5 com dez objetivos.

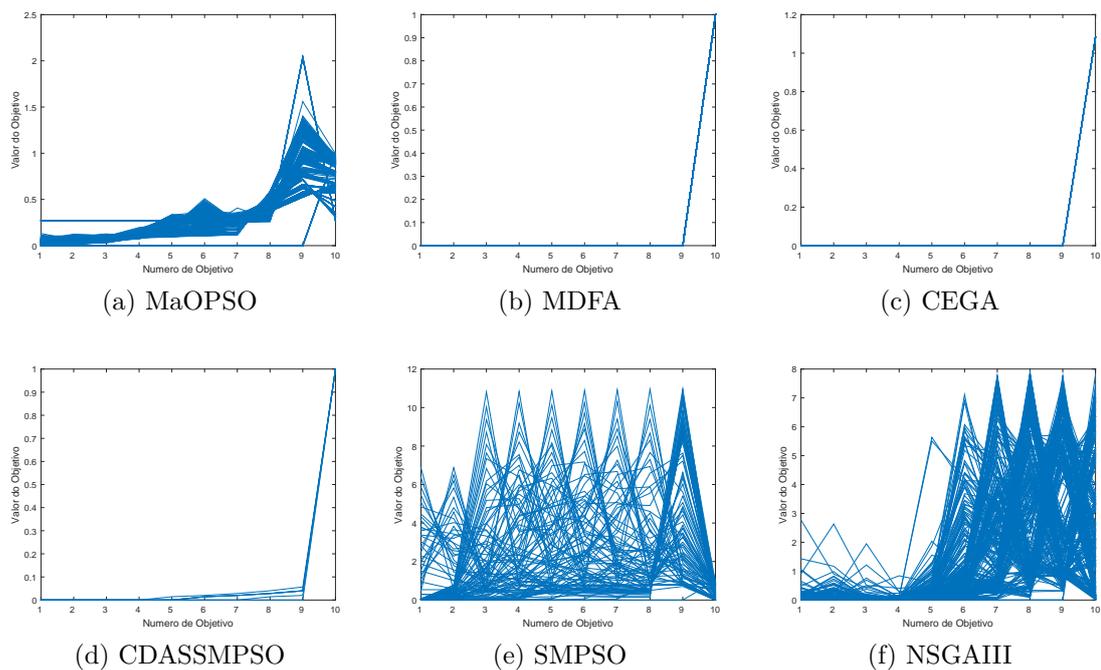


Figura 68 – Coordenadas paralelas dos conjuntos aproximação dos algoritmos após 100 000 iterações para o problema DTLZ6 com dez objetivos.

4.2.6 Análise do Tempo de Execução

Nesta seção, analisou-se o tempo de execução dos algoritmos sobre os problemas DTLZ considerando apenas 1000 iterações. O objetivo foi apenas ter uma estimativa do tempo de execução dos algoritmos. A Tabela 5 mostra a média e o desvio padrão (entre parênteses) dos tempos de execução dos algoritmos para todos os problemas DTLZ para dez objetivos (o pior caso) sobre 31 execuções independentes. A partir desta tabela, pode-se observar que o CEGA é o algoritmo com o pior tempo de execução com um tempo de execução de aproximadamente 6 000 segundos (1,67 horas), enquanto o CDAS-SMPSO apresenta o melhor tempo de execução (aproximadamente 6 segundos por execução). O MDFA apresentou um tempo de execução médio em torno de 60 segundos (1 minuto). MaOPSO e NSGA-III apresentam um desempenho similar entre si com um tempo de execução médio de aproximadamente 150 segundos (um pouco menos que dois minutos). Desta forma, MaOPSO é um algoritmo tão eficiente quanto o NSGA-III e mais eficiente que o CEGA apresentando um custo computacional um pouco mais elevado que o MDFA e o CDAS-SMPSO. Portanto, o bom desempenho do MaOPSO em termos de convergência e diversidade não aumentou muito seu custo computacional sendo seu custo computacional comparado ao algoritmo NSGA-III.

Tabela 5 – Média e desvio padrão do tempo de execução dos algoritmos em segundos.

| Problema | MaOPSO | NSGAIII | SMPSO |
|----------|----------------|------------------|---------------|
| DTLZ1 | 110.27 (34.38) | 134.59 (39.11) | 16.87 (3.69) |
| DTLZ2 | 220.18 (63.90) | 148.64 (42.18) | 57.42 (10.67) |
| DTLZ3 | 153.76 (38.55) | 127.87 (38.21) | 31.00 (5.45) |
| DTLZ4 | 240.35 (64.47) | 163.77 (45.70) | 63.93 (11.42) |
| DTLZ5 | 153.86 (40.79) | 152.75 (42.91) | 21.51 (3.67) |
| DTLZ6 | 152.34 (37.65) | 150.52 (42.75) | 42.40 (9.86) |
| Problema | MDFA | CEGA | CDAS-SMPSO |
| DTLZ1 | 57.20 (11.91) | 4958.29 (89.78) | 4.13 (2.03) |
| DTLZ2 | 59.60 (12.19) | 6030.71 (72.50) | 6.68 (3.45) |
| DTLZ3 | 60.10 (12.43) | 6021.73 (80.59) | 4.27 (1.42) |
| DTLZ4 | 66.72 (13.34) | 5912.98 (75.32) | 9.56 (0.66) |
| DTLZ5 | 61.13 (11.29) | 6032.47 (105.84) | 5.12 (0.80) |
| DTLZ6 | 58.18 (11.44) | 5808.68 (156.10) | 6.00 (1.24) |

4.3 Estudo de Caso: Aplicação do MaOPSO no Problema de Projeto de Redes Ópticas

Nesta seção, é apresentado um exemplo de aplicação do MaOPSO em um problema real de otimização com muitos objetivos. O problema selecionado foi o projeto de redes ópticas.

O problema do projeto de redes ópticas pode ser caracterizado da seguinte maneira. Dado a posição dos nós da rede, a matriz de tráfico e o algoritmo de atribuição de comprimento de onda e roteamento, o objetivo do problema consiste em encontrar a topologia física e a especificação dos dispositivos ópticos que deveriam ser empregados na rede afim de simultaneamente minimizar o custo de implementação da rede, denotado por *Capital Expenditure* (CAPEX), a probabilidade de bloqueio (PB), o consumo de energia e afim de maximizar a robustez da rede (resiliência da rede). A robustez da rede é a capacidade da rede de continuar em operação mesmo após a falha de uma ou mais conexões. Uma forma de medir isso objetivamente é por meio do cálculo da conectividade algébrica da rede denotada por λ_{N-1} , que é calculada conforme (LEWIS, 2009). Escolheu-se os seguintes parâmetros da rede como variáveis de decisão: a topologia da rede, a especificação dos *Optical Cross-Connectors* (OXC) empregados na rede e o número de comprimentos de onda nos enlaces. As posições dos nós são definidos por $P = (x_i, y_i)$, em que $(i \in 1, 2, \dots, N)$, x_i e y_i são as coordenadas cartesianas do i^{esimo} nó e N é o número total de nós da rede. O modelo computacional usa uma matriz de adjacência A com elementos $a_{i,j}$, em que

($a_{i,j} \in \{0, 1\}$). A matriz de adjacência A é mostrada na Equação 4.4. Se $a_{i,j} = 0$, os nós da rede i e j não estão conectados, caso contrário eles são conectados e o enlace correspondente usa um amplificador óptico pré-definido.

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,N} \\ a_{2,1} & a_{2,2} & \dots & a_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N,1} & a_{N,2} & \dots & a_{N,N} \end{bmatrix}. \quad (4.4)$$

Os enlaces entre cada par de nó são bidirecionais, ou seja, $a_{i,j} = a_{j,i}$. Como a matriz A é simétrica ($a_{i,j} = a_{j,i}$), pode-se representar esta matriz como um vetor pela justaposição, lado a lado, dos elementos não-nulos da matriz triangular superior da matriz A . O vetor usado para representar a rede é definida como $\mathbf{X} = \{x_i\}$, em que ($i \in 1, 2, \dots, K + 2$), em que $K = \frac{N^2 - N}{2}$. Além dos elementos da matriz A , mais dois elementos foram acrescentados no final desse vetor para formar então o vetor de decisão. O elemento $x_{K+1} = \delta$ representa a especificação OXC e o elemento $x_{K+2} = w$ representa o número de comprimentos de onda. Além disso, considerou-se na formulação do problema que o grafo que representa a rede óptica deve ser sempre conectado. Esta restrição é necessária para assegurar a comunicação de qualquer par arbitrário de nós. Finalmente, pode-se resumir a descrição do problema do projeto de redes ópticas como segue:

Descrição do problema: Dada as variáveis de decisão representadas por

$$\mathbf{X} = [a_{1,2} \ a_{1,3} \ \dots \ a_{N-1,N} \ \delta \ w], \quad (4.5)$$

o que se deseja é encontrar os valores para essas variáveis com o objetivo de minimizar as seguintes funções:

$$\text{Minimizar} \begin{cases} f_1(\mathbf{X}) = PB \\ f_2(\mathbf{X}) = \text{CAPEX} \\ f_3(\mathbf{X}) = \text{Consumo de energia} \\ f_4(\mathbf{X}) = 1/(1 + \lambda_{N-1}) \end{cases}, \quad (4.6)$$

sujeito a:

$$\lambda_{N-1} > 0, \quad (4.7)$$

em que PB é a probabilidade de bloqueio, CAPEX é o custo da rede e λ_{N-1} é a conectividade algébrica (LEWIS, 2009) do grafo que representa a rede. A conectividade algébrica é um indicador versátil porque ele assegura que a topologia da rede óptica é um grafo

conectado quando seu valor é maior que zero e a rede é mais resiliente quando seu valor é crescente. Assim, pode-se usar a conectividade algébrica para impor a restrição de conectividade da rede e para analisar a tolerância a falhas da rede quanto a falha de um enlace. Como já foi mencionado, a conectividade algébrica é diretamente proporcional a resiliência da rede. Sendo assim, afim de minimizar todos os objetivos, o quarto objetivo foi definido como o inverso da conectividade algébrica tomando o cuidado de acrescentar o valor 1 no denominador para se evitar a divisão por zero. É importante mencionar os valores assumidos pela especificação OXC δ e pelo número de comprimentos de onda. A partir dos trabalhos de (ARAÚJO et al., 2011; ARAÚJO, 2015; ARAÚJO; BASTOS-FILHO; MARTINS-FILHO, 2015; BASTOS-FILHO et al., 2011; CHAVES; BASTOS-FILHO; MARTINS-FILHO, 2010), verificou-se o uso recorrente da especificação OXC entre os valores $\delta \in \{1, 2, 3, 4, 5\}$, as especificações de cada tipo de OXC pode ser encontrada em (BASTOS-FILHO et al., 2011). Finalmente, o número comprimento de onda $w \in \{4, 5, 6, \dots, 39, 40\}$ (BASTOS-FILHO et al., 2011).

Como se pode perceber, o problema do projeto de redes ópticas é um problema com muitos objetivos e constitui um problema prático do mundo real que serve de estudo de caso para a aplicação do MaOPSO. Para se modelar uma rede e simular o seu funcionamento foi utilizado simulador de redes ópticas denominado SIMTON que simula uma rede com 14 nós. Ou seja, a rede em questão tem no máximo $14(14 - 1)/2 = 91$ enlaces. Desse modo, o vetor de decisão do problema possui $91 + 2$ componentes. Como o MaOPSO funciona com variáveis contínuas, o vetor decisão do MaOPSO precisa passar por um processo de tradução entre o vetor de decisão efetivamente usado pelo MaOPSO no processo de busca para o vetor de decisão usado no simulador de redes ópticas. Para o processo de tradução foi usado o seguinte estratagema. Os 91 componentes iniciais do vetor de decisão do MaOPSO assumem valores entre -1 e 1. Esses componentes se tornam os 91 componentes do vetor de decisão usado no simulador usando a seguinte regra. Seja x_i o valor do componente i do vetor de decisão do MaOPSO, então se $x_i > 0$ então ele é traduzido para um valor 1 no componente correspondente do vetor de decisão do simulador. Caso contrário ($x_i \leq 0$) então ele é traduzido para 0. Resta agora resolver as suas últimas variáveis: a especificação OXC e o número de comprimentos de onda usado nos enlaces. Como essas variáveis são inteiros não-negativos no vetor de decisão do simulador, suas representações binárias, com 3 e 6 bits respectivamente, podem ser utilizadas usando o mesmo esquema anterior e tomando cuidado para que valores binários que assumam valores inteiros acima ou abaixo do permitido assumam o valor dos limites superiores e inferiores dependendo do caso. Por exemplo, a especificação OXC está entre 1 e 5, logo apenas três bits são necessários para representar todos os valores inteiros desta variável do simulador. Já para o número de comprimentos de onda w que varia entre 4 e 40, o número de bits necessário para representar todos os valores de entrada do simulador é 6 bits. Com isso, o vetor de decisão do MaOPSO que possui $91 + 3 + 6 = 100$ variáveis contínuas é convertido em um vetor de

decisão com 93 variáveis inteiras usadas no simulador. Sendo assim, a partir de tudo que foi exposto, pode-se dizer que o tamanho do espaço de busca explorado pelo MaOPSO é 2^{100} , o que obviamente é um espaço de busca enorme e impossível de ser tratado por algum método exato, o que torna, com isso, a aplicação de uma meta-heurística algo necessária para resolução deste problema prático.

Colocando o problema nesses termos, o MaOPSO foi executado 20 vezes usando as seguintes configurações. O MaOPSO foi executado com 5000 iterações e utilizando um número de divisões igual a 7 o que resulta em 120 pontos. O resto de seus parâmetros foi mantido conforme experimentos anteriores. Para permitir um comparativo com o MaOPSO, o algoritmo NSGA-III também foi executado nesse problema com os mesmos parâmetros dos experimentos anteriores e também usando 5000 iterações. Escolhe-se o NSGA-III pois ele é o principal competidor do MaOPSO já que ambos são baseados em conjuntos de referência. Além disso, NSGA-III é um algoritmo para otimização de muitos objetivos ao contrário, por exemplo, do SMPSO, que apresenta geralmente resultados satisfatórios apenas para problemas com dois ou três objetivos. Por isso, o SMPSO, mesmo sendo uma técnica de enxames de partículas como o MaOPSO, foi descartado para a solução deste problema que tem quatro objetivos. Finalmente, o NSGA-III foi escolhido porque ele já foi aplicado em cenários reais em (DEB; JAIN, 2014), obtendo resultados promissores em dois problemas práticos desafiadores (com escalas diferentes nos objetivos e/ou descontínuos) (DEB; JAIN, 2014), tanto para poucos objetivos (três objetivos) como para muitos objetivos (nove objetivos). Por outro lado, tanto o CEGA, MDFA e CDAS-SMPSO não foram aplicados em problemas reais, e, além disso, esses algoritmos têm parâmetros difíceis de ajudar, como ocorre com o CDAS-SMPSO, ou são muito lentos, como é o caso do CEGA, ou apenas convergem em problemas bem comportados, como é o caso do MDFA. Desta forma, diante de tudo que foi exposto, o NSGA-III foi o único algoritmo utilizado para comparação com o MaOPSO neste cenário experimental que envolveu a otimização da configuração de uma rede óptica.

A Figura 69 mostra o valor médio do hipervolume de 1 a 5000 iterações tanto para o NSGA-III e o MaOPSO. Aplicou-se o teste de Wilcoxon não-paramétrico com $p_{value} = 0,05$ de significância para verificar se há uma diferença estatística entre o hipervolume do MaOPSO e do NSGA-III. Com este teste, verificou-se que o MaOPSO é estatisticamente superior ao NSGA-III. E, depois de 5000 gerações, o hipervolume obtido pelo MaOPSO é 9% mais alto que os valores obtidos pelo NSGA-III. O que mostra que o MaOPSO conseguiu evoluir o seu HV com relação ao NSGA-III, apesar de evoluir muito lentamente.

A Figura 70 mostra os conjuntos aproximação obtidos pelos algoritmos depois de 5000 iterações para o NSGA-III e o MaOPSO. O objetivo 1 é a probabilidade de bloqueio da rede óptica. O objetivo 2 é o valor normalizado do CAPEX. O objetivo 3 é o valor

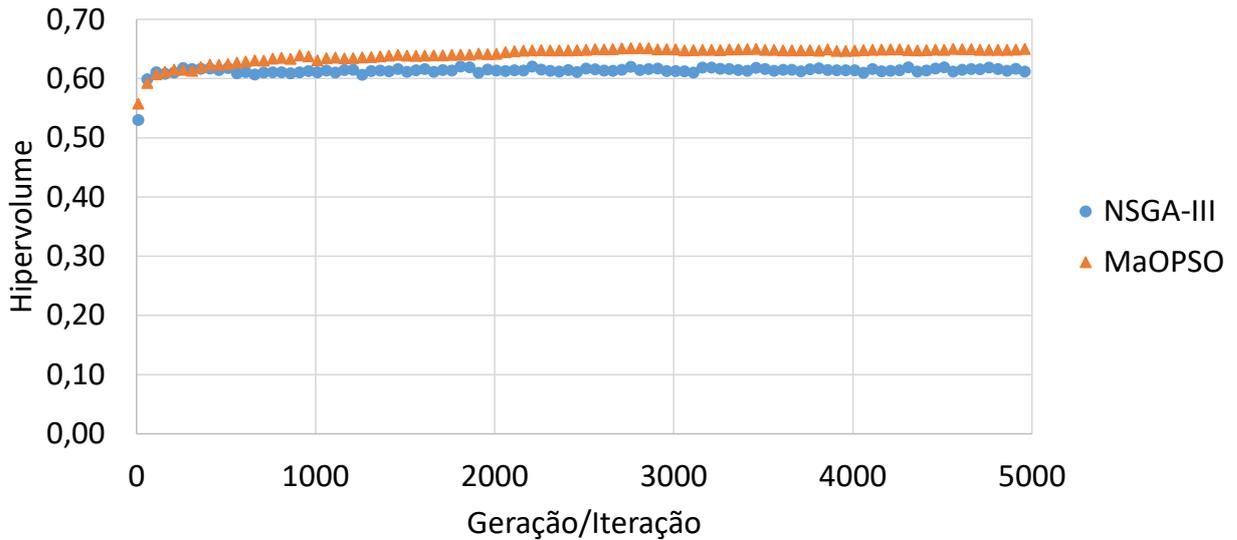
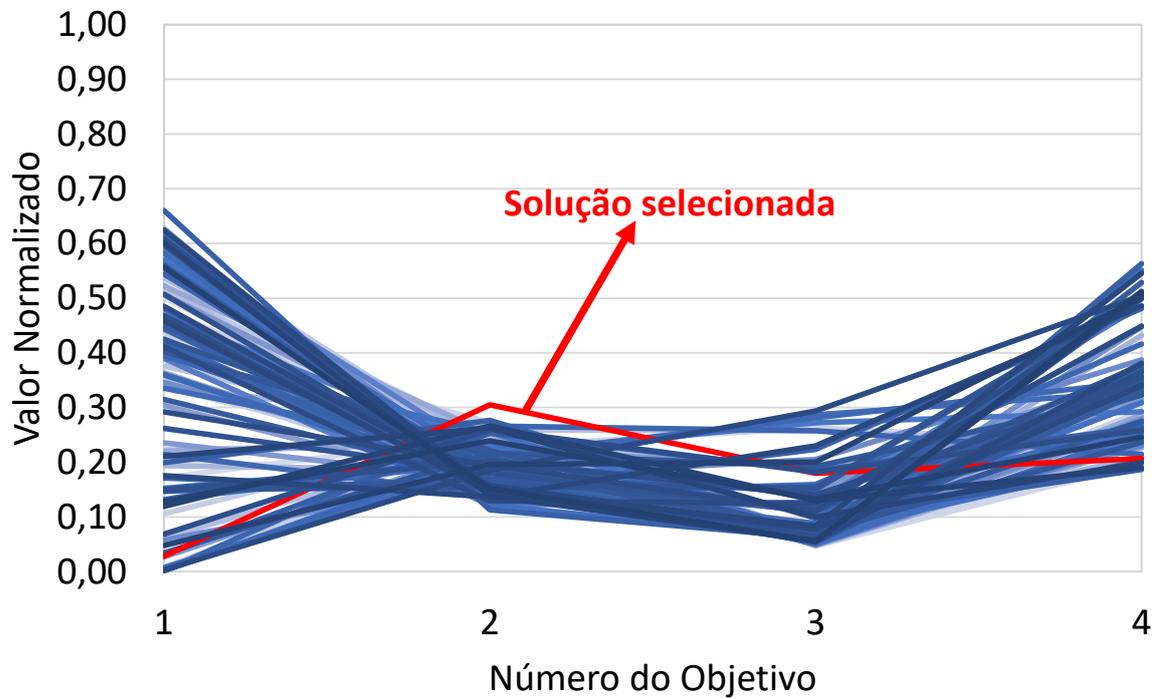


Figura 69 – Média do hipervolume ao longo das 5000 gerações para o NSGA-III e MaOPSO.

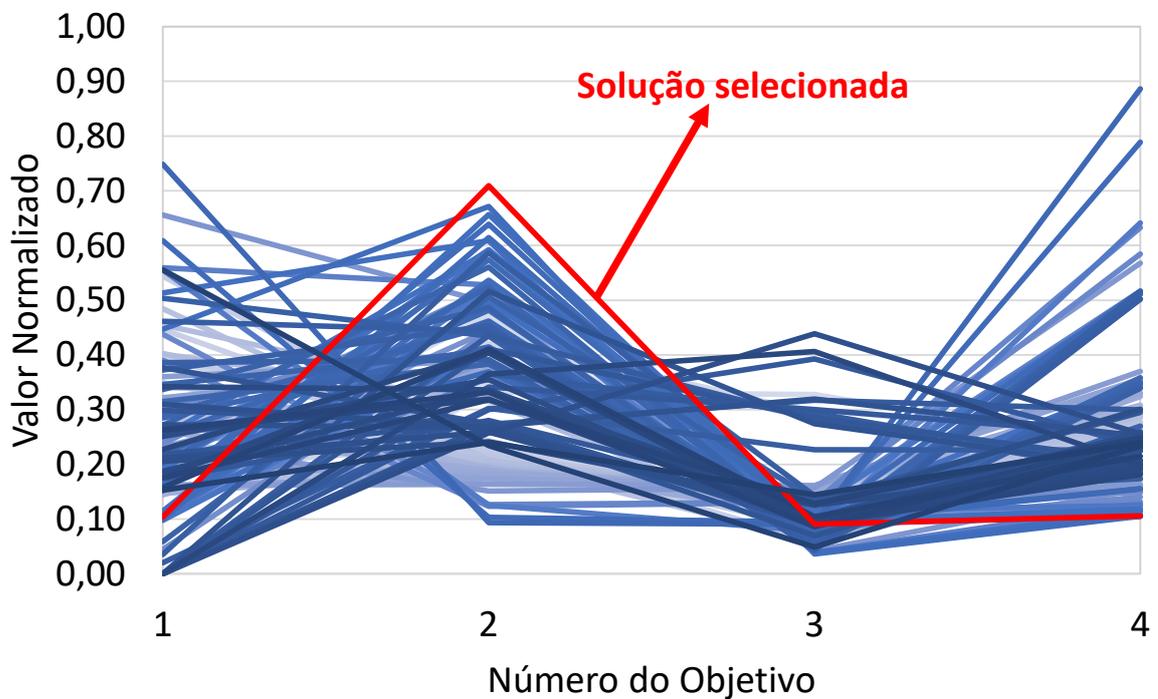
normalizado do consumo de energia. Finalmente, o objetivo 4 é o valor normalizado da expressão $1/(1 + \lambda_{N-1})$. Pela figura, pode-se observar que ambos os algoritmos apresentam soluções com valores pequenos para todos os quatro objetivos. Assim, há uma convergência adequada para ambas as abordagens. Entretanto, os operadores genéticos usados no NSGA-III promovem resultados finais para redes de baixo custo (objetivos 2 e 3 nas figuras), e não é possível obter redes com altos valores para a conectividade algébrica (objetivo 4). Por outro lado, o MaOPSO obteve redes muito caras, mas estas redes mostram um alto nível de robustez (altos valores para a conectividade algébrica) (*trade-off* entre os objetivos 2 e 4). Como consequência, MaOPSO e NSGA-III apresentaram compromissos diferentes neste problema.

Afim de mostrar o processo de projeto de uma rede óptica a partir das soluções obtidas, selecionou-se uma solução do conjunto aproximação de cada algoritmo, com base em uma escolha subjetiva pautada pela experiência de um especialista em projeto de rede. As soluções selecionadas para projeto para o NSGA-III e para o MaOPSO estão referenciadas na Figura 70a e na Figura 70b, respectivamente. A solução selecionada para o NSGA-III apresenta $PB = 2,81 \cdot 10^{-2}$, $CAPEX = 28300$ u.m., consumo de energia = 323 kW e $\lambda_{N-1} = 3,83$. Por outro lado, a solução para o MaOPSO apresenta $PB = 1,04 \cdot 10^{-1}$, $CAPEX = 66000$ u.m., consumo de energia = 163 kW e $\lambda_{N-1} = 8,45$.

Para permitir uma análise mais visual dessas soluções para os tomadores de decisão, mapas geográficos das soluções selecionadas foram criados e são apresentados na Figura 71. A rede apresentada na Figura 71b é 135% mais cara que a rede apresentada na Figura 71a. Por outro lado, a rede na Figura 71b é mais robusta que a rede produzida pelo NSGA-III. Este aspecto fica claro quando observa-se os mapas geográficos e conta-se o número de enlaces adicionais apresentados na rede retratada na Figura 71b. A primeira rede tem 42 enlaces e a segunda rede tem 75 enlaces. Assim, apesar da solução do MaOPSO ser

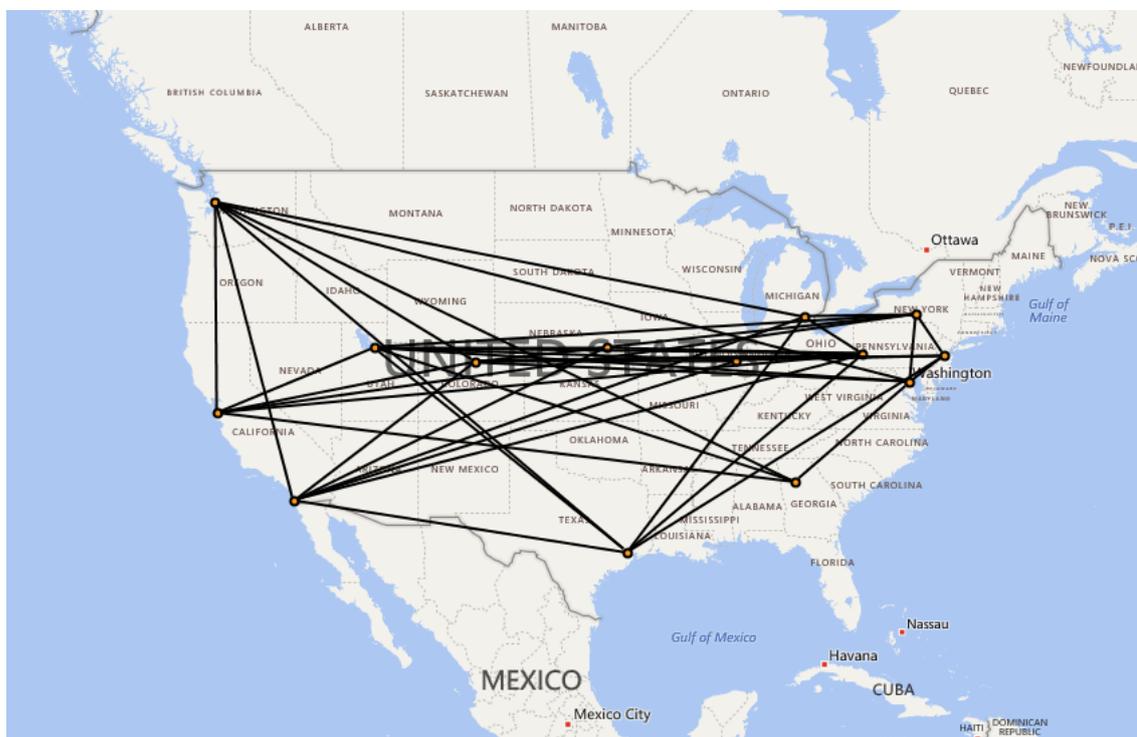


(a) Conjunto aproximação do NSGA-III e solução selecionada para projeto.

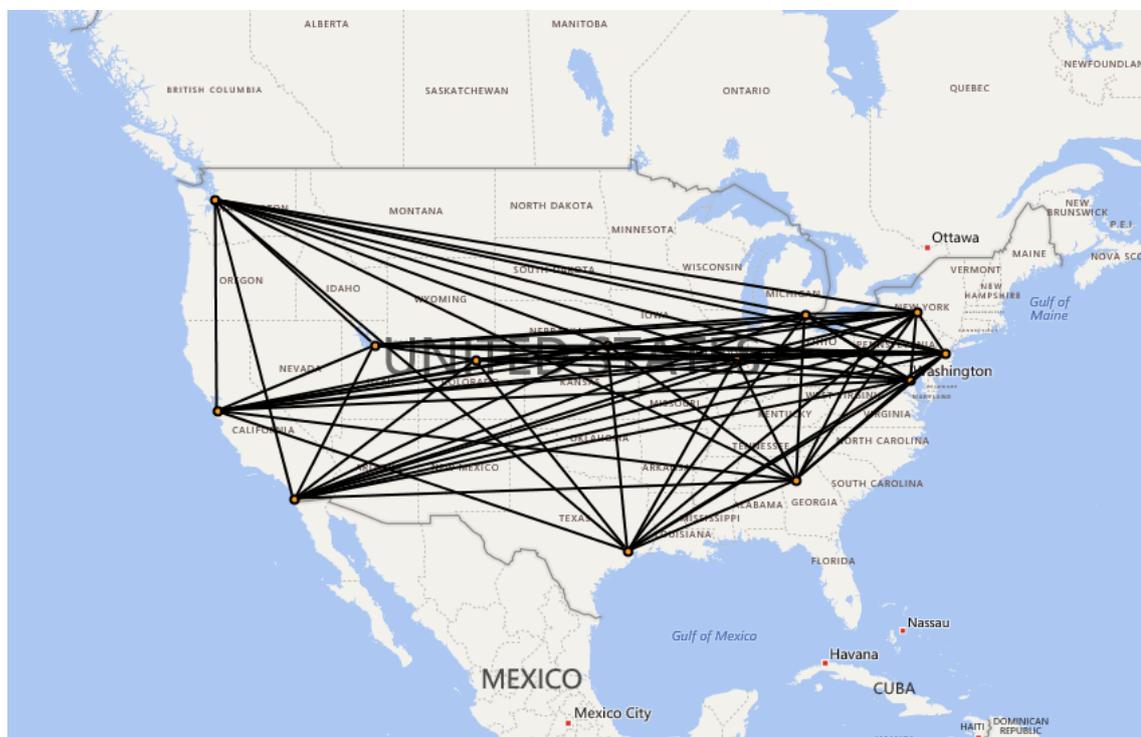


(b) Conjunto aproximação do MaOPSO e solução selecionada para projeto.

Figura 70 – Conjunto aproximação do MaOPSO e do NSGA-III depois de 5000 iterações.



(a) Solução escolhida pelo NSGA-III



(b) Solução escolhida pelo MaOPSO

Figura 71 – Topologia física das soluções destacadas na Figura 70.

mais cara, ela é mais robusta, mostrado assim um outro tipo de compromisso diferente da solução do NSGA-III. É importante aqui enfatizar o conflito entre robustez e custo. É claro que uma rede totalmente conectada tem a maior robustez, mas o custo de implementação dessa rede a torna ineficaz em termos práticos. Neste sentido, o MaOPSO oferece um compromisso importante entre robustez e custo de implementação.

Por fim, é importante destacar que o espalhamento do conjunto aproximação do MaOPSO é mais apropriado considerando todos os objetivos. Isto é, as soluções do MaOPSO atingem uma maior intervalo de valores para cada objetivo, diferente do que ocorre com o NSGA-III. Além disso, apesar de algumas soluções do MaOPSO serem mais caras, elas podem ajudar os tomadores de decisão a executar ajustes manuais nessas soluções de topologias físicas durante o processo de projeto da rede.

4.4 Considerações Finais

Como conclusão para a análise realizada neste capítulo, pode-se dizer que o MaOPSO apresentou resultados promissores e melhores comparado aos algoritmos avaliados neste trabalho em termos de convergência, diversidade e cobertura da Frente de Pareto para problemas de difícil convergência (DTLZ1 e DTLZ3) e para problemas que impõe dificuldades em termos de diversidade (DTLZ4). No entanto, o MaOPSO apresentou dificuldades em obter uma cobertura de toda Frente de Pareto nos problemas DTLZ5 e DTLZ6. Além disso, o MaOPSO foi aplicado de forma bem sucedida em um problema do mundo real envolvendo o projeto de redes ópticas de alta capacidade.

5 Conclusão e Trabalhos Futuros

Este capítulo apresenta as conclusões desta tese juntamente com as considerações finais do autor. Além disso, este capítulo também apresenta os muitos trabalhos futuros que podem se abrir a partir do desenvolvimento desta tese, e dos resultados alcançados. Finalmente, este capítulo se encerra apresentando as publicações alcançadas nesta tese. Sendo assim, este capítulo é organizado como segue. A seção 5.1 apresenta as conclusões da tese e as considerações finais. A seção 5.2 apresenta levantamento de trabalhos futuros que foram obtidos ao longo do desenvolvimento desta tese, seja no sentido de melhorias seja no sentido de inovações. Finalmente, a seção 5.3 são levantadas as publicações alcançadas durante o desenrolar desta tese.

5.1 Considerações Finais e Conclusão

Problemas multiobjetivos são problemas em que duas ou mais funções objetivos estão envolvidas e em que esses objetivos estão em conflito entre si. Um classe especial de problemas multiobjetivos são os problemas com quatro ou mais objetivos. Esses problemas aparecem com bastante frequência em diversos problemas práticos do mundo real, daí a grande importância do estudo desses problemas. Algoritmos Evolucionários são uma maneira poderosa de resolver esses problemas, pois eles não precisam de nenhuma informação a respeito do problema para funcionar. De fato, Algoritmos Evolucionários Multiobjetivos ou *Multiobjective Evolutionary Algorithms* (MOEAs) obtiveram muito sucesso na solução de problemas multiobjetivos. Contudo, MOEAs apresentam severas dificuldades quando lidando com problemas com muitos objetivos. A razão principal disto está no fato de que, como a maioria desses algoritmos se apoia no conceitos de dominância para conduzir a busca, esses algoritmos acabam tendo sua pressão de seleção enfraquecida pois à medida que o número de objetivos aumenta fica mais difícil uma solução dominar a outra e, desta forma, todas as soluções tendem a se tornarem incomparáveis entre si. Consequentemente, por causa de ausência de discriminação entre as soluções, esses algoritmos tendem a se comportar de modo aleatório em cenários com muitos objetivos.

A incapacidade de MOEAs em convergir em problemas com muitos objetivos merece toda a atenção pois esses problemas são largamente encontrados no mundo real (ZHANG; TIAN; JIN, 2015). O que faz com que o desenvolvimento de algoritmos evolucionários, ou inteligência de enxames ou qualquer outro meta-heurística para resolver um tópico de pesquisa de grande importância.

Tendo em vista a grande relevância do tema da otimização de muitos objetivos, esta tese apresentou um *Multiobjective Particle Swarm Optimization* (MOPSO) baseado

em um conjunto de pontos de referência chamado MaOPSO para resolver *Many-Objective Optimization Problems* (MaOPs). O MaOPSO usa o conceito de pontos de referência uniformemente distribuídos sobre um hiperplano para o cálculo de duas medidas, a saber, a *medida de convergência* e a *medida de densidade*, em seu método de atribuição de aptidão para impor a pressão de seleção necessária para o algoritmo convergir para a Frente de Pareto. A *medida de convergência* fornece uma estimativa da convergência do arquivo externo do MaOPSO para a Frente de Pareto, enquanto a *medida de densidade* fornece uma estimativa da concentração das soluções ao longo da Frente de Pareto.

O MaOPSO tem a capacidade de produzir um conjunto aproximação bem distribuído e próximo da Frente de Pareto como demonstrado nos experimentos conduzidos nesta tese. Nesses experimentos, ele foi comparado em métricas de avaliação de convergência (Distância Geracional) e diversidade (Distância Geracional Invertida) com um MOPSO baseado em dominância de Pareto (SMPSO) e com outros algoritmos propostos para resolver MaOPs, tais como CEGA, MDFA, CDAS-SMPSO e NSGA-III. De acordo com os resultados empíricos, o MaOPSO mostrou um alto desempenho comparado com as outras abordagens e foi o único algoritmo que obteve simultaneamente uma boa convergência e uma boa diversidade (o melhor compromisso entre convergência e diversidade). Desta maneira, MaOPSO é um bom candidato a ser usado para resolver problemas com muitos objetivos. A subseção a seguir apresenta e comenta as principais contribuições desta tese.

5.1.1 Contribuições da Tese

As principais contribuições desta tese são enumeradas a seguir:

1. O MaOPSO é, até onde vai o conhecimento do autor, o primeiro algoritmo baseado em enxame de partículas que conseguiu conciliar convergência e diversidade segundo os resultados obtidos nos experimentos. De fato, a técnica de enxame de partículas tem se destacado e recebido uma maior atenção que outras técnicas de Inteligência de Enxames (ZHANG; WANG; JI, 2015), tais como *Bacterial Foraging Optimization*, *Ant Colony Optimization*, *Artificial Bee Colony*, *Glowworm Swarm Optimization*, e *Bat Algorithm*. Portanto, a proposição em si deste algoritmo é considerada, na opinião do autor, uma contribuição importante para a área de otimização de muitos objetivos. É importante destacar que o MaOPSO é um algoritmo de enxame de partículas puro em que nenhuma hibridização de nenhum tipo nem com algoritmos evolucionários foi realizada. Isso permite que o algoritmo permaneça conceitualmente simples.
2. O MaOPSO, ao que tudo indica, é o primeiro algoritmo baseado em enxame de partículas que utiliza um conjunto de pontos de referência para guiar o processo de busca em problemas com muitos objetivos. De fato, após a publicação do NSGA-III,

muitos algoritmos evolucionários surgiram utilizando também pontos de referência como o NSGA-III mas nenhuma proposição com enxame de partículas foi sugerida.

3. Proposição de duas medidas no processo de atribuição de aptidão: a *medida de convergência* e a *medida de densidade*. Essas medidas são avaliadas de modo hierárquico do seguinte modo. Primeiramente, se avalia a *medida de densidade* e em caso de empate se avalia a *medida de convergência*. Como se pode facilmente concluir, a medida de convergência é utilizada para estimar a convergência das soluções do arquivo externo para a Frente de Pareto, enquanto a medida de densidade é utilizada para avaliar a concentração de soluções em determinadas regiões da Frente de Pareto.
4. Proposição de um método de seleção por torneio adaptativa em que o número de participantes no torneio varia de acordo com o número de objetivos no problema.

A próxima subseção deste capítulo apresenta as principais desvantagens do MaOPSO e as limitações do trabalho desenvolvido neste doutorado como um todo.

5.1.2 Limitações do Trabalho

Como já foi mencionado, o MaOPSO é, até onde vai o conhecimento do autor, o primeiro algoritmo baseado em enxame de partículas que conseguiu conciliar convergência e diversidade segundo os resultados obtidos nos experimentos. E, portanto, a proposição em si deste algoritmo é considerada, na opinião do autor, uma contribuição importante para a área de otimização de muitos objetivos. Contudo, o MaOPSO ainda apresenta algumas desvantagens e limitações que precisam ser tratadas. Esta seção apresenta as principais desvantagens e limitações não só no MaOPSO, mas também do arranjo experimental e método científico aplicado durante o desenvolvimento deste doutorado. Como já foi mencionado oportunamente, a área de otimização de muitos objetivos por meta-heurísticas (ISHIBUCHI; TSUKAMOTO; NOJIMA, 2008; MISHRA; DEHURI; CHO, 2015) ainda é muito recente e está na sua infância e provavelmente irá precisar de mais algumas décadas para atingir a sua maturidade. Com efeito, o primeiro survey sobre otimização de muitos objetivos surgiu em 2008, e foi um *survey* muito curto apontando os principais desenvolvimentos da área até aquele momento. Após esse *survey* surgiu aproximadamente dez anos depois um outro *survey* publicado em 2015 em que os desenvolvimentos e realizações da área foram explorados e onde os autores propuseram uma classificação afim de organizar as várias técnicas para resolver MaOPs. Como área de otimização de muitos objetivos ainda está amadurecimento e como muitos dos desenvolvimentos supostamente mais promissores da área se iniciaram durante o desenvolvimento deste doutorado, o autor desta tese foi obrigado a tomar várias decisões afim de continuar o trabalho da melhor maneira dentro do possível ao mesmo tempo em que era obrigado a respeitar as restrições de tempo impostas pelo doutorado. De fato, o autor foi obrigado a enfrentar vários desafios para a realização deste doutorado a

começar pelo próprio tema em si que é bastante desafiador, e depois teve enfrentar desafios de ordem mais práticas como simulações demoradas dos algoritmos, ausência de código disponível dos principais algoritmos tais como o CEGA, MDFA e NSGA-III entre outros.

Diante de tudo isso que foi exposto, esta seção é o momento oportuno para o autor comentar as principais limitações deste trabalho e também as suas deficiências quanto a aplicação do método científico que apesar de não comprometer de modo algum este trabalho, é um limitante no sentido se retirar conclusões mais gerais e robustas a respeito da técnica proposta, o MaOPSO. Desta forma, as principais críticas, desvantagens, limitações, deficiências deste trabalho são exploradas em detalhes a seguir.

Como mencionado na subseção 3.2.1.1, o *Many-Objective Particle Swarm Optimization* (MaOPSO) utiliza uma estratégia de geração de pontos de referência que distribui os pontos de modo uniforme em torno de um hiperplano. Esse método de geração de pontos de modo estruturado sobre um hiperplano foi proposto por Das e Dennis em (DAS; DENNIS, 1998). Para um número pequeno de objetivos e para um número de divisões p pequeno, a quantidade de pontos de referência produzido por este método é gerenciável, isto é, é próximo de um número de pontos normalmente utilizados em MOEAs e MOPSOs (DAS; DENNIS, 1998; Coello Coello; LAMONT; VELDHUIZEN, 2007). Contudo, em problemas com muitos objetivos, até mesmo um número pequeno de p pode produzir um número enorme de pontos de referência. Por exemplo, para um problema com oito objetivos, 5040 pontos de referência são criados usando o método de Das e Dennis, apenas um desses pontos é um ponto intermediário (DEB; JAIN, 2014). Tudo isso decorre porque o algoritmo de geração de pontos de referência proposto por Das e Dennis é algoritmo em que o número de pontos de referência criados cresce fatorialmente com o número de objetivos e com o número de divisões p .

O crescimento fatorial do número de pontos de referência pelo método de Das e Dennis faz com que os algoritmos que utilizem este método como é o caso do MaOPSO e do NSGA-III necessitem estabelecer um compromisso entre o desempenho do algoritmo, que é alcançado geralmente com o aumento do número de pontos de referência, e o custo computacional, que pode ser tornar a aplicação dos algoritmos inviável devido a grande quantidade de pontos de referência criada.

O algoritmo NSGA-III em sua proposição original conhecida como MO-NSGA-II (DEB; JAIN, 2012) estabelece que o tamanho da população seja exatamente igual ao número de pontos de referência. A mesma recomendação também aplicada no próprio NSGA-III (DEB; JAIN, 2014). De fato, nesta tese utilizou-se um tamanho de população diferente do NSGA-III, e isso ocasionou um desempenho abaixo do esperado no NSGA-III tanto em termos de convergência como em termos de diversidade. Disso pode decorrer dois pontos conflitantes. Primeiro, para certos problemas pode ser necessário utilizar uma grande quantidade de pontos de referência para impor uma convergência para o algoritmo.

E o segundo ponto é que a utilização de uma grande quantidade de pontos de referência, por sua vez, faz com que o NSGA-III tenha que usar uma população de indivíduos muito grande o que poderia exigir um número enorme de avaliações de aptidão, o que tornaria, conseqüentemente, a aplicação prática do NSGA-III em alguns casos inviável.

Nesse sentido, o MaOPSO tem uma vantagem sobre o NSGA-III que merece ser destacada. Como é intuitivo esperar, quanto maior o número de pontos de referência, maior o número de atratores para a Frente de Pareto. Sendo assim, é importante que o número de pontos de referência seja diferente do tamanho da população. Com efeito, o melhor caso é lidar com populações razoavelmente pequenas para evitar um número excessivo de avaliações de aptidão, e com um número razoavelmente grande de pontos de referência que permita uma maior atração para a Frente de Pareto mas que, por outro lado, não comprometa o desempenho do algoritmo em termos de tempo de processamento. Desta forma, diferentemente do que ocorre com o NSGA-III, em que o tamanho da população e número de pontos de referência devem ser iguais para garantir o bom desempenho do algoritmo, o MaOPSO, pelo menos em princípio, não exige que isso aconteça para obter um bom desempenho. Ou seja, ao que tudo indica, o tamanho da população, o tamanho do arquivo externo, e o número de pontos de referência são desacoplados, de modo que eles não precisam ser necessariamente iguais para garantir o bom funcionamento do MaOPSO. Para se verificar isso, basta ver o caso da execução do MaOPSO em problemas com 10 objetivos em que 300 partículas foram usadas em conjunto com um arquivo externo de tamanho igual a 500 usando 2002 pontos de referência. E mesmo assim o MaOPSO conseguiu distribuir bem os pontos pela análise do IGD e do HV mas sobretudo pela análise dos conjuntos aproximação do MaOPSO em coordenadas paralelas. O mesmo não ocorreu com o NSGA-III. Sendo assim, o MaOPSO tem maiores possibilidades de ser aplicado em problemas práticos sendo capaz de possivelmente convergir e produzir Frontes de Pareto diversas nesses cenários.

Por esta razão, ou seja, devido ao alto custo computacional que pode ser proveniente de se utilizar uma população muito grande em decorrência de um número grande de pontos de referência, Deb e Jain propuseram uma pequena modificação no processo de geração de pontos de referência. Considerando o problemático dilema entre custo computacional e desempenho/eficácia do algoritmo, Deb e Jain em (DEB; JAIN, 2014) propuseram uma estratégia para criar os pontos de referência por camadas. Uma *camada externa* em que pontos de referência são criados sobre as bordas do hiperplano, e uma *camada interna*, em que pontos a redor do ponto central do hiperplano são criadas. Essa estratégia foi motivada pelo aumento fatorial do número de pontos de referência com respeito ao número de objetivos do problema e do número de divisões p . Com efeito, para $p \geq m$ nenhum ponto intermediário é criado no hiperplano (DEB; JAIN, 2014). Sendo assim, enquanto a camada externa tem o papel de cobrir as bordas do hiperplano, a camada interna tem o papel de cobrir a região central do hiperplano. Tendo isso em vista, a geração de pontos de

referência por camadas pode ser explicada como segue tomando de novo o exemplo anterior de oito objetivos. Sendo assim, esse método primeiramente cria uma camada extrema usando um valor pequeno de p , por exemplo, $p = 3$ o que produz usando o método de Das e Dennis um total de pontos de referência igual a 120. Na camada interna, pode-se utilizar um valor menor ainda, por exemplo, $p = 2$ o que produz um total de pontos de referência na camada interna igual 36. E, com isso, o número total de pontos de referência é igual a 156. Como se pode perceber, essa estratégia tem a vantagem de reduzir o número de pontos de referência, mas têm a desvantagem de deixar partes da Frente Pareto descoberta. Apesar disso, o ganho obtido pela redução de pontos de referência estrategicamente selecionados, pode ser superior ao ganho obtido pelo custo computacional de lidar com um enorme quantidade de pontos de referência. Desta forma, o uso de uma grande quantidade de pontos de referência pode ser considerada uma limitação do MaOPSO, e a utilização da construção de pontos de referência por camada pode ser um maneira promissora de contornar essa limitação.

Outra desvantagem do MaOPSO é que ele tem problemas em lidar com problemas degenerados. Isto porque as soluções extremas (SINGH; ISAACS; RAY, 2011) em problemas degenerados não estão sobre os eixos objetivos. Isso, por sua vez, dificulta a detecção desses pontos pelo MaOPSO e a consequente determinação do hiperplano. Outra desvantagem do MaOPSO é que ele tem problemas em lidar com Frentes de Pareto desconexas, como, por exemplo, a função de teste DTLZ7 da família DTLZ (DEB et al., 2005).

Além das desvantagens já mencionadas e dos pontos de melhorias do MaOPSO que são oportunamente descritas na seção 5.2, outro quesito que pode ser um forte alvo de críticas é o definição dos parâmetros e das constantes usadas no MaOPSO. Além disso, outro ponto que pode ser fortemente criticado é o arranjo experimental desta tese. Contudo, todas essas críticas, apesar de válidas e muito pertinentes, podem justificadas pelo próprio amaduramento da área que ainda está em seu início. De fato, a área de otimização de muitos objetivos está na sua infância (LI et al., 2015b) e que, portanto, não existe uma padronização de um arranjo experimental que permita comparar adequadamente os algoritmos. Além disso, devido as restrições de tempo, não foi possível realizar experimentos mais cuidadosos e estudos mais profundos e elaborados a respeito do MaOPSO. Isso torna difícil e tedioso e muito propenso a erro o processo de comparação dos algoritmos, mas também o próprio processo de reprodução dos algoritmos. As métricas de avaliação, podem ser implementadas de várias maneiras, isto é o caso, por exemplo do IGD e do hipervolume. Com efeito, apenas recentemente a proposição de linhas diretivas para o arranjo experimental (CHENG et al., 2017). Nesse trabalho, Cheng *et al* sugeriu usar 31 execuções independente de cada algoritmo como utilizado no arranjo experimental desta tese; três valores de número de objetivos no conjunto $\{5, 10, 15\}$. E um tamanho de população que varia de acordo com o número de objetivos como feito nesta tese, mas usando um valor máximo de $25 \times m$ em que m é o número de objetivos. Esse mesmo esquema foi utilizado para definir o

tamanho do arquivo externo. De fato, fazer com que todos os indivíduos tenham o mesmo tamanho na população final e no arquivo externo final tem a vantagem de permitir uma comparação justa dos resultados (CHENG et al., 2017). Finalmente, Cheng *et al* também propuseram em seu arranjo experimental usar um número de avaliações de aptidão que varia dependendo do número de objetivos, mas que, contudo, não deveria ultrapassar o número máximo de avaliações de função (CHENG et al., 2017).

O arranjo experimental também pode ser criticado no sentido da definição do tamanho de população dos algoritmos. Isso é particularmente verdade na definição do tamanho da população no NSGA-III em que a recomendação de tamanho de população recomendada (DEB; JAIN, 2014) não foi seguida. A razão porque não se utilizou os valores recomendados de tamanho de população é porque isso poderia impedir a execução dos experimentos em tempo hábil. Contudo, a principal razão foi que a definição do tamanho da população tinha que ser de tal maneira a garantir a aplicabilidade dos algoritmos em problemas práticos do mundo real. Essa foi a razão principal porque se utilizou tamanhos pequenos de população com relação ao número de pontos de referência utilizados pois a ideia era manter o MaOPSO aplicável em cenários práticos onde em muitas situações a avaliação da função objetivo é muito custosa em termos de tempo de processamento. Portanto, esse fato justificou uma escolha diferente dos valores de população e dos pontos de referência do NSGA-III, uma vez que em cenários práticos o que se deseja é utilizar uma população relativamente pequena junto com uma quantidade de pontos de referência que permita uma maior pressão de convergência e uma maior cobertura da Frente de Pareto além de um melhor espalhamento das soluções ao longo da Frente de Pareto. Tal exigência não poderia ocorrer se a recomendação de Deb e Jain para o NSGA-III fosse seguida pois no NSGA-III o tamanho da população deveria ser igual ao número de pontos de referência, e o número de pontos de referência utilizados foi estabelecido de modo a garantir a convergência do MaOPSO nos diversos problemas de teste validados no arranjo experimental. Contudo, na análise comparativa de desempenho de algoritmos, essa decisão de projeto do arranjo experimental como explicada pode ser alvo de críticas no sentido que ele não favorece uma comparação justa com o NSGA-III uma vez que não se seguiu os valores recomendados de tamanho de população. Desta forma, o arranjo experimental utilizado neste trabalho pode ser uma possível limitação na avaliação comparativa dos algoritmos, mas isto está longe de comprometer o bom desempenho e a aplicabilidade do MaOPSO. Sendo assim, um comparativo do MaOPSO com o NSGA-III, executando o NSGA-III com um tamanho de população igual ao número de pontos de referência como recomendado é algo que precisa ser realizado no futuro. Neste caso, tanto o NSGA-III como o MaOPSO devem utilizar a abordagem de geração de pontos de referência por camadas para garantir um número gerenciável de pontos de referência, isto é, um número geralmente menor que 300 (DEB; JAIN, 2014).

Outro ponto do arranjo que também pode ser alvo de críticas é o tamanho do

arquivo externo utilizado no CDAS-SMPSO e no SMPSO que tinha um tamanho de arquivo externo constante $A_{max} = 100$. Optou-se por usar esses valores, porque esse valores já tinham sido testados pelos próprios autores dos algoritmos e uma mudança nesse valor poderia influenciar o desempenho dos algoritmos. Contudo, uma crítica que se poderia levantar é que isso poderia afetar o cálculo do GD que depende do número de pontos no conjunto aproximação do algoritmo P e também o HV, mas por outro lado a métrica IGD não é afetada por isso. Pode-se argumentar que utilizar conjuntos aproximações com tamanhos diferentes, como foi feito nesta tese por razões já esclarecidas neste capítulo e também no Capítulo 4 relacionadas a restrições de tempo e a aplicabilidade dos algoritmos, poderia levar a uma comparação injusta entre os algoritmos. Isso é verdade para as métricas GD e HV que dependem do tamanho do conjunto aproximação P gerados pelos algoritmos. Contudo o mesmo não ocorre para a métrica IGD. A métrica IGD como se pode perceber verificando a sua fórmula de cálculo, depende no tamanho do conjunto de pontos de referência P^* utilizados. Entretanto, o MaOPSO mesmo usando um conjunto aproximação de 500 soluções contra um conjunto de pontos de referência de 2002 pontos para cálculo do IGD em 10 objetivos foi capaz de produzir menores valores de IGD, superando algoritmos como o MDFA e o CEGA que possuíam um conjunto aproximação com no máximo 300 e 100 soluções, respectivamente, uma vez que utilizou-se apenas soluções não-dominadas no cálculo do IGD.

Finalmente, o MaOPSO foi aplicado em um problema prático de engenharia que envolveu o projeto de redes ópticas de alta capacidade produzindo soluções de projeto de alta robustez. E, com isso, ele obteve soluções em uma região diferente das soluções produzidas pelo seu principal concorrente, o NSGA-III.

A partir de tudo que foi exposto acima, tanto a respeito das possíveis a serem feitas no MaOPSO como na questão do arranjo experimental, é possível traçar várias linhas de trabalhos futuros que são explicadas na seção 5.2.

5.2 Trabalhos Futuros

A partir da análise das contribuições deste doutorado mas sobretudo a partir das críticas advindas do arranjo experimental e das desvantagens e limitações do MaOPSO em problemas com cenários mais complexos que os problemas aqui tratados, pode-se propor a seguinte lista de trabalhos futuros:

- Desenvolver uma versão mais robusta do MaOPSO em que um número menor de pontos de referência é usado ao mesmo tempo em que esses pontos possam fornecer uma maior pressão de convergência para a Frente de Pareto. Para isso pode-se utilizar a estratégia de geração de pontos de referência em duas camadas: uma camada externa (de fronteira) e outra camada interna (que inclui um ponto

intermediário sobre o hiperplano). Essa estratégia foi proposta por Deb e Jain (DEB; JAIN, 2014) e tem sido utilizada em outros trabalhos como em (WANG; JIAO; YAO, 2015), por exemplo.

- Avaliar novas estratégias de determinação dos líderes do sub-enxame \bar{S}_1 usando por exemplo método *Corner Sort* (WANG; YAO, 2014).
- Avaliar a convergência ao longo das iterações do NSGA-III e do MaOPSO para verificar qual deles converge primeiro e por quê. Neste caso, a ideia é fazer um estudo similar ao realizado na subseção 4.2.1 mas com todos os algoritmos sendo comparados afim de estabelecer a velocidade de convergência dos algoritmos em cenários tanto com poucos como com muitos objetivos.
- Realizar experimentos em que todos os indivíduos irão utilizar a mesma quantidade de avaliações de aptidão. Nesta tese o CEGA foi o único algoritmo que não foi executado usando o mesmo esforço computacional em termos da função objetivo com relação aos demais algoritmos. Sendo assim, executá-lo usando a mesma quantidade de avaliações de aptidão que os demais algoritmos e compará-lo com o MaOPSO é deixado como trabalho futuro. Contudo, é muito importante argumentar que como o CEGA utiliza uma abordagem baseada em agregação, existe uma grande possibilidade de toda a população convergir para um ponto do espaço objetivo e de lá permanecer devido ao seu método de aptidão agregado (GARZA-FABRE; PULIDO; Coello Coello, 2010; Garza Fabre; PULIDO; Coello Coello, 2009; Garza Fabre, 2009).
- Desenvolver, para o MaOPSO, um método mais sofisticado de agrupamento das soluções do arquivo externo em torno dos pontos de referência e com isso promover uma melhor estimativa da densidade. Isso pode ser feito utilizando o método de intersecção de fronteira ou *Normal Boundary Intersection* (NBI) em inglês, em que duas distâncias são calculadas (DAS; DENNIS, 1998; LI et al., 2015b; CAI et al., 2015). Uma delas consiste na distância da solução para um ponto de referência especificado pelo usuário ao longo de um vetor de peso. Se o ponto de referência for definido como o ponto ideal, essa distância pode ser utilizada como uma medida de convergência. A outra distância, por sua vez, consiste na distância perpendicular da solução para o vetor de peso. Essa distância pode ser utilizada para medir a proximidade da solução a um dado vetor de referência e assim realizar o agrupamento das soluções sem a necessidade de projetar as soluções no hiperplano. Para que esse método funcione, é necessário que um procedimento de normalização das soluções do arquivo externo seja utilizado. O procedimento de normalização é importante porque atualmente o hiperplano do MaOPSO pode ter lados com tamanhos diferentes o que pode enviesar o processo de busca para uma determinada região do espaço objetivo. Desta forma, a normalização é um procedimento importante e que deve ser introduzido no MaOPSO.

Alguns exemplos desse procedimento de normalização estão presentes em (DEB; JAIN, 2014; LI et al., 2015b; CHENG; YEN; ZHANG, 2015; ASAFUDDOULA; RAY; SARKER, 2015).

- É importante mencionar que Deb e Jain avaliaram também o uso de pontos de referência gerados de modo aleatório em seus trabalhos com o NSGA-III (DEB; JAIN, 2014; JAIN; DEB, 2014) de modo que a verificação do impacto dessa forma de geração de pontos de referência sobre o desempenho do MaOPSO é deixado como trabalho futuro. Além dessa abordagem de geração aleatória de pontos de referência, outras potenciais abordagens podem ser utilizadas no MaOPSO, de maneira que a avaliação do impacto dessas abordagens sobre o desempenho do MaOPSO é deixada também como trabalho futuro. Pode-se, por exemplo, criar um esquema em que os pontos de referência são criados de modo adaptativo ao longo das iterações como nos trabalhos de Yan-yan Tan *et al.* (TAN et al., 2013), ou como nos trabalhos de Rui Wang e colaboradores (WANG; PURSHOUSE; FLEMING, 2013c; WANG; PURSHOUSE; FLEMING, 2013a; WANG; ZHANG; GUO, 2013).
- Outra linha de pesquisa relacionada à anterior é a questão de como gerar um número pequeno de pontos de referência mantendo a estrutura uniforme que se espera que esses pontos possuam para que possam conduzir o processo de busca do MaOPSO. Como mencionado na subseção 3.2.1.1, o MaOPSO utiliza uma estratégia de geração de pontos de referência que são distribuídos de modo uniforme em torno de um hiperplano. Para um número pequeno de pontos de referência e para um número de divisões p pequeno, a quantidade de pontos de referência produzidos pelo método é gerenciável, isto é, é próximo de um número de pontos normalmente utilizado em MOEAs e MOPSOs. Contudo, para problemas com muitos objetivos, até mesmo um número pequeno de p pode produzir um número enorme de pontos de referência. Considerando tudo isso, Deb e Jain em (DEB; JAIN, 2014) propuseram criar esses pontos por camadas. Uma *camada externa* em que pontos de referência são criados sobre as bordas do hiperplano, e uma *camada interna*, em que pontos a redor do ponto central do hiperplano são criados. Esse método então permitiu criar pontos que estão uniformemente espaçados sobre as bordas do hiperplano e também na região central deste mesmo hiperplano. Desta forma, um trabalho futuro é aplicar esta geração de pontos de referência por camadas no MaOPSO.
- Avaliar o efeito do uso de dois arquivos externos no desempenho do MaOPSO, sendo que um deles é utilizado para promover a convergência enquanto o outro é utilizado para promover a diversidade. Nesse caso, ao invés de se utilizar um líder social e um líder cognitivo, esta nova proposta utiliza dois líderes sociais provenientes de cada um desses arquivos. Essa ideia é inspirada na proposta original de Wang *et al.* (WANG; JIAO; YAO, 2015)

- Investigar exaustivamente o desempenho do MaOPSO para uma maior variedade de problemas. Mais especificamente, pretende-se investigar o desempenho do MaOPSO sobre os problemas da família WFG (HUBAND et al., 2006), a saber, os problemas WFG1 até o problema WFG9, e sobre um conjunto de problemas de teste proposto em 16 de janeiro de 2017 para a competição do CEC 2017 envolvendo a otimização de problemas com muitos objetivos (CHENG et al., 2017). Além desses problemas, outros podem ser utilizados, tais como os problemas propostos por Ishibuchi *et al.* (ISHIBUCHI; MASUDA; NOJIMA, 2016), os problemas de teste escaláveis propostos por Miqing Li (LI; YANG; LIU, 2014b; LI et al., 2017) e os problemas LSMOPs propostos por Cheng (CHENG et al., 2016b).
- Fazer um estudo paramétrico para entender a influência e interferência de cada parâmetro e constante sobre o desempenho do MaOPSO, tais como: (1) o tamanho da população N ; (2) o número de divisões p e, portanto, o número de pontos de referência utilizados no hiperplano; (3) o tamanho máximo do arquivo externo $|A|_{max}$; (4) os parâmetros c_1 , c_2 , ω na equação de velocidade; (5) o termo ou fator de divisão do enxame S em dois sub-enxames \bar{S}_1 e \bar{S}_2 , que nesta tese foi definido empiricamente como uma constante igual $1/3$ e que em termos gerais pode ser considerado como um parâmetro ζ especificado pelo usuário ($\zeta = \frac{1}{3}$ atualmente); (6) a proporção de partículas que são afetadas pelo operador de mutação, que nesta tese foi definido como uma constante de 15% e que em termos gerais pode ser considerado como um parâmetro π especificado pelo usuário ($\pi = 15\%$ atualmente); e (7) o índice de distribuição de mutação η_m (DEB; GOYAL, 1996; NEBRO et al., 2009). Todos esses parâmetros podem influir de modo substancial no comportamento do MaOPSO, e, por tanto, um estudo aprofundado do impacto de cada um desses parâmetros sobre o desempenho do MaOPSO em termos de convergência e diversidade é deixado como trabalho futuro. A ideia é que, a partir desses estudos paramétricos, possa-se definir uma versão do MaOPSO livre de parâmetros (DEB et al., 2002), ou seja, um MaOPSO com apenas os parâmetros usuais tais como o tamanho da população, o tamanho do arquivo externo, o número máximo de avaliações de aptidão (ou o número máximo de iterações).
- Durante o desenvolvimento desta tese, muitos algoritmos para otimização de muitos objetivos surgiram na literatura. A esmagadora maioria deles seguiam o paradigma evolucionário. Obviamente, devido a restrições de tempo, não foi possível acompanhar tal desenvolvimento. E, desta forma, a comparação do MaOPSO foi realizada com o que era até então o estado da arte. Portanto, um trabalho futuro muito importante é comparar o desempenho do MaOPSO com outros algoritmos mais recentes que os utilizados nesta tese e que são até o presente momento o estado da arte em otimização de muitos objetivos, tais como o *Clustering-Ranking Evolutionary*

Algorithm (crEA) (CAI et al., 2015), o *Improved Decomposition-based Evolutionary Algorithm* (I-DBEA) (ASAFUDDOULA; RAY; SARKER, 2015), o *Many-Objective Evolutionary Algorithm (MaOEA) based on Directional Diversity (DD) and Favorable Convergence (FC) (MaOEA-DDFC)* (CHENG; YEN; ZHANG, 2015), o MOEA/DD (LI et al., 2015b), o θ -DEA (YUAN et al., 2016), o *Reference Vector guided Evolutionary Algorithm* (RVEA) (CHENG et al., 2016a), o *Vector angle-based Evolutionary Algorithm* (VaEA) (XIANG et al., 2017), além do já utilizado NSGA-III (DEB; JAIN, 2014). É importante destacar que os algoritmos citados são todos algoritmos evolucionários que seguem o mesmo princípio do MaOPSO e do NSGA-III, isto é, todos esses algoritmos utilizam um conjunto de pontos de referência em sua estrutura. Contudo, existem algoritmos recentes para otimização de muitos objetivos que não seguem essa abordagem, tais como os algoritmos C-Multi-(JR.; SANTANA; POZO, 2016), GrEA (YANG et al., 2013; LI et al., 2010), *Shift-based Density Estimation* (SDE) (LI; YANG; LIU, 2014a) e *Two-Archive Algorithm* (Two_Arch) (WANG; JIAO; YAO, 2015). O algoritmo Two_Arch é particularmente interessante porque ele é baseado em uma ideia inovadora de usar dois tipos de arquivos no processo de seleção dos líderes das partículas, um arquivo para promoção de convergência e outro arquivo para geração de diversidade. O algoritmo C-Multi-(JR.; SANTANA; POZO, 2016; CASTRO; POZO, 2014) também é particularmente interessante de ser comparado com o MaOPSO porque ele é uma técnica baseada em enxame de partículas baseada em multi-enxames. Outros algoritmos relativamente recentes baseados em enxame de partículas que podem ser utilizados são o I-MOPSO (2012) (BRITTO; POZO, 2012a) e o I-Multi (2013) (BRITTO; MOSTAGHIM; POZO, 2013), que também é baseado na ideia de sub-enxames e consiste numa versão anterior do C-Multi. A diferença principal do C-Multi para o I-Multi é que C-Multi não é um algoritmo de enxames de partículas puro pois ele é hibridizado com uma técnica de estimação de distribuição para gerar soluções. Por fim, uma outra opção de comparação é o MOPSO-GD, que foi um algoritmo proposto pelo autor desta tese em sua dissertação de Mestrado (FIGUEIREDO, 2013), que utiliza uma abordagem de agregação chamada de *Global Detriment* em seu método de atribuição de aptidão. A razão pela qual o MOPSO-GD não foi utilizado como algoritmo de comparação nesta tese foi explicada no Capítulo 4.

5.3 Publicações Descorrentes do Doutorado

Os resultados obtidos nesta pesquisa de doutorado originaram as seguintes publicações:

- Many Objective Particle Swarm Optimization, artigo aceito para publicação na

revista *Information Sciences* (FIGUEIREDO; LUDERMIR; BASTOS-FILHO, 2016).

- Physical Topology Design of Optical Networks Aided by Many-Objective Optimization Algorithms, publicação aceita no *Brazilian Conference on Intelligent Systems* (BRACIS, 2016), Recife, Brazil (FIGUEIREDO et al., 2016).

Referências

- ABRAHAM, A.; JAIN, L.; GOLDBERG, R. *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. London: Springer-Verlag, 2005.
- ADRA, S. F. *Improving Convergence, Diversity and Pertinency in Multiobjective Optimisation*. Tese (Doutorado) — University of Sheffield, 2007.
- ADRA, S. F.; FLEMING, P. J. Diversity management in evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 15, n. 2, p. 183–195, April 2011.
- AGUIRRE, H.; TANAKA, K. Space partitioning with adaptive e-ranking and substitute distance assignments: A comparative study on many-objective mnk-landscapes. In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2009. p. 547–554.
- ALAM, S.; DOBBIE, G.; KOH, Y. S.; RIDDLE, P.; REHMAN, S. U. Research on particle swarm optimization based clustering: A systematic review of literature and techniques. *Swarm and Evolutionary Computation*, v. 17, p. 1–13, 2014.
- ARAÚJO, D. R. B.; BASTOS-FILHO, C. J. A.; BARBOZA, E. A.; CHAVES, D. A. R.; MARTINS-FILHO, J. F. A performance comparison of multi-objective optimization evolutionary algorithms for all-optical networks design. In: *2011 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MDCM)*. USA: IEEE, 2011. p. 89–96.
- ARAÚJO, D. R. B. de. *Planejamento de Redes Ópticas Usando Inteligência Computacional e Ciência das Redes*. Tese (Doutorado) — Universidade Federal de Pernambuco, 2015.
- ARAÚJO, D. R. de; BASTOS-FILHO, C. J.; MARTINS-FILHO, J. F. An evolutionary approach with surrogate models and network science concepts to design optical networks. *Eng. Appl. Artif. Intell.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 43, n. C, p. 67–80, ago. 2015.
- ASAFUDDOULA, M.; RAY, T.; SARKER, R. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 19, n. 3, p. 445–460, June 2015.
- AZEVEDO, C. R. B. *Geração de Diversidade na Otimização Dinâmica Multiobjetivo Evolucionária por Paisagens de Não-Dominância*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2011.
- BÄCK, T.; SCHWEFEL, H.-P. An overview of evolutionary algorithms for parameter optimization. *Evol. Comput.*, v. 1, n. 1, p. 1–23, mar. 1993.
- BADER, J.; ZITZLER, E. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.*, v. 19, n. 1, p. 45–76, 2011.
- BASTOS-FILHO, C. J. A.; aES, A. C. S. G. Multi-objective fish school search. *Int. J. Swarm. Intell. Res.*, IGI Global, Hershey, PA, USA, v. 6, n. 1, p. 23–40, 2015.

- BASTOS-FILHO, C. J. A.; ARAUJO, D. R. B.; BARBOZA, E. A.; CHAVES, D. A. R.; MARTINS-FILHO, J. F. Design of transparent optical networks considering physical impairments, capex and energy consumption. In: *2011 13th International Conference on Transparent Optical Networks*. USA: IEEE, 2011. p. 1–4.
- BATISTA, L. S.; CAMPELO, F.; GUIMARÃES, F. G.; RAMÍREZ, J. A. A comparison of dominance criteria in many-objective optimization problems. In: *2011 IEEE Congress of Evolutionary Computation (CEC)*. USA: IEEE, 2011. p. 2359–2366.
- BEUME, N.; NAUJOKS, B.; EMMERICH, M. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, v. 181, n. 3, p. 1653–1669, 2007.
- BLICKLE, T.; THIELE, L. A comparison of selection schemes used in evolutionary algorithms. *Evol. Comput.*, MIT Press, Cambridge, MA, USA, v. 4, n. 4, p. 361–394, dez. 1996.
- BONABEAU, E.; MEYER, C. Swarm intelligence: a whole new way to think about business. *Harv. Bus. Rev.*, v. 5, n. 79, p. 106–115, 2001.
- BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. *Information Sciences*, v. 237, p. 82–117, 2013.
- BOYD, S.; VANDENBERGHE, L. *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
- BRANKE; DEB; MIETTINEN; SŁOWIŃSKI (Ed.). *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin, Heidelberg: Springer-Verlag, 2008.
- BRATTON, D.; KENNEDY, J. Defining a standard for particle swarm optimization. In: *IEEE Symposium on Swarm Intelligence*. USA: IEEE, 2007. (SIS' 2007), p. 120–127.
- BRITTO, A.; MOSTAGHIM, S.; POZO, A. Iterated multi-swarm: A multi-swarm algorithm based on archiving methods. In: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2013. p. 583–590.
- BRITTO, A.; POZO, A. I-mopso: A suitable pso algorithm for many-objective optimization. In: *2012 Brazilian Symposium on Neural Networks*. USA: IEEE, 2012. p. 166–171.
- BRITTO, A.; POZO, A. Using archiving methods to control convergence and diversity for many-objective problems in particle swarm optimization. In: *2012 IEEE Congress on Evolutionary Computation*. USA: IEEE, 2012. p. 1–8.
- BROCKHOFF, D.; ZITZLER, E. Objective reduction in evolutionary multiobjective optimization: Theory and applications. *Evol. Comput.*, MIT Press, Cambridge, MA, USA, v. 17, n. 2, p. 135–166, jun. 2009.
- CAI, L.; QU, S.; YUAN, Y.; YAO, X. A clustering-ranking method for many-objective optimization. *Applied Soft Computing*, v. 35, p. 681–694, 2015. ISSN 1568-4946.
- CASTRO, O. R.; POZO, A. T. R. A hybrid competent multi-swarm approach for many-objective problems. In: *2014 Brazilian Conference on Intelligent Systems*. USA: IEEE, 2014. p. 426–431.

- CHAN, F. T. S.; TIWARI, M. K. *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*. Croatia: In-Tech, 2009.
- CHAVES, D. A. R.; BASTOS-FILHO, C. J. A.; MARTINS-FILHO, J. F. Multiobjective physical topology design of all-optical networks considering qos and capex. In: *2010 Conference on Optical Fiber Communication (OFC/NFOEC), collocated National Fiber Optic Engineers Conference*. USA: IEEE, 2010. p. 1–3.
- CHENG, J.; YEN, G. G.; ZHANG, G. A many-objective evolutionary algorithm with enhanced mating and environmental selections. *IEEE Transactions on Evolutionary Computation*, v. 19, n. 4, p. 592–605, Aug 2015.
- CHENG, R.; JIN, Y.; OLHOFFER, M.; SENDHOFF, B. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 20, n. 5, p. 773–791, Oct 2016.
- CHENG, R.; JIN, Y.; OLHOFFER, M.; SENDHOFF, B. Test problems for large-scale multiobjective and many-objective optimization. *IEEE Transactions on Cybernetics*, PP, n. 99, p. 1–14, 2016.
- CHENG, R.; LI, M.; TIAN, Y.; ZHANG, X.; YANG, S.; JIN, Y.; YAO, X. *Benchmark Functions for CEC'2017 Competition on Evolutionary Many-Objective Optimization*. United Kingdom, 2017.
- CLERC, M.; KENNEDY, J. The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 1, p. 58–73, 2002.
- COELLO, C. A. C.; LECHUGA, M. S. Mopso: A proposal for multiple objective particle swarm optimization. In: *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02*. USA: IEEE, 2002. p. 1051–1056.
- COELLO, C. A. C.; PULIDO, G. T.; LECHUGA, M. S. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, v. 8, n. 3, p. 256–279, June 2004.
- Coello Coello, C. A. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, v. 1, n. 1, p. 28–36, 2006.
- Coello Coello, C. A.; LAMONT, G. B.; VELDHUIZEN, D. A. V. *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York, USA: Springer, 2007.
- CURRY, D. M.; DAGLI, C. H. Computational complexity measures for many-objective optimization problems. *Procedia Computer Science*, v. 36, p. 185–191, 2014.
- DAS, I.; DENNIS, J. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal of Optimization*, v. 8, n. 3, p. 631–657, 1998.
- DE CARVALHO, A. B.; POZO, A. Measuring the convergence and diversity of CDAS Multi-Objective Particle Swarm Optimization Algorithms: A study of many-objective problems. *Neurocomputing*, v. 75, n. 1, p. 43–51, 2012.

- DEB, K.; GOYAL, M. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and Informatics*, v. 26, n. 4, p. 30–45, 1996.
- DEB, K.; JAIN, H. Handling many-objective problems using an improved nsga-ii procedure. In: *IEEE Congress on Evolutionary Computation (CEC)*. USA: IEEE, 2012.
- DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, v. 18, n. 4, p. 577–601, Aug 2014.
- DEB, K.; KUMAR, A. Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2007. p. 781–788.
- DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, 2002.
- DEB, K.; SUNDAR, J. Reference point based multi-objective optimization using evolutionary algorithms. In: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2006. p. 635–642.
- DEB, K.; THIELE, L.; LAUMANN, M.; ZITZLER, E. Scalable test problems for evolutionary multiobjective optimization. In: *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*. London: Springer London, 2005. p. 105–145.
- DORIGO, M. *Optimization, Learning and Natural Algorithms*. Tese (Doutorado) — Politecnico di Milano, Italy, 1992.
- DORIGO, M.; BLUM, C. Ant colony optimization theory: A survey. *Theoretical Computer Science*, v. 344, n. 2, p. 243–278, 2005.
- DRECHSLER, N.; DRECHSLER, R.; BECKER, B. Multi-objective optimisation based on relation favour. In: *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*. London, UK, UK: Springer-Verlag, 2001. (EMO '01), p. 154–166.
- DURILLO, J. J.; NEBRO, A. J. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, v. 42, p. 760–771, 2011.
- ENGELBRECHT, A. P. *Fundamentals of Computational Swarm Intelligence*. USA: John Wiley & Sons, 2006.
- ENGELBRECHT, A. P. *Computational Intelligence An Introduction*. The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England: John Wiley Sons, 2007.
- FARINA, M.; AMATO, P. A fuzzy definition of "optimality" for many-criteria optimization problems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, v. 34, n. 3, p. 315–326, 2004.
- FIGUEIREDO, E.; LUDERMIR, T.; BASTOS-FILHO, C. Many objective particle swarm optimization. *Information Sciences*, v. 374, p. 115–134, 2016.

FIGUEIREDO, E. M. do N. *Algoritmo Baseado em Enxame de Partículas para Otimização de Problemas com Muitos Objetivos*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2013.

FIGUEIREDO, E. M. N.; ARAÚJO, D. R. B.; BASTOS-FILHO, C. J. A.; LUDERMIR, T. B. Physical topology design of optical networks aided by many-objective optimization algorithms. In: *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*. USA: IEEE, 2016. p. 409–414.

FILHO, C. J. A. B.; NETO, F. B. de L.; LINS, A. J. C. C.; NASCIMENTO, A. I. S.; LIMA, M. P. A novel search algorithm based on fish school behavior. In: *2008 IEEE International Conference on Systems, Man and Cybernetics*. USA: IEEE, 2008. p. 2646–2651.

GANDOMI, A. H.; YANG, X.-S.; ALAVI, A. H.; TALATAHARI, S. Bat algorithm for constrained optimization tasks. *Neural Computing and Applications*, v. 22, n. 6, p. 1239–1255, 2013.

Garza Fabre, M. *Optimización de problemas con más de tres objetivos mediante algoritmos evolutivos*. Dissertação (Mestrado) — Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Ciudad Victoria, Tamaulipas, México, 2009.

Garza Fabre, M.; PULIDO, G. T.; Coello Coello, C. A. Ranking methods for many-objective optimization. In: *Proceedings of the 8th Mexican International Conference on Artificial Intelligence*. USA: IEEE, 2009. (MICAI '09), p. 633–645.

GARZA-FABRE, M.; PULIDO, G. T.; Coello Coello, C. A. Two Novel Approaches for Many-Objective Optimization. In: *Congress on Evolutionary Computation*. USA: IEEE, 2010. p. 1–8.

HADKA, D.; REED, P. Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. *Evol. Comput.*, MIT Press, Cambridge, MA, USA, v. 20, n. 3, p. 423–452, set. 2012.

HADKA, D.; REED, P. M.; SIMPSON, T. W. Diagnostic assessment of the borg moea for many-objective product family design problems. In: *2012 IEEE Congress on Evolutionary Computation*. USA: IEEE, 2012. p. 1–10.

HEDAYATZADEH, R.; SALMASSI, F. A.; KESHTGARI, M.; AKBARI, R.; ZIARATI, K. Termite colony optimization: A novel approach for optimizing continuous problems. In: *2010 18th Iranian Conference on Electrical Engineering*. USA: IEEE, 2010. p. 553–558.

HORN, J.; NAFPLIOTIS, N.; GOLDBERG, D. E. A niched pareto genetic algorithm for multiobjective optimization. In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. USA: IEEE, 1994. p. 82–87.

HU, W.; YEN, G. G. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system. *IEEE Transactions on Evolutionary Computation*, v. 19, n. 1, p. 1–18, Feb 2015.

HUBAND, S.; HINGSTON, P.; BARONE, L.; WHILE, L. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, v. 10, n. 5, p. 477–506, Oct 2006.

- HUGHES, E. J. Multiple single objective pareto sampling. In: *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*. USA: IEEE, 2003. v. 4, p. 2678–2684.
- HUGHES, E. J. Msops-ii: A general-purpose many-objective optimiser. In: *2007 IEEE Congress on Evolutionary Computation*. USA: IEEE, 2007. p. 3944–3951.
- IMRAN, M.; HASHIM, R.; KHALID, N. E. A. An overview of particle swarm optimization variants. *Procedia Engineering*, v. 53, p. 491–496, 2013.
- ISHIBUCHI, H.; MASUDA, H.; NOJIMA, Y. Pareto fronts of many-objective degenerate test problems. *IEEE Transactions on Evolutionary Computation*, v. 20, n. 5, p. 807–813, Oct 2016.
- ISHIBUCHI, H.; SETOGUCHI, Y.; MASUDA, H.; NOJIMA, Y. Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Transactions on Evolutionary Computation*, v. 21, n. 2, p. 169–190, April 2017.
- ISHIBUCHI, H.; TSUKAMOTO, N.; NOJIMA, Y. Evolutionary many-objective optimization: A short review. In: *IEEE Congress on Evolutionary Computation*. USA: IEEE, 2008.
- JAIMES, A. L.; COELLO, C. A. C.; CHAKRABORTY, D. Objective reduction using a feature selection technique. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2008. p. 673–680.
- JAIMES, A. L.; Coello Coello, C. A. Some techniques to deal with many-objective problems. In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2009. (GECCO '09), p. 2693–2696.
- JAIN, H.; DEB, K. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, v. 18, n. 4, p. 602–622, Aug 2014.
- JR., O. R. C.; SANTANA, R.; POZO, A. C-multi: A competent multi-swarm approach for many-objective problems. *Neurocomputing*, v. 180, p. 68–78, 2016.
- KACHROUDI, S.; GROSSARD, M. Average rank domination relation for NSGAI and SMPSO algorithms for many-objective optimization. In: *2nd World Congress on Nature and Biologically Inspired Comp. (NaBIC)*. USA: IEEE, 2010. p. 19–24.
- KARABOGA, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*. Turkey, 2005.
- KARABOGA, D.; BASTURK, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, v. 39, n. 3, p. 459–471, 2007.
- KARABOGA, D.; BASTURK, B. On the performance of artificial bee colony (abc) algorithm. *Applied Soft Computing*, v. 8, n. 1, p. 687–697, 2008.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*. USA: IEEE, 1995. p. 1945–1948.

- KENNEDY, J.; EBERHART, R. C. *Swarm Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.
- KHARE, V. *Performace scaling of multi-objective evolutionary algorithms*. Dissertação (Mestrado) — The University of Birmingham, 2002.
- KUKKONEN, S.; LAMPINEN, J. Ranking-dominance and many-objective optimization. In: *2007 IEEE Congress on Evolutionary Computation*. USA: IEEE, 2007. p. 3983–3990.
- LAZINICA, A. *Particle Swarm Optimization*. Croatia: In-Tech, 2009.
- LEWIS, T. G. *Network Science-Theory and Applications*. New Jersey, USA: John Wiley & Sons, 2009.
- LI, B.; LI, J.; TANG, K.; YAO XINLI, B.; LI, J.; TANG, K.; YAO, X. Many-objective evolutionary algorithms: A survey. *ACM Comput. Surv.*, 2015.
- LI, K.; DEB, K.; ZHANG, Q.; KWONG, S. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, v. 19, n. 5, p. 694–716, Oct 2015.
- LI, K.; KWONG, S.; CAO, J.; LI, M.; ZHENG, J.; SHEN, R. Achieving balance between proximity and diversity in multi-objective evolutionary algorithm. *Information Sciences*, v. 182, n. 1, p. 220–242, 2012.
- LI, L.; WANG, W.; XU, X. Multi-objective particle swarm optimization based on global margin ranking. *Information Sciences*, v. 375, p. 30–47, 2017.
- LI, M.; GROSAN, C.; YANG, S.; LIU, X.; YAO, X. Multi-line distance minimization: A visualized many-objective test problem suite. *IEEE Transactions on Evolutionary Computation*, p. 1–1, 2017.
- LI, M.; YANG, S.; LIU, X. Shift-based density estimation for pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 18, n. 3, p. 348–365, June 2014.
- LI, M.; YANG, S.; LIU, X. A test problem for visual investigation of high-dimensional multi-objective search. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. USA: IEEE, 2014. p. 2140–2147.
- LI, M.; ZHENG, J.; SHEN, R.; LI, K.; YUAN, Q. A grid-based fitness strategy for evolutionary many-objective optimization. In: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2010. (GECCO '10), p. 463–470.
- LUO, J.; LIU, Q.; YANG, Y.; LI, X.; CHEN, M. rong; CAO, W. An artificial bee colony algorithm for multi-objective optimisation. *Applied Soft Computing*, v. 50, p. 235–251, 2017.
- MARTENS, D.; BAESENS, B.; FAWCETT, T. Editorial survey: swarm intelligence for data mining. *Machine Learning*, v. 82, n. 1, p. 1–42, 2011.
- MEZA, J.; ESPITIA, H.; MONTENEGRO, C.; GIMÉNEZ, E.; GONZÁLEZ-CRESPO, R. Movpso: Vortex multi-objective particle swarm optimization. *Applied Soft Computing*, v. 52, p. 1042–1057, 2017.

- M.FARINA; AMATO, P. On the optimal solution definition for many-criteria optimization problems. In: *Fuzzy Information Processing Society, 2002. Proceedings. NAFIPS. 2002 Annual Meeting of the North American*. USA: IEEE, 2002.
- MILLER, B. L.; GOLDBERG, D. E. Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, v. 4, n. 2, p. 113–131, June 1996.
- MILLER, B. L.; MILLER, B. L.; GOLDBERG, D. E.; GOLDBERG, D. E. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, v. 9, p. 193–212, 1995.
- MISHRA, B. S. P.; DEHURI, S.; CHO, S.-B. Swarm intelligence in multiple and many objectives optimization: A survey and topical study on eeg signal analysis. In: _____. *Multi-objective Swarm Intelligence: Theoretical Advances and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. p. 27–73.
- NEBRO, A. J.; DURILLO, J. J.; COELLO, C. A. C. Analysis of leader selection strategies in a multi-objective particle swarm optimizer. In: *2013 IEEE Congress on Evolutionary Computation*. USA: IEEE, 2013. p. 3153–3160.
- NEBRO, A. J.; DURILLO, J. J.; García-Nieto, J.; Coello Coello, C. A.; LUNA, F.; ALBA, E. SMP SO: A New PSO-based Metaheuristic for Multi-objective Optimization. In: *IEEE Symp. on Comp. Intelligence in Multicriteria Decision-Making (MCDM 2009)*. USA: IEEE Press, 2009. p. 66–73.
- NOCEDAL, J.; WRIGHT, S. J. *Numerical Optimization*. USA: Springer-Verlag, 1999.
- OROSA, L. M. *Scalable Multi-Objective Optimization*. Tese (Doutorado) — UNIVERSIDAD CARLOS III DE MADRID, 2011.
- PARSOPOULOS, K. E.; VRAHATIS, M. N. Particle swarm optimization method in multiobjective problems. In: *Proceedings of the 2002 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2002. p. 603–607.
- PASSINO, K. M. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems*, v. 22, n. 3, p. 52–67, Jun 2002.
- PIOTROWSKI, A. P.; NAPIORKOWSKI, M. J.; NAPIORKOWSKI, J. J.; ROWINSKI, P. M. Swarm intelligence and evolutionary algorithms: Performance versus speed. *Information Sciences*, v. 384, p. 34–85, 2017.
- POLI, R.; KENNEDY, J.; BLACKWELL, T. Particle swarm optimization. *Swarm Intelligence*, v. 1, n. 1, p. 33–57, 2007.
- PURSHOUSE, R.; FLEMING, P. Evolutionary many-objective optimisation: an exploratory analysis. In: *Congress on Evolutionary Computation*. USA: IEEE, 2003. v. 3, p. 2066–2073.
- PURSHOUSE, R. C. *On the Evolutionary Optimisation of Many Objectives*. Tese (Doutorado) — University of Sheffield, 2003.
- RAO, S. S. *Engineering Optimization Theory and Practice*. New Jersey, USA: John Wiley & Sons, 2009.

- RAQUEL, C. R.; NAVAL JR., P. C. An effective use of crowding distance in multiobjective particle swarm optimization. In: *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2005. p. 257–264.
- REYES-SIERRA, M.; COELLO, C. A. C. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *Int. Journal of Computational Intelligence Research*, v. 2, n. 3, p. 287–308, 2006.
- RUNKLER, T. A. Wasp swarm optimization of the c-means clustering model. *Int. J. Intell. Syst.*, John Wiley & Sons, Inc., New York, NY, USA, v. 23, n. 3, p. 269–285, 2008. ISSN 0884-8173.
- SANTANA, R. A.; PONTES, M. R.; BASTOS-FILHO, C. J. A. A multiple objective particle swarm optimization approach using crowding distance and roulette wheel. In: *2009 Ninth International Conference on Intelligent Systems Design and Applications*. USA: IEEE, 2009. p. 237–242.
- SATO, H.; AGUIRRE, H. E.; TANAKA, K. Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs. In: *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization*. Berlin, Heidelberg: Springer-Verlag, 2007. (Lecture Notes in Computer Science, v. 4403), p. 5–20.
- SAXENA, D. K.; DEB, K. Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding. In: _____. *Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 772–787.
- SEADA, H.; DEB, K. A unified evolutionary optimization procedure for single, multiple, and many objectives. *IEEE Transactions on Evolutionary Computation*, v. 20, n. 3, p. 358–369, June 2016.
- SINDHYA, K.; MIETTINEN, K.; DEB, K. A hybrid framework for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 17, n. 4, p. 495–511, 2013.
- SINGH, H.; ISAACS, A.; RAY, T. A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, v. 15, n. 4, p. 539–556, aug. 2011.
- TALBI, E.-G. *Metaheuristics: from design to implementation*. New Jersey, USA: John Wiley & Sons, 2009.
- TALUKDER, S. *Mathematical Modelling and Applications of Particle Swarm Optimization*. Dissertação (Mestrado) — Blekinge Institute of Technology, 2011.
- TAN, Y. yan; JIAO, Y. chang; LI, H.; WANG, X. kuan. Moead plus uniform design: A new version of moead for optimization problems with many objectives. *Computers & Operations Research*, v. 40, n. 6, p. 1648–1660, 2013.
- Van Den Bergh, F. *An Analysis of Particle Swarm Optimizers*. Tese (Doutorado) — University of Pretoria, 2006.

- Van Den Bergh, F.; ENGELBRECHT, A. A study of particle swarm optimization particle trajectories. *Information Sciences*, v. 176, n. 8, p. 937–971, 2006.
- WANG, H.; JIAO, L.; YAO, X. Twoarch2: An improved two-archive algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 19, n. 4, p. 524–541, Aug 2015.
- WANG, H.; YAO, X. Corner sort for pareto-based many-objective optimization. *IEEE Transactions on Cybernetics*, v. 44, n. 1, p. 92–102, 2014.
- WANG, R.; PURSHOUSE, R.; FLEMING, P. Local preference-inspired co-evolutionary algorithms. In: *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2012. p. 513–520.
- WANG, R.; PURSHOUSE, R. C.; FLEMING, P. J. On finding well-spread pareto optimal solutions by preference-inspired co-evolutionary algorithm. In: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2013. p. 695–702.
- WANG, R.; PURSHOUSE, R. C.; FLEMING, P. J. Preference-inspired co-evolutionary algorithm using weights for many-objective optimization. In: *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2013. (GECCO '13 Companion), p. 101–102.
- WANG, R.; PURSHOUSE, R. C.; FLEMING, P. J. Preference-inspired coevolutionary algorithms for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 17, n. 4, p. 474–494, 2013.
- WANG, R.; ZHANG, T.; GUO, B. An enhanced moea/d using uniform directions and a pre-organization procedure. In: *2013 IEEE Congress on Evolutionary Computation*. USA: IEEE, 2013. p. 2390–2397.
- WHILE, L.; BRADSTREET, L.; BARONE, L. A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation*, v. 16, n. 1, p. 86–95, Feb 2012.
- WICKRAMASINGHE, U.; LI, X. Using a distance metric to guide PSO algorithms for many-objective optimization. In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2009. (GECCO '09), p. 667–674.
- XIANG, Y.; ZHOU, Y.; LI, M.; CHEN, Z. A vector angle-based evolutionary algorithm for unconstrained many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 21, n. 1, p. 131–152, Feb 2017.
- YANG, S.; LI, M.; LIU, X.; ZHENG, J. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 17, n. 5, p. 721–736, Oct 2013.
- YANG, X.-S. *Engineering Optimization: An Introduction with Metaheuristic Applications*. New Jersey, USA: John Wiley & Sons, 2010.
- YANG, X.-S. *Nature-Inspired Metaheuristic Algorithms: Second Edition*. United Kingdom: Luniver Press, 2010.

- YUAN, Y.; XU, H.; WANG, B.; YAO, X. A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 20, n. 1, p. 16–37, Feb 2016.
- ZHANG, Q.; LI, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.*, v. 11, n. 6, p. 712–731, 2007.
- ZHANG, X.; TIAN, Y.; JIN, Y. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 19, n. 6, p. 761–776, Dec 2015.
- ZHANG, Y.; WANG, S.; JI, G. A comprehensive survey on particle swarm optimization algorithm and its applications. *Mathematical Problems in Engineering*, Hindawi Publishing Corporation, v. 2015, 2015.
- ZHOU, A.; QU, B.-Y.; LI, H.; ZHAO, S.-Z.; SUGANTHAN, P. N.; ZHANG, Q. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, v. 1, n. 1, p. 32–49, 2011.
- ZITZLER, E. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Tese (Doutorado) — Institut für Technische Informatik und Kommunikationsnetze, 1999.
- ZITZLER, E.; KÜNZLI, S. Indicator-based selection in multiobjective search. In: _____. *Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference, Birmingham, UK, September 18-22, 2004. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 832–842. ISBN 978-3-540-30217-9. Disponível em: <http://dx.doi.org/10.1007/978-3-540-30217-9_84>.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Zurich, Switzerland, 2001.
- ZITZLER, E.; THIELE, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, v. 3, n. 4, p. 257–271, Nov 1999.
- ZITZLER, E.; THIELE, L.; LAUMANN, M.; FONSECA, C. M.; FONSECA, V. G. da. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 2, p. 117–132, 2003.