



**Universidade
Federal de Itajubá**

TESE DE DOUTORADO

**Algoritmo de Otimização Multi-Objetivo
Assistida por Metamodelagem com Aplicações
em Problemas de Aerodinâmica**

Autor:

Nelson Jose Díaz Gautier

**INSTITUTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

Itajubá, Abril de 2020

Minas Gerais – Brasil

UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

TESE DE DOUTORADO

Algoritmo de Otimização Multi-Objetivo Assistida por Metamodelagem com Aplicações em Problemas de Aerodinâmica

Autor: **Nelson Jose Díaz Gautier**

Orientador: **Nelson Manzanares Filho**

Co-orientadora: **Edna Raimunda da Silva Ramirez**

Curso: **Doutorado em Engenharia Mecânica**

Área de Concentração: **Térmica, e Máquinas de Fluxo**

Tese submetida ao Programa de Pós-Graduação em Engenharia Mecânica como parte dos requisitos para obtenção do Título de Doutor em Ciências em Engenharia Mecânica.

Itajubá, Abril de 2020

Minas Gerais – Brasil

UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

TESE DE DOUTORADO

Algoritmo de Otimização Multi-Objetivo Assistida por Metamodelagem com Aplicações em Problemas de Aerodinâmica

Autor:

Nelson Jose Díaz Gautier

Tese aprovada por banca examinadora em 30 de Abril de 2020, conferindo ao autor o título de **Doutor em Ciências em Engenharia Mecânica**.

Banca Examinadora:

Prof. Dr. Ney Rafael Secco - ITA

Prof. Dr. Antônio Marcos Gonçalves de Lima - UFU

Prof. Dr. Ramiro Gustavo Ramirez Camacho - IEM /UNIFEI

Prof. Dr. Jesus García Sanchez - IEM /UNIFEI

Profa. Dra. Edna Raimunda da Silva Ramirez - LHV/ (Co-orientador)

Prof. Dr. Nelson Manzanares Filho - IEM/UNIFEI (Orientador)

Dedicatória

A Deus e à meu anjo guerreiro Andrés filho amado fonte da minha inspiração, atingimos a meta filho.

A minha filha amada Sabrina minha bebé obrigado pelo seu amor e por estar sempre presente e confiar em mi.

A minha amada esposa Belkys, sem você, sem seu amor, paciência e apoio eu não poderia haver culminado com certeza.

A minha amada mãe, Mercedes e a meu pai amado Sabás, que sempre me apoiaram e se sacrificaram muito para me garantir um melhor futuro e para que Eu fosse um homem do bem.

A minhas famílias, Diaz Gautier, Ramirez Alarcón e Quiroz Suarez, especialmente minha irmã Glamerly de Mijares seus filhos e seu esposo Jorge Mijares, Don Sixto Jesús Ramirez e Dona Alicia Alarcón de Ramirez, Ernesto Júlio Quiroz e Leopoldina Suarez.... Os amo à todos.

A meus amados irmãos Venezuelanos.

Agradecimentos

Todos os que realizam um trabalho de pesquisa como este têm plena consciência de que não é possível fazer sozinho, neste pequeno espaço pretendo agradecer a todas aquelas pessoas que estiveram comigo durante esta parte do processo, sem vocês este humilde trabalho não teria se tornado possível, espero não ser injusto pelo fato algumas pessoas não aparecerem aqui.

Aos orientadores deste trabalho, Prof. Dr. Nelson Manzanares Filho. Pela orientação, ensinamentos, experiências e incentivo, obrigado por acreditar na minha capacidade de conduzir esta pesquisa, um especial agradecimento pelos questionamentos, ideias, ensinamentos e pelas horas dedicadas a este trabalho as quais foram essenciais para o desenvolvimento e conclusão do mesmo, palavras nunca serão suficientes para agradecer. À Profa. Dra. Edna Raimunda da Silva de Ramirez, pelo seu apoio em tudo momento em que eu precisei, da mesma forma ao Prof. Dr. Ramiro Gustavo Ramirez Camacho pela ajuda logística e acadêmica brindada no Laboratório de Hidrodinâmica Virtual (LHV).

Ao Instituto de Engenharia Mecânica da UNIFEI, à Pró-reitora de Pós-Graduação e aos seus dedicados funcionários e professores pelos conhecimentos transmitidos e os sábios conselhos, à Universidade Nacional Experimental da Força Armada UNEFA da Venezuela Campus Táchira e todo seus professores pelo seu apoio especialmente à Cnel. Carlota Reverón.

À Coordenação de Aperfeiçoamento de Pessoal a Nível Superior (CAPES), pelo apoio financeiro através do programa de bolsas e acesso a bibliografia atualizada. E finalmente um respeitoso e adiantado agradecimento aos membros da banca: Dr. Ney Rafael Secco, Dr. Antônio Marcos Goncalves de Lima, Dr. Ramiro Ramirez e o Dr. Jesús García por ter aceitado corrigir, fazer sugestões e contribuir para alcançar a excelência deste trabalho.

Aos colegas e parceiros: Germán, Tania, Yina, Edwin, Donald, Marcela, Angel e família, Darwin e familia, Yohan, Yullian, Hayle e Paola pelo apoio, amizade, companheirismo, convívio e especialmente pelas breves, mas produtivas trocas de ideias e conhecimentos.

Às minhas famílias, Diaz, Ramirez e Quiroz, país, irmãos, tios e sobrinhos. Agradeço a todos vocês que mesmo na distância têm sido minha fonte inesgotável de inspiração, forças e exemplo para não desistir dos meus sonhos. Especialmente a Don Sixto Ramirez, obrigado por tudo, Glamerly de Mijares por todo seu apoio e pelo cuidado da nossa mãe e Carolina Ramirez por seu apoio em todo momento desde Venezuela.

“O meu ideal político é a democracia.

Seja cada homem respeitado como indivíduo, e ninguém idolatrado”

Albert Einstein.

Resumo

DÍAZ, N. J. G. (2020), *Algoritmo de Otimização Multi-Objetivo Assistida por Metamodelagem com Aplicações em Problemas de Aerodinâmica*, Itajubá, 175 f. Tese – Térmica, Fluidos e Máquinas de Fluxo – Instituto de Engenharia Mecânica, Universidade Federal de Itajubá.

Os algoritmos de otimização multi-objetivo comumente usados em projetos reais de engenharia são baseados em estratégias evolutivas que requerem frequentemente um grande número de avaliações das funções objetivo para atingir uma boa aproximação da frente de Pareto. Quando esses algoritmos são usados para resolver um problema de otimização real de engenharia, em que as funções objetivo são computacionalmente custosas, o tempo necessário para atingir a convergência pode ser proibitivo. Nesse sentido, o foco desta pesquisa foi desenvolver um algoritmo de otimização multi-objetivo, acoplado a uma estratégia de metamodelagem, para aprimorar os processos de otimização em problemas de engenharia. O algoritmo foi desenvolvido com base na construção de metamodelos usando funções de base radial para aproximar as funções computacionalmente custosas. Esses metamodelos são otimizados num processo de amostragem iterativo para obter novos pontos no espaço de decisão, com os quais as próximas avaliações das funções custosas devem ser feitas. Além de poder aplicar-se a problemas multi-objetivo, a estratégia desenvolvida faz com que o algoritmo derive para o algoritmo CORS, no caso de problemas mono-objetivo. Os resultados mostraram um desempenho muito satisfatório do algoritmo desenvolvido quando aplicado, tanto nos problemas de teste selecionados quanto em problemas específicos de engenharia relacionados ao projeto otimizado de pás de turbinas eólicas, otimização aerodinâmica da geometria de asas e projeto otimizado de grades lineares de máquinas de fluxo axiais. Na maioria dos casos, o número de avaliações das funções custosas usadas pelo algoritmo desenvolvido baseado em técnicas de metamodelagem, foi pelo menos três vezes menor do que as empregadas pela aplicação direta de um algoritmo de otimização multi-objetivo até atingir a convergência com valores médios semelhantes das métricas de cobertura e diversidade da frente de Pareto.

Palavras-chave

Algoritmo MO-CORS, otimização multi-objetivo, metamodelos, Funções de Base Radial, superfícies aerodinâmicas, máquinas de fluxo.

Abstract

DÍAZ, N. J. G. (2020), *Metamodel Assisted Multi-Objective Optimization Algorithm with Application in Aerodynamics Problems*, Itajubá, 172 f. Thesis - Thermal, Fluids, and Flow Machines - Institute of Mechanical Engineering, Federal University of Itajubá.

The multi-objective optimization algorithms commonly used in real engineering designs are based on evolutionary strategies. These algorithms often require a large number of evaluations of the objective function to achieve a good approximation of the Pareto front. In the case in which these algorithms are used to solve a real engineering optimization problem, which usually has computationally expensive objective functions, the time required to achieve convergence can be some time unfeasible. In this sense, the focus of this research was to develop a multi-objective optimization algorithm, based on a metamodeling strategy, to improve the optimization processes in engineering problems. The algorithm was developed, based on metamodel construction using radial based functions, to approximate the computationally expensive functions. These metamodels are optimized in an iterative sampling process to obtain new points in the decision space, with which the next expensive function evaluations must be made. In addition to being able to apply to multi-objective problems, the results showed a very satisfactory performance of the developed algorithm when applied to the select test problems chosen herein and in three real engineering problems: optimized design of wind turbine blades, aerodynamic optimization of wing geometry, and optimized design of linear cascades of axial flow machines. In most cases, the number of evaluations of expensive functions used by the developed algorithm was at least 3 times less than the expensive function evaluation employed, during the direct application of the evolutionary multi-objective optimization algorithm to achieve convergence with similar average values of coverage and diversity metrics of Pareto front.

Keywords

MO-CORS algorithm, multi-objective optimization, metamodels, Radial Basis Functions, aerodynamic surfaces, flow machines.

Sumário

Lista de Figuras	v
Lista de Tabelas	viii
Lista de Abreviaturas e Siglas	x
Lista de Símbolos	xii
Capítulo 1	1
Introdução	1
1.1. Contexto e Motivação.....	1
1.2. Revisão da Literatura.....	3
1.3. Justificativa.....	7
1.4. Objetivos do Trabalho.....	8
1.4.1. Objetivos específicos.....	8
1.5. Delineamento do Trabalho.....	8
Capítulo 2	10
Problemas de Otimização	10
2.1. Otimização Mono Objetivo.....	11
2.1.1. Otimização local e global em problemas mono-objetivo.....	11
2.1.2. Convexidade e não convexidade em otimização mono-objetivo.....	12
2.1.3. Otimização robusta.....	13
2.1.4. Taxa de convergência.....	13
2.2. Otimização Multi-Objetivo.....	14
2.2.1. Espaço de decisão e espaço objetivo.....	15
2.2.2. Linearidade e não linearidade em otimização multi-objetivo.....	16
2.2.3. Convexidade e não convexidade em otimização multi-objetivo.....	16
2.2.4. Vetor objetivo ideal e conceito de dominância.....	17
2.2.5. Conjunto não dominado e frente de Pareto.....	18
2.2.6. Algoritmos para identificação de conjuntos não dominados.....	20
2.3. Abordagem dos Problemas de Otimização Multi-Objetivo.....	22
2.3.1. Métodos Clássicos.....	22
2.3.2. Métodos Diretos.....	24

2.4.	Algoritmos Evolutivos e Populacionais.....	24
2.4.1.	Algoritmos genéticos.....	31
2.4.2.	Algoritmos heurísticos não baseados em algoritmos genéticos.....	34
2.5.	Métricas em Otimização Multi-objetivo.....	37
2.5.1.	Métricas de convergência.....	37
2.5.2.	Métricas de cobertura.....	38
2.5.3.	Métricas de capacidade.....	39
2.5.4.	Métricas de diversidade.....	39
2.5.5.	Métricas de Sucesso.....	40

Capítulo 3 **41**

	Otimização Assistida por Metamodelos	41
3.1.	Tipos de Otimização Assistida por Metamodelos.....	42
3.2.	Algoritmos de otimização assistida por metamodelos (AOAM).....	42
3.3.	Plano de Experiências.....	46
3.4.	Construção do Metamodelo.....	47
3.4.1.	Funções de Base Radial.....	48
3.4.2.	Determinação automática do parâmetro de forma c	50
3.5.	Problema de Otimização Auxiliar.....	51
3.6.	Seleção dos Novos Pontos de Avaliação.....	52
3.6.1.	Amostragem aleatória dos pontos não dominados.....	52
3.6.2.	Amostragem aleatória no espaço de decisão.....	54
3.6.3.	Busca dos extremos da Frente de Pareto.....	54
3.6.4.	Máxima das mínimas distâncias no espaço de decisão.....	55
3.6.5.	Máxima das mínimas distâncias no espaço objetivo.....	55
3.6.6.	Máxima distância de aglomeração.....	55
3.7.	Otimização Restrita Assistida Por Metamodelos (CORS).....	56

Capítulo 4 **58**

	Otimização Multi-Objetivo Restrita Assistida por Metamodelos	58
4.1.	Definição Geral do Algoritmo MO-CORS.....	59
4.2.	Aspectos Relevantes dos AOAM dentro do Algoritmo MO-CORS.....	59
4.2.1.	Entrada de dados.....	59
4.2.2.	Construção do banco de dados inicial.....	60
4.2.3.	Critério de parada.....	61
4.2.4.	Determinação da matriz de cobertura.....	62
4.2.5.	Avaliação no modelo real (funções objetivo).....	63
4.2.6.	Determinação do conjunto não dominado.....	64
4.2.7.	Normalização dos espaços de decisão e objetivo.....	64
4.2.8.	Construção do metamodelo.....	65
4.2.9.	Problemas de otimização auxiliar.....	66

4.2.10. Seleção de pontos para avaliação na função custosa.....	69
4.2.11. Avaliação do conjunto de pontos selecionados na função custosa.....	73
4.2.12. Atualização do banco de dados.....	74
4.3. Algoritmo MO-CORS.....	74
4.4. Problemas de Teste.....	79
4.4.1. Experimentos computacionais mono-objetivo.....	79
4.4.2. Experimentos computacionais multi-objetivo.....	83
4.4.3. Análise de resultados dos problemas de teste multi-objetivo.....	94
4.4.4. Contribuição dos problemas auxiliares.....	97

Capítulo 5 **100**

Aplicações em Problemas de Aerodinâmica e Máquinas De Fluxo	100
5.1. Otimização das Pás de uma Turbina Eólica.....	100
5.1.1. Teoria de quantidade de movimento do elemento da pá (BEM).....	101
5.1.2. Problema de otimização da pá.....	102
5.1.4. Metodologia de projeto otimizado das pás.....	105
5.1.5. Resultados dos processos de otimização das pás.....	106
5.1.6. Curvas operacionais dos projetos resultantes.....	108
5.2. Otimização Aerodinâmica de Asas.....	109
5.2.1. Função objetivo para problema de asas subsônicas.....	109
5.2.2. Problemas de otimização de asas subsônicas.....	111
5.2.3. Resultados dos problemas de otimização de asas.....	114
5.3. Otimização de Grade Linear.....	120
5.3.1. Cálculo das funções para problema de grade linear.....	121
5.3.2. Problema de otimização de grade linear.....	126
5.3.3. Resultados do problema de otimização da grade linear.....	127
5.3.5. Curvas das características operativas da grade ótima.....	131
5.4. Tempo Computacional Próprio do MO-CORS.....	133

Capítulo 6 **1344**

Conclusões e Trabalhos Futuros	1344
6.1 Conclusões.....	1344
6.1.1. Conclusões gerais sobre o desempenho do algoritmo MO-CORS.....	1344
6.1.2. Desempenho de MO-CORS em funções de teste mono-objetivo.....	1355
6.1.3. Desempenho de MO-CORS em problemas de teste multi-objetivo.....	1366
6.1.4. Desempenho de MO-CORS em problemas de aerodinâmica.....	1366
6.2 Trabalhos Futuros.....	1388

Referências Bibliográficas	14040
Apêndice A	1499
Algoritmo para construção de metamodelos baseado em (RBF)	1499
Apêndice B	1533
Algoritmo para automatização da escolha do parâmetro c	1533
B.1. Experimentos computacionais LOOCV.....	
1533	
Apêndice C	1588
Desempenho de algoritmos de otimização em problema aerodinâmico de asas	1588
C.1. Metodologia de otimização.....	
1588	
C.1.1. Diagrama de fluxo do problema de otimização de asas.....	
1599	
C.2. Experimentos computacionais.....	
16060	
C.3. Resultados para os problemas 1 e 2.....	
16060	
C.3.1. Evolução dos algoritmos de otimização (Problema 1).....	
16060	
C.3.2. Evolução dos algoritmos de otimização (Problema 2).....	
1611	
C.3.3. Frente das frentes de Pareto dos problemas 1 e 2.....	
1622	
C.4. Métricas de desempenho.....	
1622	

Lista de Figuras

Figura 2.1 - Mínimos e máximos locais e globais	12
Figura 2.2 - Conjuntos convexos e não convexos	13
Figura 2.3 - Espaço de decisão e espaço objetivo	16
Figura 2.4 - Frente de Pareto e conjuntos não dominados durante a evolução	19
Figura 2.5 - Combinações de frente de Pareto para dois objetivos	20
Figura 2.6 - Abordagem clássica dos algoritmos de otimização multi-objetivo	22
Figura 2.7 - Abordagem direta dos algoritmos de otimização multi-objetivo	24
Figura 2.8 - Diagrama de fluxo simples de algoritmo genético	26
Figura 2.9 - Ilustração gráfica de compartilhamento (<i>sharing</i>)	28
Figura 2.10 - Ilustração gráfica de aglomeração (<i>crowding</i>)	29
Figura 3.1 - Esquema geral do processo de otimização baseado em metamodelos	43
Figura 3.2 - Amostragem aleatória de pontos no espaço de decisão	53
Figura 4.1 - Ilustração da matriz de cobertura, pontos do banco de dados e distância Δ_t	62
Figura 4.2 - Função dominância ($D_{A,B}$)	64
Figura 4.3 - Diagramas para a análise da influência do otimizador e LOOCV em problemas mono-objetivo	82
Figura 4.4 - Métricas e aceleração para o problema MOP1	86
Figura 4.5 - Espaço de decisão e frente de Pareto MOP1	86
Figura 4.6 - Métricas e aceleração para o problema MOP2 (12 Variáveis)	87
Figura 4.7 - Frente de Pareto MOP2 (12 Variáveis)	87
Figura 4.8 - Médias das métricas e aceleração para o problema MOP3	88
Figura 4.9 - Espaço de decisão e frente de Pareto MOP3	88
Figura 4.10 - Métricas e aceleração para o problema MOP4	89
Figura 4.11 - Espaço de decisão e frente de Pareto MOP4	89
Figura 4.12 - Métricas e aceleração para o problema MOP5	90

Figura 4.13 - Espaço de decisão e frente de Pareto MOP5	90
Figura 4.14 - Métricas e aceleração para o problema MOP6	91
Figura 4.15 - Espaço de decisão e frentes de Pareto MOP6	91
Figura 4.16 - Métricas e aceleração para o problema MOP7	92
Figura 4.17 - Espaço de decisão e frentes de Pareto MOP7	92
Figura 4.18 - Métricas e aceleração para o problema ZDT1 (12 variáveis)	93
Figura 4.19 - Frentes de Pareto do problema ZDT1 (12 variáveis)	93
Figura 4.20 - Métricas e aceleração para o problema ZDT2 (6 variáveis)	94
Figura 4.21 - Frentes de Pareto do problema ZDT2 (6 variáveis)	94
Figura 4.22 - Frentes de Pareto e pontos por problema e por iteração (MOP1)	97
Figura 4.23 - Frentes de Pareto e pontos por problema e por iteração (MOP2)	98
Figura 4.24 - Frentes de Pareto e pontos por problema e por iteração (MOP6)	98
Figura 4.25 - Frentes de Pareto e pontos por problema e por iteração (MOP5)	99
Figura 5.1 - Ângulos, velocidades e forças em cada seção da pá	101
Figura 5.2 - Torção da pá	102
Figura 5.3 - Ângulo ψ para a distribuição de raios	106
Figura 5.4 - Distribuições de torção ao longo da pá (MO-CORS e NSGA II)	107
Figura 5.5 - Evolução de MO-CORS e NSGA II	107
Figura 5.6 - Curvas operacionais dos projetos otimizados	108
Figura 5.7 - Parâmetros geométricos da asa	111
Figura 5.8 - Aerofólio NACA 23015 parametrizado com curvas de Bézier	113
Figura 5.9 - Evolução dos conjuntos não dominados MO-CORS (Variando Geometria da asa)	114
Figura 5.10 - Frentes de Pareto do Problema 1 de asas (a) Final (b) Ampliação	115
Figura 5.11 - Frente das frentes entre MO-CORS e NSGA II	115
Figura 5.12 - Contribuição dos problemas auxiliares de MO-CORS (Problema 1 da asa). (a) Final (b) Ampliação	116
Figura 5.13 - Isometria e vistas da asa base e das soluções 1 e 2	117
Figura 5.14 - Evolução dos conjuntos não dominados MO-CORS (Variando Geometria do aerofólio)	118
Figura 5.15 - Frentes de Pareto do Problema 2 de asas (a) Final (b) Ampliação	118
Figura 5.16 - Frente das frentes entre MO-CORS e NSGA II	119
Figura 5.17 - Contribuição dos problemas auxiliares de MO-CORS (problema 2 da asa). (a) Final (b) Ampliação	119
Figura 5.18 - Aerofólios ótimos 1 e 2 do problema aerodinâmico de asas	120

Figura 5.19 - Forças e velocidades na grade linear	122
Figura 5.20 - Curvas características que incluem o projeto base da grade	126
Figura 5.21 - Evolução dos conjuntos não dominados MO-CORS (Problema da grade)	127
Figura 5.22 - Frentes de Pareto MO-CORS e NSGA II. (a) Total (b) Proximidades da base	128
Figura 5.23 - Frente das frentes entre MO-CORS e NSGA II (Problema da grade)	128
Figura 5.24 - Contribuição dos problemas auxiliares de MO-CORS (Problema da grade). (a) Final (b) Ampliação	129
Figura 5.25 - Curvas aerodinâmicas da grade. (a) Curva C_L vs α . (b) Curva C_L/C_D vs α	130
Figura 5.26 - Curvas aerodinâmicas da grade. (a) Curva θ vs α . (b) Curva C_D vs α	130
Figura 5.27 - Curvas de desempenho operacional das grades	132
Figura B.1 - Erro RMS e F_{erro} (l_1 e l_2) - função Shekel 5 - RBF multiquádrica	154
Figura B.2 - Erro RMS e F_{erro} (l_1 e l_2) - função Hartmann 3 - RBF inversa multiquádrica	155
Figura B.3 - Erro RMS e F_{erro} (l_1 e l_2) - função Hartmann 6 - RBF Gaussiana	155
Figura C.1 - Diagrama de fluxo do problema aerodinâmico de asas 1	159
Figura C.2 - Diagrama de fluxo do problema aerodinâmico de asas 2	159
Figuras C.3 - Evolução de (a) MOGAI, (b) MOPSO, (c) MOSA e (d) NSGAI (Prob.1)	160
Figuras C.4 - Evolução de (a) MOGAI, (b) MOPSO, (c) MOSA e (d) NSGAI (Prob.2)	161
Figuras C.5 - Comparação entre frentes (a) problemas 1 e (b) problema 2	162
Figuras C.6 - Métricas de taxa de sucesso percentual (HRM%) e espaçamento (SM)	163
Figura C.7 - Métrica percentual de melhores pontos na frente das frentes (BPV)	163
Figura C.8 - Métrica de distância geracional (GD). (a) MOGAI, (b) MOSA, (c) MOPSO e (d) NSGAI	164

Lista de Tabelas

Tabela 2.1 - Pseudocódigo MOGA.	32
Tabela 2.2 - Pseudocódigo NSGA II.	34
Tabela 2.3 - Pseudocódigo MOPSO.	36
Tabela 2.4 - Pseudocódigo MOSA.	37
Tabela 3.1 - RBF dependentes do parâmetro de forma c .	48
Tabela 3.2 - RBF independentes do parâmetro de forma c .	49
Tabela 3.3 - Especificação das matrizes componentes da Equação 3.6.	49
Tabela 4.1 - Número de identificação da RBF.	66
Tabela 4.2 - Número máximo de pontos selecionados.	70
Tabela 4.3 - Funções de teste mono-objetivo	79
Tabela 4.4 - Número médio de avaliações necessárias para OGM, MO-CORS com CRSI.	80
Tabela 4.5 - Número médio de avaliações necessárias para OGM, MO-CORS com NSGAI	80
Tabela 4.6 - Número médio de avaliações comparativas com MO-CORS-LOOCV	81
Tabela 4.7 - Funções de teste de Van Veldhuizen (MOP's)	84
Tabela 4.8 - Funções de teste de Zitzler (ZDT's)	84
Tabela 4.9 - Valores das métricas de desempenho e aceleração	95
Tabela 4.10 - Contribuição de pontos de cada problema de otimização auxiliar	99
Tabela 5.1 - Dados do projeto base	102
Tabela 5.2 - Limites superiores e inferiores das variáveis de projeto (Problema da pá)	103
Tabela 5.3 - Resultados da otimização da pá da turbina eólica	106
Tabela 5.4 - Valores de C_p para cada pá.	108
Tabela 5.5 - Valores da asa base e limites laterais (Problema 1 da asa)	112
Tabela 5.6 - Valores da asa base e limites laterais (Problema 2 da asa)	113

Tabela 5.7 - Valores das métricas de desempenho para MO-CORS no problema 1 da asa	116
Tabela 5.8 - Valores das variáveis de projeto das soluções 1 e 2 no problema 1 da asa.	117
Tabela 5.9 - Valores das métricas de desempenho para MO-CORS no problema 2 da asa	120
Tabela 5.10 - Valores da grade base e limites laterais	127
Tabela 5.11 - Valores das métricas de desempenho para MO-CORS no problema da grade	129
Tabela 5.12 - Valores das variáveis de projeto da solução 2 no problema da grade	130
Tabela 5.13 - Tempo próprio dos algoritmos MO-CORS e NSGA II.	133
Tabela A.1 - Exemplo: X_i pontos da matriz de banco de dados	149
Tabela A.2 - Exemplo: F_i vetores objetivo para cada ponto do banco de dados	150
Tabela A.3 - Valores da diagonal principal segundo a RBF utilizada	151

Lista de Abreviaturas e Siglas

MO-CORS	<i>Multi-Objective Constrained Optimization Response Surface</i>
CORS	<i>Constrained Optimization Response Surface</i>
CRSA	<i>Controlled Random Search Algorithm</i>
UAV	<i>Unmanned Aerial Vehicle</i>
LHV	Laboratório de H idrodinâmica V irtual
CFD	<i>Computational Fluid Dynamics</i>
NACA	<i>National Advisory Committee for Aeronautics</i>
NSGA	<i>Non-dominated Sorting Genetic Algorithm</i>
VEGA	<i>Vector Evaluated Genetic Algorithm</i>
SPEA	<i>Strength Pareto Evolutionary Algorithm</i>
MOGA	<i>Multi-Objective Genetic Algorithm</i>
AOAM	Algoritmos de O timização A ssistido por M etamodelos
DOE	<i>Design Of Experiment</i>
RBF	<i>Radial Basis Function</i>
LOOCV	<i>Leave One Out Cross Validation</i>
MOOP	<i>Multi-Objective Optimization Problem</i>
MOOA	<i>Multi-Objective Optimization Algorithm</i>
MOEA	<i>Multi-Objective Evolutionary Algorithm</i>
MOPSO	<i>Multi-Objective Particles Swarm Optimization</i>
MOSA	<i>Multi-Objective Simulated Annealing</i>
MOPM	<i>Multi-Objective Performance Metric</i>
GD	<i>Generacional Distance</i>
HRM	<i>Hit Rate Metric</i>
MVFP	Melhores Valores na F rente de P areto (BPV)

ACMU	Algoritmos baseados na C onstrução de um M etamodelo Ú nico
ACDM	Algoritmos baseados na C onstrução D inâmica de M etamodados
PFC	P lano F atorial C ompleto
PFI	P lano F atorial I ncompleto
HCL	H iper C ubo L atino
OGM	O timização G lobal por M etamodelagem
CRSI	<i>Controlled Random Search with quadratic Interpolations</i>
TPS	<i>Thin Plate Spline</i>
MOPs	<i>Multi-Objective Problems</i>
ZDTs	<i>Zitzler Data Test</i>
BEM	<i>Blade Element Momentum - Theory</i>
CND	C onjunto N ão D ominado

Lista de Símbolos

Símbolos Latinos

b	Envergadura
c	Fator de forma da RBF
f	Vetor objetivo
j	Contador de variáveis de decisão
k	Contador de funções objetivo
S	Espaço de decisão
$sh(d)$	Função de compartilhamento (<i>sharing</i>)
t	Contador de número de iterações
X	Vetor de decisão
C_0	Corda na raiz
C_f	Coefficiente de atrito
C_L	Coefficiente de sustentação
C_m	Coefficiente de momento
C_p	Coefficiente de potência
C_t	Corda na ponta
C_T	Coefficiente de empuxo
D_p	Diferença percentual
f_d	Função de ponderação das distâncias
f_k	Componentes do vetor objetivo ou funções objetivo
F_p	Fator de perdas de Prandtl
g_m	Função de restrição de desigualdade
h_l	Função de restrição de igualdade
M_p	Matriz de dados avaliados (Matriz da população)
P_3	Ponto resultante do problema auxiliar 3

S_0	Banco de dados inicial
S_N	Conjunto normalizado de pontos seletos
x_j	Variáveis de decisão
f^*	Vetor objetivo ideal
\mathcal{F}^*	Conjunto de pontos do fronte de Pareto
P^*	Conjunto de pontos não dominados
x^*	Ponto ótimo
\tilde{f}_i	Metamodelo em uma iteração i
\hat{f}_k	Componentes normalizadas do vetor de objetivos
\bar{f}_s	Função de ponderação do valor do metamodelo
$\tilde{\mathcal{F}}^*$	Frente de Pareto do metamodelo
S_N^1	Conjunto normalizado de pontos seletos do problema auxiliar 1
S_N^2	Conjunto normalizado de pontos seletos do problema auxiliar 2
S_N^3	Conjunto normalizado de pontos seletos do problema auxiliar 3
\hat{M}_P	Matriz de dados avaliados normalizada (Matriz da população normalizada)
\hat{x}_j	Componentes normalizadas do vetor de decisão
C_{Di}	Coefficiente de arrasto induzido
F_{AR}	Função de peso da nuvem de pontos aleatórios
P_{de}	Ponto que apresenta a máxima das mínimas distância no espaço de decisão
P_{ex}	Conjunto de pontos extremos da frente de Pareto
P_{ob}	Ponto que apresenta a máxima das mínimas distância no espaço objetivo
x_{LI}	Limite lateral inferior
x_{LS}	Limite lateral superior
Z_{Pi}	Componentes do vetor de pontos de controle do aerofólio
N_{exp}	Número de experiências
N_{ind}	Número de indivíduos
N_{niv}	Número de níveis
N_{obj}	Número de funções objetivo
n_{RBF}	Número de identificação da RBF
n_{var}	Número de variáveis de decisão
\bar{d}_{min}	Distância média mínima
N_{eval}	Número de avaliações da função custosa
P_{can1}	Conjunto de pontos candidatos do problema auxiliar 1
P_{can2}	Conjunto de pontos candidatos do problema auxiliar 2
X^{crow}	Ponto com maior distância de aglomeração

f_k^{max}	Valor máximo da k-ésima função objetivo
f_k^{min}	Valor mínimo da k-ésima função objetivo
x_j^{max}	Vetor de valores máximos das variáveis de decisão
x_j^{min}	Vetor de valores mínimos das variáveis de decisão
$I[i]_{dis}$	Distância de aglomeração (<i>crowding</i>)
$ND_{set}(\cdot)$	Função para determinação do conjunto não dominado
$N_{maxeval}$	Número máximo permitido de avaliações da função custosa

Símbolos Gregos

β	Vetor padrão de busca
β	Ângulo de torção da pá (Turbina eólica); Ângulo de montagem (Grade)
γ	Taxa de convergência
Γ	Ângulo de varredura
Δ	Métrica de espalhamento (<i>Spread</i>)
θ	Ângulo de desvio da grade
λ	Relação de velocidades na ponta da pá
σ	Solidez da grade
Φ	Coefficiente de vazão
ϕ	Função de base radial
Ψ	Coefficiente de pressão na pá
Ω	Espaço objetivo
ω	Velocidade angular do rotor
α_i	Coefficientes da função de aproximação (metamodelo)
δ_y	Coefficiente de deflexão
Δ_t	Distância max-min no espaço de decisão
ε_d	Ângulo de dihedral
η_h	Eficiência hidráulica da grade
θ_B	Ângulo de desvio da grade base
θ_M	Ângulo de montagem no cubo
θ_p	Ângulo de montagem na ponta da pá
λ_a	Afunilamento da asa
μ_i	Coefficientes dos polinômios de extensão
χ_i	Pontos aleatórios
ψ_i	Ângulo de divisão da pá

$\boldsymbol{\tau}^*$	Vetor alvo
φ_{ij}	Matriz de raio distâncias
$\mathcal{X}_i^{C_j}$	Nuvem de pontos aleatórios
σ_{share}	Raio de compartilhamento

Capítulo 1

Introdução

1.1. Contexto e Motivação

Nas últimas décadas, os avanços dos recursos computacionais têm permitido aos cientistas desenvolver modelos de simulação computacional de fenômenos físicos cada vez mais complexos e com maior fidelidade. Esses modelos são geralmente definidos por um conjunto de procedimentos que visam resolver numericamente as equações governantes ou modelos empíricos ou semi-empíricos complexos do problema analisado, dentro de uma estrutura que é chamada de “caixa preta”, quando o usuário pode até desconhecer certos procedimentos e parâmetros internos do modelo, cuja resolução é requerida. Essa resolução pode envolver complexas discretizações do domínio geométrico e funcional do espaço analisado devido aos parâmetros internos do modelo, cuja resolução requer a discretização geométrica do espaço analisado. O procedimento básico consiste em definir variáveis de projeto que têm influência significativa no problema, como dados de entrada no modelo físico, ou seja, no modelo “caixa preta”, e então obter as saídas que resultam da solução do problema. Em geral, o uso desses modelos trazem consigo um alto custo computacional, que se torna ainda mais relevante quando as saídas são utilizadas para se resolver problemas de otimização, principalmente aqueles com múltiplos objetivos (multi-objetivo). Nessas situações, os modelos fazem parte da avaliação recorrente de uma ou várias funções objetivo e de restrição, cujas descrições analíticas e de suas derivadas geralmente não estão disponíveis. Assim, são necessários métodos de solução livres de derivadas, (CONN *et al.*, 2009).

Neste contexto, os modelos *substitutos*, ou *metamodelos*, têm sido usados com frequência como aproximações relativamente mais rápidas dos modelos computacionalmente custosos. Basicamente, o uso dessa *metamodelagem* se dá de duas formas. Na primeira delas, a metamodelagem fica baseada em um banco de dados estanque, contendo uma quantidade

significativamente alta de pontos de avaliação das funções custosas. Esse banco de dados é montado previamente e os metamodelos construídos permanecem inalterados ao longo do processo de otimização. Não há dessa forma como se avaliar a priori, se o banco de dados é excessivo ou insuficiente para se atingir uma otimização satisfatória. Na segunda forma de emprego, a metamodelagem parte de um banco de dados inicial, com menor quantidade possível de pontos de avaliação. Realiza-se, então uma busca orientada por melhores soluções do problema, através de um processo iterativo. Em cada iteração, o processo iterativo consiste na construção de metamodelos provisórios de cada função objetivo ou de restrição, que são então aplicados na solução de um problema auxiliar de otimização para fazer a escolha do(s) novo(s) ponto(s) que irão atualizar o banco de dados para a construção de novos metamodelos provisórios na próxima iteração, e assim sucessivamente, até que a convergência seja atingida.

Este trabalho, concentra-se no desenvolvimento de uma metodologia de otimização global multi-objetivo baseada na construção de metamodelos para aplicações em problemas de aerodinâmica. Em particular, faz-se uma extensão para problemas multi-objetivo do trabalho realizado por Silva (2011), em que foi desenvolvida uma técnica de otimização global mono-objetivo baseada em um algoritmo de construção iterativa de superfícies de resposta (metamodelos) com restrições de distância para buscas no espaço de decisão. Trata-se do CORS *Constrained Optimization Response Surface* (REGIS e SHOEMAKER, 2005), em que, os metamodelos são construídos iterativamente usando funções de base radial (RBF). Para a seleção de um novo ponto de avaliação da função custosa (solução do problema auxiliar de otimização), Silva (2011) empregou um algoritmo de otimização populacional de busca aleatória controlada (*Control Random Search Algorithm* CRSA) (MANZANARES-FILHO *et al.*, 2018). O objetivo agora é o desenvolvimento de uma extensão do algoritmo CORS para aplicações em problemas multi-objetivo, referida aqui como MO-CORS - *Multi-Objective Constrained Optimization Response Surface*. Pretende-se conservar a natureza do mesmo, e aproveitar sua característica de convergência já verificada em problemas mono-objetivo, além de incorporar novos procedimentos próprios dos algoritmos multi-objetivo que contribuam para melhorar o desempenho na aceleração de processos de otimização. O algoritmo de otimização mono-objetivo CRSA dará lugar ao algoritmo multi-objetivo NSGA II (DEB *et al.*, 2002). Algumas aplicações de teste em problemas de aerodinâmica incompressível serão apresentadas neste trabalho. Espera-se, contudo, que MO-CORS venha a ser aplicado em diversos problemas de engenharia em geral e, em especial, que possa contribuir para estudos nas linhas de pesquisa do Laboratório de Hidrodinâmica Virtual da UNIFEI (LHV).

A seguir é apresentada uma breve revisão da literatura sobre os algoritmos de otimização multi-objetivo, a utilização de técnicas de metamodelagem em problemas de otimização e as aplicações em problemas de aerodinâmica.

1.2. Revisão da Literatura

Frequentemente, os modelos aplicados em projetos de engenharia envolvem a aplicação de métodos de análise matematicamente complexos para simular a realidade física. Na maioria dos casos, as simulações têm que ser realizadas pelo uso de computadores. Dependendo do nível de fidelidade e complexidade do modelo físico requerido pelo projetista, a avaliação das variáveis de projeto no modelo podem requerer tempos computacionais proporcionalmente elevados. Por exemplo, em problemas de aerodinâmica de asa e de máquinas de fluxo (de que tratam as aplicações feitas neste trabalho), a literatura mostra que a otimização computacional tem sido aplicada há décadas na resolução sistemática de problemas de projeto aerodinâmico. Dados da década de 1970, os trabalhos de Hicks *et al.* (1974) e de Hicks e Henne (1978) foram pioneiros na investigação do problema da otimização de formas aerodinâmicas. Embora encorajador, seus resultados para a otimização de asas e aerofólios mostraram-se bastante inesperados e apontaram várias dificuldades. A principal delas era o número insuficiente de parâmetros de projeto utilizados, devido ao limitado avanço tecnológico da ciência computacional da época e ao inerente custo computacional.

Com o desenvolvimento tecnológico em velocidade computacional e capacidade de armazenamento de dados, a limitação do número de parâmetros de projeto foi amplamente superada. Entre os avanços mais ressaltantes em pesquisa durante a década de 80, é possível mencionar o desenvolvimento de estratégias de otimização para projeto de aerofólios e asas, por exemplo, Drela (1989) empregou um método de otimização global integrado com um esquema inverso para projeto de perfis aerodinâmicos. Cosentino e Holst (1986) empregaram um método potencial tridimensional acoplado a um algoritmo de otimização de uma asa transônica. Trabalhos bem mais recentes orientam as aplicações aeronáuticas para a otimização multidisciplinar de aeronaves. Bravo *et al.* (2017) e Bravo *et al.* (2018) desenvolveram modelos para projeto conceitual de aeronaves de baixo porte como os UAV (*Unmanned Aerial Vehicle*), aeronaves de combate e aeronaves agrícolas, onde se pode observar que os objetivos do projeto envolvem múltiplas disciplinas, dependendo da aplicação da aeronave. Essas tendências estimularam a ampliação da quantidade de variáveis e funções objetivo envolvidas na estruturação de dois problemas de otimização simples de asas utilizados nesta tese para testar o desempenho do algoritmo desenvolvido.

Outra aplicação pretendida é na otimização aerodinâmica de máquinas de fluxo. Neste sentido especificamente, o grupo de pesquisa do LHV tem desenvolvido ao longo dos últimos anos, modelos baseados em CFD. Dentre estes está, o trabalho de Arispe *et al.* (2018), em que a parametrização geométrica e a análise da eficiência do tubo de sucção de uma turbina Francis foram desenvolvidas. Reis *et al.* (2019) propuseram uma metodologia de otimização robusta de

carregamentos aerodinâmicos para projeto inverso de aerofólios. O procedimento é baseado em modelos de camada limite, com o objetivo de reduzir o tempo computacional do processo de otimização robusta, que é computacionalmente custoso por natureza. Castilho (2013) apresenta uma metodologia para otimização de grades representativas de turbomáquinas axiais com perfis da família NACA-65. No esquema de parametrização geométrica, as variáveis de projeto consideradas foram o ângulo de montagem, o espaçamento e o arqueamento dos perfis. O cálculo do escoamento baseou-se no método dos painéis de Hess e Smith (1967) com modificações, de modo a introduzir os efeitos viscosos através do cálculo das camadas-limite e da simulação da esteira descolada. Esta metodologia de interação viscosa/não-viscosa tem como vantagem um baixo custo computacional, sendo favorável para o processo de otimização (e, conseqüentemente, para testes comparativos de algoritmos).

Nos processos de otimização de projetos de engenharia, o tempo computacional é severamente afetado, devido às grandes quantidades de avaliações requeridas pelo algoritmo de otimização durante seu processo de convergência. Em tal sentido os pesquisadores abordaram o desenvolvimento de algoritmos e técnicas especiais de aceleração, orientados pela necessidade de diminuir os altos custos computacionais dos problemas de otimização mono-objetivo, os quais são ainda maiores nos problemas multi-objetivo, de particular interesse neste trabalho. Cumpre notar que esses algoritmos foram inicialmente projetados através da reformulação dos problemas multi-objetivo como mono-objetivo. Nessa linha, destacam-se o método das somas ponderadas, em que se otimiza a soma ponderada de cada uma das funções objetivo, e o método da restrição- ϵ , em que o objetivo mais importante é selecionado para otimização e os valores das outras funções objetivo são colocados como restrições laterais do problema, segundo uma tolerância ϵ , como por exemplo nos trabalhos de Miettinen (2008), Ehrgott (2006) e Coello *et al.* (2007). A desvantagem desses métodos, no entanto, é que a forma de ajuste dos pesos ou a seleção de ϵ , respectivamente, devem ser definidas pelo usuário; contudo, eles podem ser utilizados alternativamente no contexto de algoritmos mais complexos, no intuito de orientar a busca de soluções melhores com um menor tempo computacional. Nos algoritmos multi-objetivo baseados no conceito de dominância, milhares de avaliações das funções objetivo podem ser necessárias para se obter a chamada frente de Pareto (conjunto não-dominado ótimo). Foram assim desenvolvidos diferentes algoritmos de otimização multi-objetivo baseados em heurísticas, as quais promovem a obtenção de ótimos globais. Entre eles, destaca-se o algoritmo NSGA II (DEB *et al.*, 2002), baseado na ordenação de conjuntos de soluções não dominadas utilizando algoritmo genético. Trata-se de uma versão melhorada do algoritmo NSGA (*Non Sorting Genetic Algorithm*) proposto por Srinivas e Deb (1994), com melhorias em relação aos algoritmos: VEGA (*Vector Evaluation Genetic Algorithm*) proposto por Schaffer (1985) e o algoritmo SPEA-II (*Strength Pareto Evolutionary Algorithm*) de Zitzler *et al.* (2001).

Em muitos problemas reais da engenharia as funções objetivo não são dadas de forma explícitas em termos das variáveis de projeto. Conhecidos os valores das variáveis de projeto, os valores das funções objetivo são obtidos através das análises numéricas, como análise estrutural, análise fluidodinâmica, análise termodinâmica e assim por diante. Tais análises podem também ser chamadas de “funções caixa preta”, quando o projetista não tem controle sobre a totalidade dos parâmetros próprios da função objetivo ou modelo computacional.

Através dos algoritmos de otimização mono e multi-objetivo, são diversas as aplicações envolvendo “funções caixa preta” que podem ser feitas no âmbito da engenharia, entre elas a aerodinâmica, para as quais, o uso de algoritmos genéticos prevalece na literatura. Lyu *et al.* (2015) realizaram um *benchmarking* de algoritmos de otimização aplicados à projeto aerodinâmico de asas. Hunsaker *et al.* (2017) realizou um procedimento de otimização aerodinâmica de asas *Morphing* em Várias Condições de Vôo. Bartoli *et al.* (2016) desenvolveram um método de otimização global para o projeto de asas de aeronaves, Diaz (2015) desenvolveu uma ferramenta computacional para a otimização aerodinâmica de um conjunto de asas usando o método de ferradura de vórtices baseado em algoritmo genético; Chen (2014) trabalhou com a otimização aerodinâmica de pás e aerofólios de turbinas eólicas; Caixeta (2017) realizou um projeto aerodinâmico de asas flexíveis baseado em algoritmos de otimização; Obayashi *et al.* (1997) fizeram uma otimização multi-objetivo de asas transônicas de aeronaves de transporte baseada em algoritmo genético (MOGA – *Multi-Objective Genetic Algorithm*); Zhu *et al.* (2014) realizaram aplicações do seu algoritmo de otimização multi-objetivo baseado nas emissões econômicas de potência eólica. Na maioria dos casos anteriores as funções objetivo foram do tipo “caixa preta”.

Frequentemente, as avaliações de “funções caixa preta” consomem consideravelmente mais tempo do que as avaliações de funções de forma explícita. Sendo assim, em longos processos de otimização, é interessante substituir essas funções custosas por funções de aproximação computacionalmente mais baratas, conhecidas na literatura como modelos substitutos, superfícies de resposta ou metamodelos. Sua construção pode ser feita, por exemplo, por métodos polinomiais clássicos, ou métodos baseados em interpolação por funções de base radial. Um esquema geral dos algoritmos de otimização assistido por metamodelos (AOAM) é apresentado por Knowles e Nakayama (2008). Nesse esquema, podem ser definidos os aspectos básicos destes algoritmos, como a definição do banco de dados ou construção do plano de experiências (DOE – *Design Of Experiments*), a definição das funções de aproximação, os métodos de construção das superfícies de resposta e no caso dos problemas multi-objetivo, as técnicas de seleção de pontos. Nesse sentido, um dos pontos mais significativos para os pesquisadores é o estudo de diversos tipos de modelos substitutos, por exemplo, os modelos de regressão polinomial e as *splines* de regressão adaptativa multivariada

(FRIEDMAN, 1991), os modelos de função de base radial (POWELL, 1987) e os modelos de *krigagem* (MATHERON, 1967). Assim, aplicando tais modelos, inclusive os interpolantes, a literatura mostra pesquisas orientadas ao desenvolvimento de algoritmos mono-objetivo e multi-objetivo com diferentes características.

No caso de algoritmos de otimização mono-objetivo, um grande número de autores têm desenvolvido trabalhos, propondo aproximações computacionalmente baratas da função objetivo custosa. Por exemplo, Jones *et al.* (1998), apresentam uma metodologia baseada em superfícies de resposta que é especialmente satisfatória em modelar as funções não-lineares e multimodais que geralmente ocorrem na engenharia. Os autores discutem como essas funções de aproximação podem ser usadas para construir um algoritmo de otimização global eficiente. Regis e Shoemaker (2007), introduzem uma estrutura de programação paralela para otimização global de funções computacionalmente custosas usando modelos de superfícies de resposta; em particular, paralelizam dois métodos de otimização global baseados em RBF de otimização global: o método proposto por Gutmann (2001) e o método CORS-RBF desenvolvido pelos próprios autores (REGIS e SHOEMAKER, 2005). Wild *et al.* (2007) analisaram algoritmos de região de confiança livremente convergentes e sem derivadas baseados em modelos de interpolação de função de base radial. Glaz *et al.* (2008) pesquisaram a eficácia dos modelos substitutos aplicados num problema de análise de vibrações de helicópteros e fizeram uma comparação entre as precisões de três diferentes métodos de aproximação: *krigagem*, interpolação com funções de base radial e regressão polinomial. Müller e Shoemaker (2014) examinaram a influência de dois aspectos principais na qualidade da solução de algoritmos de modelos substitutos para problemas de otimização global de “funções caixa preta” computacionalmente caras, a saber, a escolha do modelo substituto e o método de seleção iterativa de pontos de amostragem. Müller (2015) apresentou uma estrutura de otimização de modelo substituto que visa resolver problemas de otimização de “funções caixa preta” computacionalmente caros com variáveis inteiras, misturadas a variáveis contínuas.

Em relação à aplicação de técnicas que aceleram o processo de otimização em problemas multi-objetivo, os metamodelos acabaram sendo usados para reduzir o custo computacional, dos algoritmos de otimização evolutivos. Nessa linha, alguns trabalhos devem ser destacados; Nain e Deb (2005) combinam NSGA-II com redes neurais artificiais. Syberfeldt *et al.* (2008) usaram redes neurais artificiais com uma estrutura de computação paralela para reduzir o número de avaliações de funções caras que precisam ser feitas. Kunakote e Bureerat (2013) combinam SPEA-II com diferentes tipos de modelos substitutos e examinam a influência da escolha do modelo nos resultados de otimização. Ong *et al.* (2005) desenvolveram um algoritmo baseado em modelos substitutos para resolução de problemas de engenharia com alta fidelidade baseado numa abordagem da região de confiança.

Outra aplicação das técnicas de metamodelagem nos algoritmos de otimização multi-objetivo é na assistência dada aos processos de otimização, em busca de melhores pontos para avaliação, feita de forma iterativa e, como consequência, a melhoria da qualidade do metamodelo nas proximidades dos pontos ótimos. Neste sentido, destacam-se os trabalhos de Akhtar e Shoemaker (2016) que propuseram um algoritmo evolutivo assistido por modelo substituto, com o emprego de uma variedade de métodos de amostragem. Mais recentemente, Datta e Regis (2016) introduziram uma estratégia evolutiva auxiliar baseada em modelos substitutos para a resolução de problemas de otimização multi-objetivo com restrições. Mais recentemente, Müller (2017) desenvolveu uma metodologia baseada em modelos substitutos para a resolução de problemas multi-objetivo com funções objetivo tipo “caixa preta”. Os algoritmos desenvolvidos nos três trabalhos mencionados anteriormente, são exclusivamente multi-objetivo e não usam técnicas que garantam, a priori, a convergência de cada um deles. É interessante observar que a convergência é matematicamente garantida pelo algoritmo CORS (REGIS e SHOEMAKER, 2005), mas unicamente para problemas mono-objetivo.

1.3. Justificativa

Como explicado na seção anterior, os metamodelos para assistir os processos de otimização mono-objetivo e multi-objetivo, têm sido aplicados pelos pesquisadores de duas formas: na primeira, através da construção de um metamodelo estante como uma única superfície de resposta que substitua o modelo computacionalmente custoso durante todo o processo de otimização; neste caso a amostragem necessária tem que ser bem representativa para atingir resultados fidedignos, senão pode não ser viável em muitos problemas de engenharia, quando simulações computacionalmente muito caras estão envolvidas como, por exemplo, aquelas baseadas em Dinâmica de Fluidos Computacional CFD. A segunda forma, consiste na construção de metamodelos aprimorados iterativamente ao longo do processo de otimização; no caso particular dos problemas multi-objetivo a construção destes metamodelos podem também ser usada como técnica de seleção de novos pontos que permitam obter maiores quantidades de boas soluções, com a menor quantidade possível de avaliações da função computacionalmente custosa.

Portanto, seguindo a linha de pesquisa proposta, pretende-se implementar uma metodologia de otimização global que possa ser aplicada tanto para problemas multi-objetivo, como mono-objetivo, estendendo o algoritmo CORS de construção de superfícies de resposta com funções de base radial. O algoritmo tem mecanismos eficazes de exploração global do espaço de decisão, que garantem sucesso de convergência para problemas mono-objetivo, Silva (2011). O trabalho contempla aplicações em aerodinâmica de asas e de máquinas de fluxo e

vislumbra um amplo espectro de aplicações em engenharia, em particular aquelas no âmbito das linhas de pesquisa do Laboratório de Hidrodinâmica Virtual (LHV) da UNIFEI.

1.4. Objetivos do Trabalho

O objetivo principal deste estudo é fornecer um algoritmo de aceleração de processos de otimização multi-objetivo baseado em técnicas de metamodelagem com aplicações em projetos aerodinâmicos.

1.4.1. Objetivos específicos

- Fazer uma revisão bibliográfica dos aspectos teóricos relacionados com: os algoritmos de otimização multi-objetivo, algoritmos de otimização assistido por metamodelos e metodologias de projeto aerodinâmico.
- Desenvolver o algoritmo de otimização multi-objetivo assistida por metamodelos, denominado MO-CORS, como uma extensão do algoritmo CORS, e implementar o correspondente código computacional.
- Aplicar o algoritmo MO-CORS em problemas de teste mono-objetivo e multi-objetivo.
- Avaliar métricas de desempenho de algoritmos de otimização multi-objetivo nos resultados da implementação de MO-CORS nos problemas de teste.
- Testar e avaliar o algoritmo MO-CORS em problemas de aerodinâmica de pás de turbinas eólicas, de asas de aeronaves e de grades lineares de máquinas de fluxo.

1.5. Delineamento do Trabalho

No **Capítulo 2**, são apresentados os conceitos básicos relacionados com problemas de otimização multi-objetivo, como: dominância, não dominância, espaço de decisão, espaço objetivo, linearidade, conjunto não dominado, frente de Pareto e finaliza com a descrição dos diferentes tipos de algoritmos de otimização multi-objetivo evolucionários e as definições das métricas de avaliação de desempenho.

No **Capítulo 3**, é discutido o algoritmo geral de otimização assistida por metamodelos, com uma análise dos seus pontos chave: as técnicas de construção da população inicial, as técnicas mais relevantes para a construção dos metamodelos, os procedimentos para a seleção dos pontos de avaliação custosa.

O **Capítulo 4**, é dedicado à proposição e implementação do algoritmo de metamodelagem para a aceleração de problemas de otimização multi-objetivo. O algoritmo é aplicado em

funções de teste mono-objetivo e problemas de teste multi-objetivo. Para o caso mono-objetivo, é introduzida a automatização da escolha do parâmetro de forma (presente em certas funções de base radial), através de um novo algoritmo baseado na técnica de Validação Cruzada com Um de Fora por Vez (em inglês: *Leave One Out Cross Validation- LOOCV*), (RIPPA, 1999).

No Capítulo 5, são apresentados os resultados para três aplicações de engenharia: a primeira na otimização mono-objetivo de um projeto de pás de turbinas eólicas de eixo horizontal baseado no método de elemento de pá. Na segunda aplicação mostram-se dois problemas de otimização multi-objetivo de asas de aeronaves subsônicas. Por último uma metodologia de otimização de uma grade linear de máquina de fluxo, buscando minimizar o desvio do ângulo de escoamento na saída da grade com respeito ao projeto base, com a menor perda de energia possível.

No Capítulo 6, são apresentadas as conclusões e os enunciados dos trabalhos futuros.

Capítulo 2

Problemas de Otimização

A otimização pode ser entendida como um procedimento iterativo para encontrar e comparar soluções viáveis num domínio particular, até que não seja possível encontrar uma solução melhor num problema específico. As soluções são denominadas melhores ou piores em termos do valor que apresentam ao serem avaliadas por *funções objetivo*. Os *problemas de otimização* têm então por finalidade, encontrar a melhor combinação dentre um conjunto de variáveis de decisão \mathbf{x} representadas numa *região viável* que minimize ou maximize uma *função objetivo* $f(\mathbf{x})$ (COELLO, 2011). Na prática, é comum considerar apenas problemas de minimização da função objetivo, pois aqueles que requerem maximização podem ser representados também como problemas de minimização, uma vez que maximizar $f(\mathbf{x})$, é equivalente a minimizar $-f(\mathbf{x})$. As funções objetivo no campo da engenharia são representadas geralmente: pelo custo de fabricação, pela eficiência de um processo ou uma máquina, pela confiabilidade de um produto ou outros fatores. Um passo importante no processo da otimização é fazer a correta classificação dos problemas a serem otimizados, uma vez que os diversos algoritmos existentes são adaptados para cada tipo de problema. Os problemas podem ser classificados com base na natureza das *variáveis de decisão*, da *função objetivo* e das *restrições*, podendo ser contínuos ou discretos, restritos ou irrestritos, lineares ou não-lineares, convexos ou não-convexos, unimodais ou multimodais, locais ou globais, mono-objetivo ou multi-objetivo, dentre outros. Um grande grupo de pesquisas e aplicações no campo da otimização considera um único objetivo, embora a maioria dos problemas reais envolva mais de um objetivo. A seguir são apresentadas definições importantes no campo da otimização mono-objetivo e multi-objetivo, no intuito de que algum deles sejam utilizados na elaboração do algoritmo MO-CORS e outros sejam utilizados pelo leitor para o esclarecimento de procedimentos utilizados no mesmo.

2.1. Otimização Mono-Objetivo

Quando um problema de otimização que modela um sistema físico envolve apenas uma função objetivo, o procedimento de encontrar a “melhor” solução é chamado algoritmo de otimização mono-objetivo. A forma matemática de apresentar os problemas de otimização mono-objetivo na forma mais simples é:

$$\min_{\mathbf{x} \in S} f(\mathbf{x}) \quad (2.1)$$

sendo $f: \mathbb{R}^n \rightarrow \mathbb{R}$, a função objetivo que se deseja otimizar e $\mathbf{x} = [x_1, x_2 \dots x_n]^T \in \mathbb{R}^n$, as variáveis de decisão e $S \subseteq \mathbb{R}^n$, é a região de busca ou região viável. Se $S = \mathbb{R}^n$, o problema é chamado sem restrições ou irrestrito, mas se S é um subconjunto de \mathbb{R}^n , o problema é dito com restrições e sua expressão matemática pode ser colocada da seguinte forma:

$$\min_{\mathbf{x} \in S} f(\mathbf{x}) \quad (2.2)$$

Sujeito a:

$$g_m(\mathbf{x}) \leq 0 \quad m = 1, 2, 3 \dots m_T \quad (2.3)$$

$$h_l(\mathbf{x}) = 0 \quad l = 1, 2, 3 \dots l_T \quad (2.4)$$

$$\mathbf{x}_{LI} \leq \mathbf{x} \leq \mathbf{x}_{LS} \quad (2.5)$$

onde a região viável S do problema de otimização está formada pelas restrições de desigualdade $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$, as restrições de igualdade $h: \mathbb{R}^n \rightarrow \mathbb{R}^l$, e os limites laterais inferior e superior \mathbf{x}_{LI} e \mathbf{x}_{LS} (também chamadas restrições laterais).

2.1.1. Otimização local e global em problemas mono-objetivo

Nos problemas de otimização, a solução procurada pode ser local ou global, atendendo, segundo seja o caso, às seguintes definições.

Definição 1: segundo Sun e Yuan (2006) \mathbf{x}^* é definido como *ótimo local* se existe um $\varepsilon > 0$, tal que $f(\mathbf{x}^*) \leq f(\mathbf{x})$, para todo $\mathbf{x} \in S$, satisfazendo $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$. Da mesma forma, \mathbf{x}^* é definido como *ótimo local estrito*, se existe um $\varepsilon > 0$, tal que $f(\mathbf{x}^*) < f(\mathbf{x})$, para todo $\mathbf{x} \in S$ com $\mathbf{x} \neq \mathbf{x}^*$ satisfazendo $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$.

Definição 2: \mathbf{x}^* é definido como *ótimo global* se $f(\mathbf{x}^*) \leq f(\mathbf{x})$, para todo $\mathbf{x} \in S$. Da mesma forma \mathbf{x}^* é definido como *ótimo global estrito* se $f(\mathbf{x}^*) < f(\mathbf{x})$, para todo $\mathbf{x} \in S$, $\mathbf{x} \neq \mathbf{x}^*$.

Na Figura 2.1 ilustram-se, como exemplo, as definições 1 e 2, onde $(P_1$ e $P_2)$ são ótimos locais estritos, (P_3) é um mínimo local, (P_5) é mínimo global estrito e $(P_4$ e $P_6)$ são máximos globais.

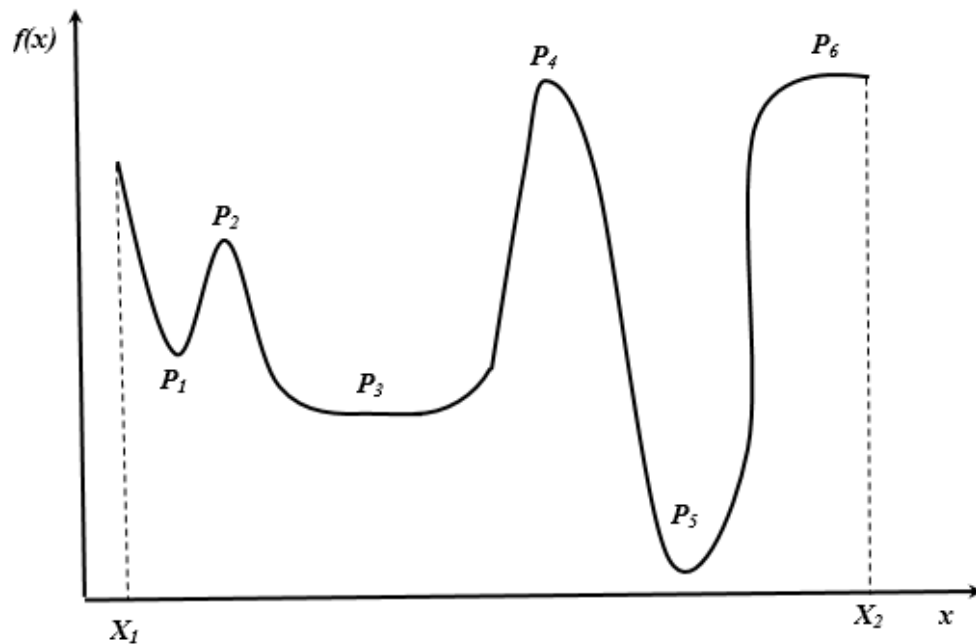


Figura 2.1: Mínimos e máximos locais e globais.

Em geral, o interesse dos projetistas e pesquisadores na área de engenharia concentra-se na busca de ótimos globais, que costuma não ser uma tarefa simples para os algoritmos de otimização: durante uma iteração, só se tem conhecimento dos valores nos pontos avaliados, e que as estratégias de busca dos pontos das seguintes avaliações, não fornecem garantias de que um ótimo encontrado seja de fato global. Neste sentido a complexidade da estratégia usada pelo algoritmo incrementa a possibilidade de obtenção de um ótimo global, mas isso traz um aumento no custo computacional.

Outro aspecto importante a ser considerado dentro do planejamento das estratégias de busca dos problemas de otimização, é o conhecimento da convexidade da função, por exemplo, quando a função objetivo é convexa, demonstra-se que o mínimo local é também o mínimo global da função. (SOUZA e DINIZ-EHRHARDT, 2011).

2.1.2. Convexidade e não convexidade em otimização mono-objetivo

Um conjunto $S \subseteq \mathbb{R}^n$, é dito convexo, se o segmento de reta entre quaisquer dois pontos em S está contido em S isto é, se para quaisquer $[\mathbf{x}, \mathbf{y}] \in S$ e $\theta \in [0,1]$, verifica-se que $\theta\mathbf{x} + (1 - \theta)\mathbf{y} \in S$, conforme ilustrado na Figura 2.2.

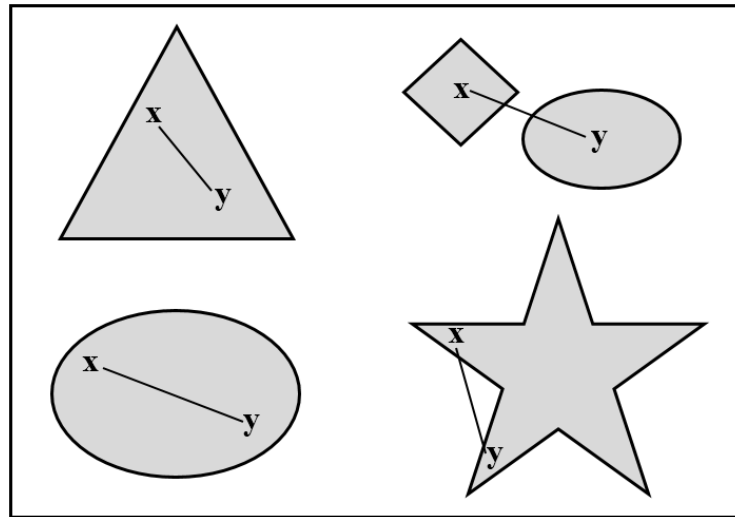


Figura 2.2: Conjuntos convexos (esquerda) e não convexos (direita).

Segundo Souza e Diniz-Ehrhardt (2011) a definição matemática de uma função convexa no espaço n -dimensional é:

Definição 3: Seja S um subconjunto convexo de \mathbb{R}^n . Uma função $f: S \rightarrow \mathbb{R}$ é dita convexa se: $f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$, $\forall x, y \in S$, $\forall \theta \in [0,1]$.

2.1.3. Otimização robusta

É dito que o valor ótimo de um processo de otimização é robusto quando a variância do valor da função objetivo é minimizada, ou seja, alterações aleatórias nas variáveis e/ou parâmetros do problema alteram pouco o valor da função objetivo desejada, (RENÓ, 2007). Nestes problemas, chamados de otimização robusta, a análise da variância das variáveis de projeto pode ser usada para reduzir o número de variáveis a serem consideradas como aleatórias, simplificando a obtenção da variância da função objetivo. Além disso, esta análise de variância das variáveis de projeto também pode ser usada para pesquisar, se num problema dado é significativa a inclusão da própria variância como uma função objetivo adicional, transformando um problema mono-objetivo em multi-objetivo. Os pontos P_3 e P_6 ilustrados na Figura 2.1, são exemplos gráficos de soluções robustas num problema de otimização.

2.1.4. Taxa de convergência

A taxa de convergência (γ) é uma maneira de medir a eficácia de um método de otimização. Assim, o comportamento de um algoritmo de otimização é considerado aceitável quando, numa determinada iteração t , o ponto $x^{(t)}$ se move no sentido de um mínimo local x^* ,

e rapidamente converge para o mesmo ponto nas sucessivas iterações. Sendo assim, Sun e Yuan (2006), definem matematicamente a taxa de convergência da seguinte forma:

Seja $\mathbf{x}^{(t)}$, $t = 1, 2, 3, \dots, N$, uma sequência de pontos obtidos iterativamente por um algoritmo de otimização. Então, o algoritmo converge para \mathbf{x}^* , se:

$$\lim_{t \rightarrow \infty} \|\mathbf{x}^{(t)} - \mathbf{x}^*\| = 0 \quad (2.6)$$

A Equação 2.6 apresenta a definição clássica de convergência a um ótimo \mathbf{x}^* , mas é muito mais prático nos problemas mono-objetivo estabelecer a convergência normalizada γ entre duas aproximações \mathbf{x}_{t-1}^* , \mathbf{x}_t^* , obtidas em iterações contíguas:

$$\lim_{t \rightarrow \infty} \frac{\|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|}{\|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|^p} = \gamma \quad (2.7)$$

Quando $p = 2$ e $\gamma \in [0, 1]$, é dito que a sequência, $\mathbf{x}^{(t)}$ converge *quadraticamente*

2.2. Otimização Multi-Objetivo

Existem muitas variedades de problemas reais em que surgem dois ou mais objetivos conflitantes. Esses tipos de problemas são conhecidos como problemas multi-objetivo, cujas soluções são aquelas que têm a melhor solução de compromisso possível entre cada um dos objetivos envolvidos (COELLO, 2011). Portanto, nos problemas de otimização multi-objetivo, é importante encontrar o conjunto de melhores soluções relacionadas possíveis através do conceito de dominância. Em geral, os Problemas de Otimização Multi-Objetivo (em inglês: *Multi-Objective Optimization Problem* - MOOP) ou também chamados na literatura de problemas de otimização multi-desempenho ou de multicritérios (OSYCZKA, 1985), buscam minimizar ou maximizar simultaneamente os valores de todas as funções objetivos, no intuito de encontrar um conjunto de soluções ótimas (não dominadas), denominadas frente de Pareto.

A forma matemática mais simples de apresentar os problemas de otimização multi-objetivo é:

$$\min_{\mathbf{x} \in S} f_k(\mathbf{x}) \text{ ou } \max_{\mathbf{x} \in S} f_k(\mathbf{x}) : k = 1, 2, 3 \dots K \quad (2.8)$$

sendo $f_k: \mathbb{R}^n \rightarrow \mathbb{R}$ as funções objetivo que se desejam otimizar, $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ é o

vetor de decisão que contém as variáveis de decisão no espaço $S \subseteq \mathbb{R}^n$. Assim como nos problemas mono-objetivo, se $S = \mathbb{R}^n$, o problema é chamado sem restrições ou irrestrito, mas se S é um subconjunto de \mathbb{R}^n , o problema é dito com restrições e sua expressão matemática é:

$$\min_{x \in S} f_k(\mathbf{x}) \text{ ou } \max_{x \in S} f_k(\mathbf{x}) : k = 1, 2, 3 \dots K \quad (2.9)$$

Sujeito a:

$$g_m(\mathbf{x}) \leq 0 \quad m = 1, 2, 3 \dots m_T \quad (2.10)$$

$$h_l(\mathbf{x}) = 0 \quad l = 1, 2, 3 \dots l_T \quad (2.11)$$

$$\mathbf{x}_{LI} \leq \mathbf{x} \leq \mathbf{x}_{LS} \quad (2.12)$$

onde $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$, são as restrições de desigualdade, $f: \mathbb{R}^n \rightarrow \mathbb{R}^l$ são as restrições de igualdade e \mathbf{x}_{LI} e \mathbf{x}_{LS} representam os limites laterais inferior e superior de cada variável (como no problema mono-objetivo).

2.2.1. Espaço de decisão e espaço objetivo

Da mesma forma que nos problemas mono-objetivo, nos problemas de otimização multi-objetivo, cada vetor de decisão está contido num espaço n-dimensional cujas coordenadas são as variáveis de projeto do MOOP. Esse espaço é chamado *espaço de decisão* (S), como é expresso na Equação 2.13.

$$S = \{ \mathbf{x}_j = [x_1, x_2 \dots x_{n_{var}}]^T / \mathbf{x}_j \in \mathbb{R}^{n_{var}} \} \quad (2.13)$$

A característica principal dos MOOP é que estes possuem mais de uma função objetivo onde serão avaliadas as variáveis de decisão. Denotando as funções objetivo como $f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), \dots f_k(\mathbf{x})$, (onde k é o contador de funções objetivo), O vetor de objetivos \mathbf{f} pode ser expresso como $\mathbf{f} = [f_1(\mathbf{x}) \dots f_k(\mathbf{x})]^T \in \mathbb{R}^k$. Então o espaço objetivo (Ω) é o conjunto de pontos $\Omega \subseteq \mathbb{R}^k$, resultantes da avaliação dos pontos do espaço de decisão em cada uma das k funções objetivo. Como é expresso na Equação 2.14.

$$\Omega = \{ \mathbf{f}_k = [f_1(\mathbf{x}), \dots f_k(\mathbf{x})]^T / \mathbf{f}_k \in \mathbb{R}^k \} \quad (2.14)$$

Na Figura 2.3, ilustram-se, nos casos bidimensionais (sem perder a generalidade teórica), as definições de espaço de decisão e do espaço objetivo, segundo Coello *et al.* (2007).

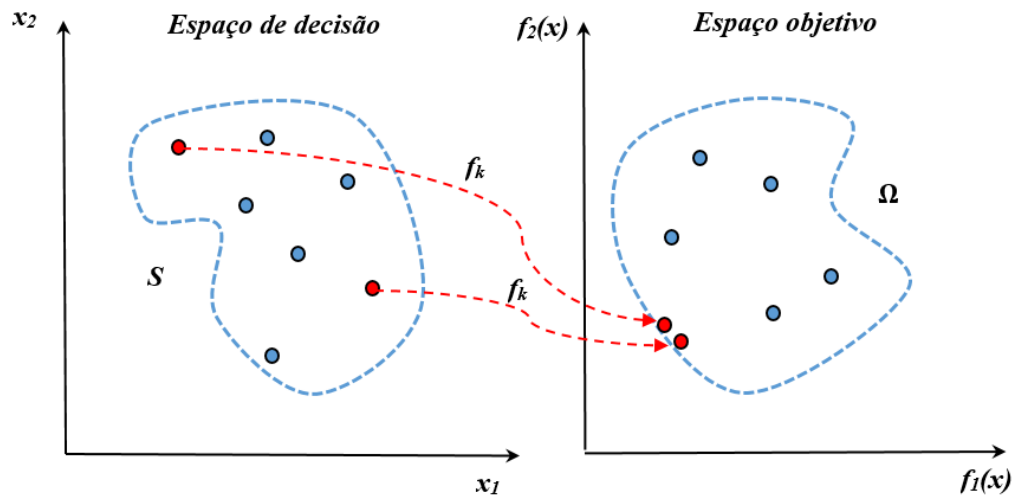


Figura 2.3: Espaço de decisão e espaço objetivo.

Embora o espaço objetivo seja dependente do espaço de decisão, é possível que dois ou mais pontos muito próximos no espaço objetivo, sejam produto da avaliação de dois ou mais pontos muito afastados no espaço de decisão como mostra a Figura 2.3. Isto faz com que nos MOOP, seja necessário o planejamento de ambos os espaços para obter um bom desempenho do algoritmo.

2.2.2. Linearidade e não linearidade em otimização multi-objetivo

A linearidade ou não-linearidade dos problemas de otimização multi-objetivo, é definida com base da linearidade ou não linearidade das funções objetivo e de restrições do problema. Se todas as funções objetivo e as restrições do problema são lineares, então o MOOP é dito *Linear*; caso contrário, se ao menos uma das funções objetivo ou de restrições é não linear então o MOOP é dito *Não Linear* (DEB, 2001).

2.2.3. Convexidade e não convexidade em otimização multi-objetivo

Segundo Deb (2001) um problema de otimização multi-objetivo é convexo se as duas seguintes afirmações são cumpridas:

- Todas as funções objetivo e a região viável são convexas.
- Todas as restrições de desigualdade g são convexas e as restrições de igualdade h são lineares.

Da mesma forma que nos problemas mono-objetivo, nos problemas multi-objetivo a convexidade das funções objetivo apresenta as seguintes propriedades:

- Os valores das funções $f_k(\mathbf{x})$, podem ser estimados usando aproximações lineares.

- A matriz Hessiana de cada $f_k(\mathbf{x})$, é positiva definida para todo \mathbf{x} .
- Um ótimo local da função convexa é sempre um ótimo global.

2.2.4. Vetor objetivo ideal e conceito de dominância

Dois dos conceitos próprios e muito importantes a serem considerados nos problemas multi-objetivo são: o vetor ideal e a dominância entre os pontos solução.

O vetor objetivo ideal é aquele que contém todas as soluções ótimas, mínimos ou máximos, de cada função objetivo.

Definição 4: Seja $\mathbf{x}_k^O = [x_{k,1}^O, \dots, x_{k,n}^O]^T$, o vetor de variáveis que minimiza ou maximiza a k -ésima função objetivo $f_k(\mathbf{x})$, $k = 1, \dots, K$, onde $\mathbf{x}_k^O \in S$. Isso significa que $f_k(\mathbf{x}_k^O) = \max/\min_{\mathbf{x} \in S} f_k(\mathbf{x})$, é a k -ésima coordenada do vetor \mathbf{f}^* denominado *vetor objetivo ideal*.

$$\mathbf{f}^* = [f_1^O, \dots, f_k^O]^T \in \mathbb{R}^k \quad (2.15)$$

Na maioria dos casos, nos problemas multi-objetivo é muito pouco provável que o resultado final seja um vetor objetivo ideal. Assim, é necessário introduzir nestes problemas um conceito que permita a comparação de vetores no espaço objetivo, no intuito de estabelecer o conjunto de melhores soluções do problema. Trata-se do conceito de *dominância* entre soluções.

O conceito de *dominância* é usado nos algoritmos de otimização multi-objetivo (em inglês, *Multi-Objective Optimization Algorithm's*, MOOAs), de forma equivalente ao uso das relações “maior que” e “menor que”, nos algoritmos mono-objetivo, no intuito de comparar quando uma solução é melhor ou pior que outra, já que no caso dos MOOPs o espaço objetivo é multidimensional. e não unidimensional. Esse conceito pode ser expresso na definição 5, segundo Deb (2001) (apenas minimização).

Definição 5: Uma solução \mathbf{x}_i *domina* outra solução \mathbf{x}_j , ($\mathbf{x}_i < \mathbf{x}_j$), se as seguintes condições são satisfeitas:

- A solução \mathbf{x}_i , não é pior que a solução \mathbf{x}_j para todas as funções objetivo.

$$\forall k \in \{1, \dots, K\}: f_k(\mathbf{x}_i) \leq f_k(\mathbf{x}_j) \quad (2.16)$$

- A solução de x_i , é estritamente melhor que a solução de x_j em ao menos uma das funções objetivo.

$$\exists k \in \{1, \dots, K\}: f_k(x_i) < f_k(x_j) \quad (2.17)$$

Caso alguma dessas condições não seja satisfeita, dir-se-á que a solução x_i *não domina* a solução x_j , ou que a solução x_j *não é dominada* pela solução x_i .

2.2.5. Conjunto não dominado e frente de Pareto

Os conceitos de conjunto dominado e frente de Pareto são cruciais em problemas multi-objetivo. As seguintes definições tratam deles, com base em Deb (2001):

Definição 6: Dentre um conjunto de soluções P , o *conjunto não dominado* de soluções P' é aquele formado pelos membros não dominados por nenhum membro do conjunto P .

Definição 7: O conjunto não dominado de todo o espaço de decisão viável S é denominado *conjunto ótimo de Pareto global*. Sua imagem no espaço objetivo é denominada *frente de Pareto (global)*.

Definições alternativas de frente de Pareto global aparecem na literatura. Como exemplo, segue a definição de, Reyes e Coello (2006): se $f_k(x_i^*) \in \mathcal{F}^*$, onde P^* é o conjunto de soluções não dominadas por outra solução em todo o espaço objetivo Ω , o conjunto de vetores \mathcal{F}^* é chamado *frente de Pareto* e o conjunto de suas respectivas contra imagens x_i^* é chamado de *ótimo de Pareto*.

$$\mathcal{F}^* = \{f_k(x_i^*) \in \mathbb{R}^K / f_k(x_i^*) < f_k(x_i) \forall f_k(x_i) \in \Omega\} \quad (2.18)$$

A definição de frente de Pareto local também é relevante (DEB, 2001):

Definição 8: Se para cada membro x de um conjunto $P^* \subset S$, não existe solução y que domine qualquer membro do conjunto P^* (na vizinhança de x , de modo que $\|y - x\| < \varepsilon$, sendo ε um número positivo arbitrariamente pequeno), então P^* é denominado um *conjunto ótimo de Pareto local*. Sua imagem no espaço objetivo é denominada uma *frente de Pareto local*.

As definições de conjunto ótimo de Pareto global e local ótimo dos problemas multi-objetivos de certa forma estendem as definições de ótimo global e local dos problemas mono-objetivos.

Baseados nos conceitos de dominância acima apresentados, alguns MOOAs são orientados, ao longo do processo de otimização, pela busca iterativa de um conjunto não dominado de pontos (soluções) o mais representativo possível. Essa busca naturalmente é discreta. Em cada iteração i do processo, são aplicados os critérios da *definição 5* entre todos os pontos-solução avaliados e selecionados para busca. São então obtidos os pontos pertencentes ao *conjunto não dominado* da i -ésima iteração. O algoritmo então opera no sentido de selecionar e avaliar novos pontos, de modo a intensificar a busca na próxima iteração. Assim sucessivamente então, o conjunto não dominado resultante da última iteração prevista – ou quando se atinja um critério de convergência adotado – corresponderá a uma aproximação final do conjunto ótimo global, cuja imagem no espaço objetivo será então uma aproximação da *frente de Pareto* global do problema.

Na Figura 2.4, ilustram-se para duas funções objetivo f_1 e f_2 , as representações das imagens dos conjuntos não dominados P_1^* e P_2^* , e a frente de Pareto \mathcal{F} no espaço objetivo Ω .

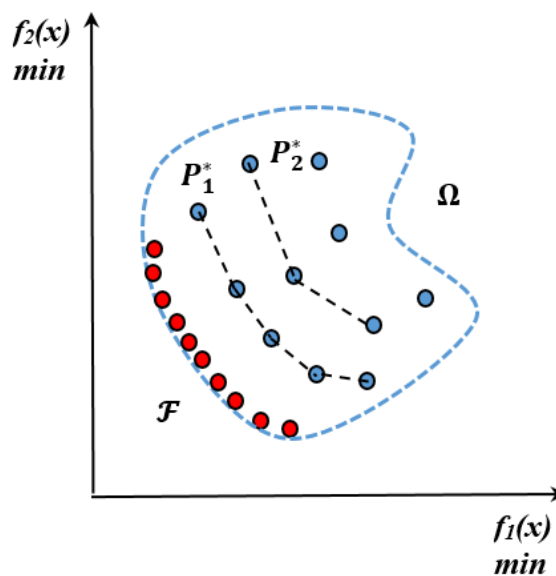


Figura 2.4: Frente de Pareto e conjuntos não dominados durante a evolução.

Devido à característica de multiplicidade de funções dos problemas multi-objetivos, estes não costumam ser caracterizados somente como uma maximização ou minimização, senão como uma combinação de minimização e maximização em cada função objetivo. Por exemplo, se o problema tem apenas duas funções objetivos, têm-se 4 possíveis combinações: *min-min*, *max-max*, *min-max* e *max-min*. Na Figura 2.5, ilustram-se as frentes de Pareto de cada uma destas combinações.

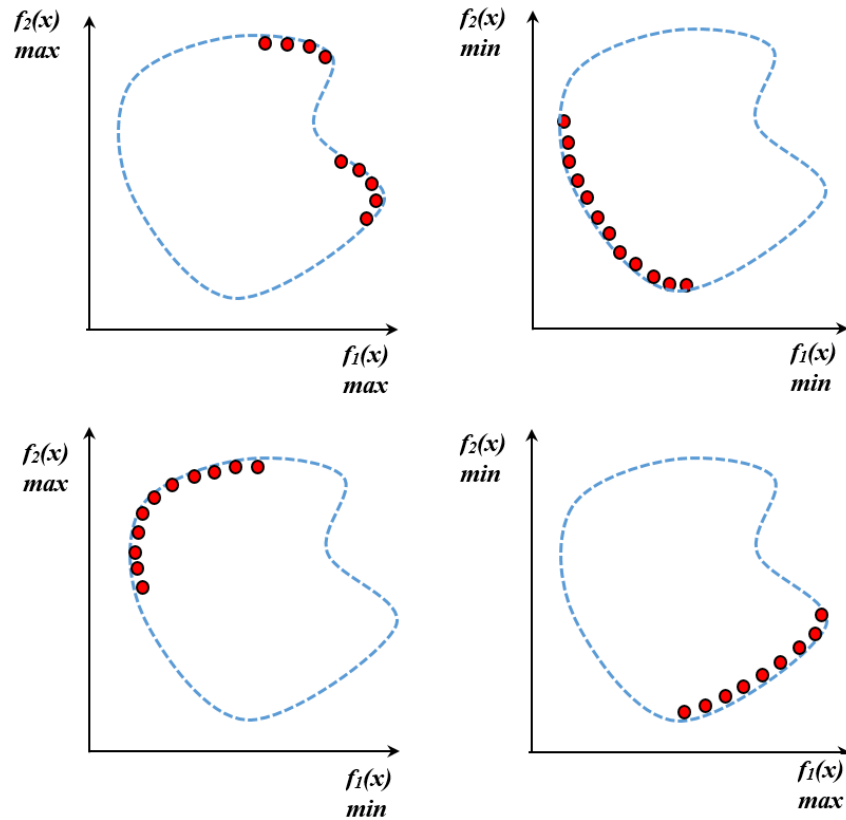


Figura 2.5: Combinações de frente de Pareto para dois objetivos.

2.2.6. Algoritmos para identificação de conjuntos não dominados

Encontrar as soluções não-dominadas de um dado conjunto é similar ao princípio de encontrar o valor mínimo de um conjunto de número reais. No caso da determinação do valor mínimo em um conjunto dos números reais, dois números são comparados para verificar qual é o menor e, desta forma, o operador “<” (menor) é usado para fazer esta verificação. No entanto, para encontrar um conjunto não-dominado a relação de dominância “<”, vista anteriormente na definição 5, pode ser usada para identificar a melhor de duas soluções. Assim como, existem diferentes procedimentos para encontrar o valor mínimo de um conjunto finito de números reais, encontrar um conjunto de soluções não-dominadas de um dado conjunto não é diferente (DEB, 2001). Os procedimentos para determinar um conjunto não-dominado geralmente possuem diferentes complexidades computacionais e dentre os métodos mais utilizados estão: (i) Suave e lento, (ii) atualização contínua e (iii) o método eficiente de Kung.

Suave e lento: neste método cada solução do espaço objetivo é comparada com o restante das soluções em cada passo. Cada solução não é considerada pertencente ao conjunto não dominado, enquanto ela é dominada por ao menos uma das soluções restantes (DEB, 2001). A vantagem desta abordagem é que possui uma estrutura simples de fácil implementação, mas com a desvantagem que sua aplicação implica um alto custo computacional.

Atualização contínua: Neste método, todos os pontos a serem analisados são colocados dentro do conjunto não dominado. Em cada iteração, é verificada a dominância de cada ponto respeito dos outros. No instante que seja verificado que um ponto é dominado por algum outro, atualiza-se o conjunto de soluções não dominadas através da eliminação do ponto dominado. Essa abordagem é mais rápida do que a abordagem suave e lenta por causa da melhor contabilidade das soluções em cada iteração. (DEB, 2001).

Método de Kung: Esta abordagem inicialmente elaborada por Kung *et al.* (1975) e apresentada em Deb (2001), propõe uma divisão recursiva do conjunto de soluções até atingir o conjunto não dominado, seguindo o procedimento seguinte:

Passo 1: ordena-se de melhor até pior o conjunto de soluções do espaço objetivo. Este conjunto é inicializado como conjunto de soluções não dominadas P^* .

Passo 2: O segundo passo consiste em dividir o conjunto total em 2 subconjuntos do mesmo tamanho (melhores, E e piores D), colocados na esquerda e na direita em cada subdivisão. Isto é feito através de uma divisão recursiva do conjunto P^* até que cada subconjunto E e D tenham unicamente um elemento.

Passo 3: Verifica-se a dominância dos elementos do conjunto E respeito de D . Eliminam-se os pontos dominados em cada nível depois de cada verificação.

Esta abordagem é a mais eficiente em termos de tempo computacional, mas a complexidade de codificação é mais alta.

Método de atualização contínua (*Continuously Update*).

Dentre os métodos citados, o método de atualização contínua se situa em posição intermediária em termos de nível de desempenho e complexidade de codificação: é computacionalmente menos custoso e com estrutura mais complexa que o método Lento e Suave, e computacionalmente mais custoso, mas com uma complexidade computacional bem menor que o método de Kung. Assim sendo, este método foi eleito para ser aplicado nesta pesquisa na determinação dos conjuntos não dominados.

O método de atualização contínua é muito similar ao Lento e Suave, exceto pelo uso de uma melhor estratégia de contabilização que faz este método computacionalmente mais rápido. Aqui, a dominância de cada solução do conjunto é verificada com respeito a outras no mesmo conjunto, como se explica a seguir: para começar, todos os N_p pontos do espaço objetivo Ω , são atribuídos ao conjunto de pontos não dominados $P' = \Omega$, o qual representa a primeira solução. Depois, se a solução Ω_i domina algum ponto P'_j , ele é removido do conjunto P' . Dessa forma, nenhum ponto da solução não dominada é removido de P' . Caso contrário, se a solução Ω_i for dominada por algum ponto P'_j , a solução Ω_i será ignorada. Quando todas as soluções do

conjunto Ω forem verificadas, os membros restantes de P' constituem o conjunto não dominado P^* , (DEB, 2001).

2.3. Abordagem dos Problemas de Otimização Multi-Objetivo

Devido à complexidade no tratamento de dois espaços possivelmente multidimensionais nos problemas de otimização multi-objetivo, é preciso estabelecer um método de abordagem que se ajuste ao problema estudado. Na literatura encontram-se dois tipos de métodos de abordagem, os métodos clássicos e os métodos diretos.

2.3.1. Métodos Clássicos

Estes métodos são baseados no uso de informações do problema obtidas a priori para formar uma única função objetivo, que pode ser otimizada através do uso de algoritmos de otimização mono-objetivo. Dentre estes destaca-se, por exemplo, o método das somas ponderadas das funções objetivos, onde cada peso seria proporcional ao nível de preferência atribuído à respectiva função objetivo. Nesse caso, o problema de otimização multi-objetivo seria então, reduzido a um problema de otimização mono-objetivo e a abordagem seguiria a sequência mostrada na Figura 2.6.

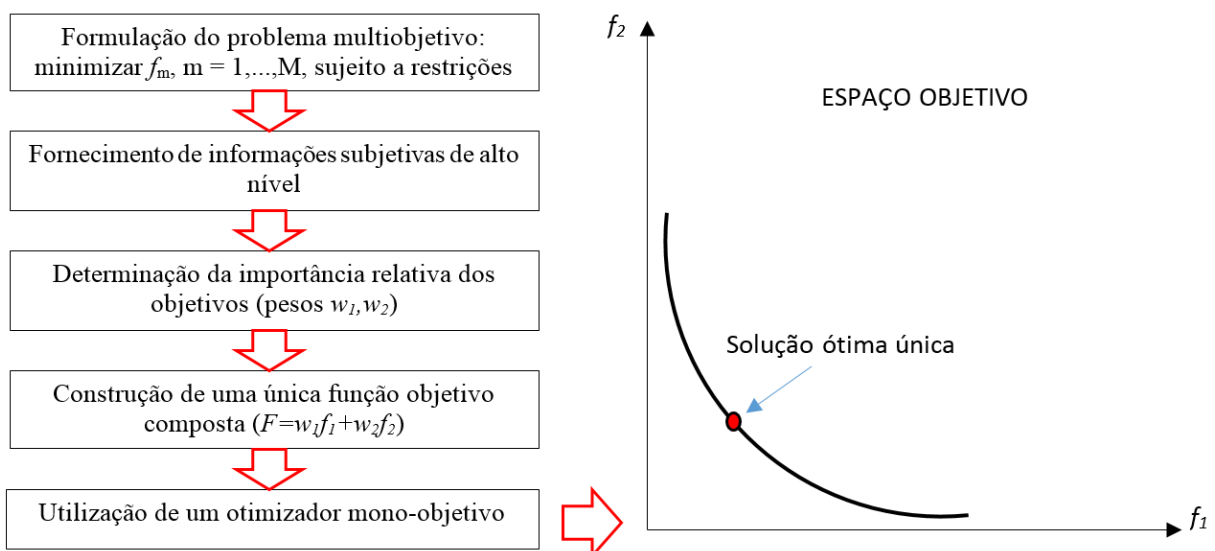


Figura 2.6: Abordagem clássica dos algoritmos de otimização multi-objetivo.

O método das somas ponderadas (DEB, 2001) pode ser formulado da seguinte maneira:

$$\text{Seja } \min_{x \in S} f_k(x) \quad k = 1, \dots, K \quad (2.19)$$

$$\text{Então } F(x) = \sum_{k=1}^K w_k f_k(x) \quad (2.20)$$

Sujeito a:

$$g_m(x) \leq 0 \quad m = 1, 2, 3 \dots m_T \quad (2.21)$$

$$h_l(x) = 0 \quad l = 1, 2, 3 \dots l_T \quad (2.22)$$

$$x_{LI} \leq x \leq x_{LS} \quad (2.23)$$

onde w_k representa o peso de cada função objetivo $f_k(x)$ e $F(x)$, é a função objetivo ponderada única. Embora muito utilizado, uma das grandes desvantagens do método das somas ponderadas é sua incapacidade de lidar com espaço objetivo e trechos da frente de Pareto não convexa.

Outro método clássico, bastante conhecido, é o *Método da Restrição- ε* , que consiste em converter todas as funções objetivo, menos uma, em restrições de ε_i níveis, onde a função restante continuará sendo minimizada. Variando-se os valores ε_i dos níveis das restrições. Qualquer região da frente de Pareto poderá ser acessada, tendo-se então transformado o problema multi-objetivo em diversos problemas com um único objetivo. O método da restrição- ε foi desenvolvido com a finalidade de diminuir as dificuldades que tem o método das somas ponderadas quando o espaço objetivo é não convexo. A representação matemática deste método é apresentado a seguir no caso de minimização.

$$\text{Seja } \min_{x \in S} f_k(x) \quad k = 1, 2, 3 \dots K \quad (2.24)$$

$$\text{Então } \min f_\mu(x) \quad \mu \in [1, K] \quad (2.25)$$

Sujeito a:

$$f_\theta(x) \leq \varepsilon_\theta \quad \forall \theta \in [1, K] \mid \theta \neq \mu \quad (2.26)$$

$$g_m(x) \leq 0 \quad m = 1, 2, 3 \dots m_T \quad (2.27)$$

$$h_l(x) = 0 \quad l = 1, 2, 3 \dots l_T \quad (2.28)$$

$$x_{LI} \leq x \leq x_{LS} \quad (2.29)$$

A principal desvantagem deste método é a dependência dos valores do vetor ε_θ os quais devem de ser estabelecidos pelo projetista.

Cabe ressaltar que a principal desvantagem dos métodos clássicos, também chamados de métodos “a priori”, é a grande dificuldade em se estabelecer o nível relativo de preferência entre os diversos objetivos. Por exemplo, no método de somas ponderadas seria necessário realizar mais de um processo de otimização, variando-se os pesos atribuídos às funções objetivo, na esperança de se obter mais informações sobre o problema que está sendo analisado. Observa-se, porém, que os resultados do problema de otimização podem ser altamente sensíveis

aos valores dos pesos. Muitas regiões do espaço objetivo podem inclusive ficar inacessíveis para qualquer combinação possível dos valores dos pesos atribuídos às funções objetivo.

2.3.2. Métodos Diretos

Para evitar as dificuldades apresentadas pelos métodos clássicos, foi desenvolvida outra forma de abordagem dos problemas de otimização multi-objetivo, chamada abordagem direta. Neste caso, um procedimento considerado ideal para abordagem do problema de otimização multi-objetivo seria a obtenção direta da frente de Pareto ótima, conforme a sequência apresentada na Figura 2.7.

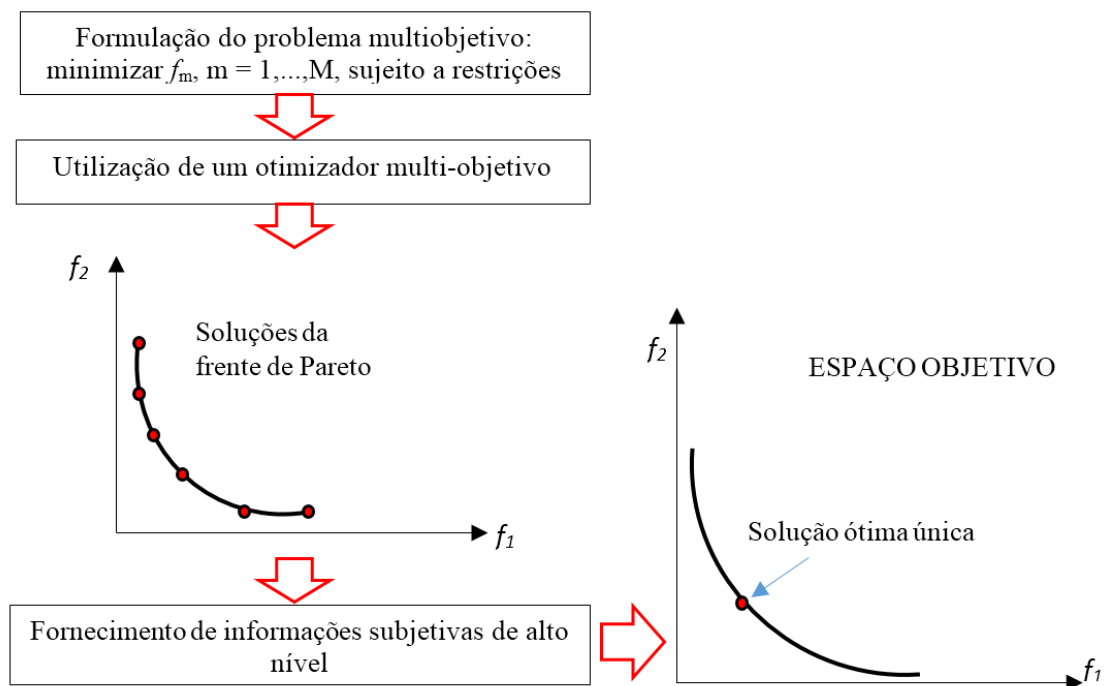


Figura 2.7: Abordagem direta dos algoritmos de otimização multi-objetivo.

Considerando que a abordagem utilizada neste trabalho é direta, i.e., baseada na busca da frente de Pareto, os algoritmos que serão tratados com uma maior profundidade situam-se no contexto de otimizador multi-objetivo.

2.4. Algoritmos Evolutivos e Populacionais

O termo “algoritmo evolutivo” é usado para indicar qualquer algoritmo de otimização baseado numa população e que utilize operadores inspirados em mecanismos biológicos de evolução, tais como a seleção, reprodução e mutação, entre outros. Os indivíduos da população são os potenciais candidatos à resolução do problema de otimização, sendo as regras da

evolução cumpridas pelos indivíduos através da sua avaliação numa função objetivo. Estes algoritmos vêm sendo muito utilizados em diferentes áreas da engenharia, principalmente devido a sua capacidade de encontrar boas soluções.

Embora os algoritmos evolutivos sejam populacionais, nem todos os algoritmos populacionais são evolutivos; alguns deles, por exemplo, são baseados em heurísticas de processos físicos, mecânicos e de tratamento de materiais. Os algoritmos evolutivos procuram utilizar conceitos e princípios da evolução natural das espécies como estratégia de otimização de problemas. A aplicação desses conceitos aos problemas de otimização multi-objetivo pode ser vista como uma extensão a partir do entendimento do funcionamento dos mesmos na abordagem de problemas mono-objetivo.

Para a busca de soluções ótimas em problemas mono-objetivo, existem à disposição métodos numéricos tradicionais como o de gradiente conjugado e outros, que têm algumas características fundamentais. Uma delas é trabalhar com regra determinística na transição de uma solução a outra durante a busca ponto-a-ponto. Para obterem maior eficiência, eles utilizam os gradientes da função objetivo. Embora os métodos numéricos tradicionais sejam muito eficientes para problemas com muitas variáveis, estes podem não ser eficientes em problemas com regiões com descontinuidade ou não-diferenciáveis ou ainda com variáveis discretas, o que pode requerer uma implementação para cada tipo de problema. A convergência pode depender da solução inicial escolhida e o ponto ótimo encontrado pode ser apenas um ponto ótimo local. Os algoritmos evolutivos conseguem contornar essas desvantagens. As regras de transição entre soluções são definidas por operações do tipo genéticas, que são implementadas visando a busca do ponto ótimo global. Uma vantagem adicional dos algoritmos evolutivos em relação aos métodos numéricos clássicos é que podem utilizar as facilidades da computação paralela para avaliação das funções objetivo. Na verdade, isso se torna uma necessidade quando se trabalha com funções que requerem um alto custo computacional, que serão avaliadas para cada indivíduo de cada geração de soluções indicadas pelo próprio algoritmo de otimização.

O método evolutivo mais importante dos últimos tempos é o “*Algoritmo Genético*”, que é baseado nos mecanismos de seleção natural e envolve a sobrevivência do indivíduo mais apto. Ele é especialmente atrativo por não exigir que se saiba como encontrar uma solução ótima para um problema, mas sim como reconhecê-la como ótima. Por esse motivo estes algoritmos vêm sendo muito utilizados nas últimas décadas como métodos de busca e de otimização em vários domínios.

As atividades básicas presentes na resolução de um problema de otimização baseado em algoritmo genético, são as seguintes:

- Criar uma representação da solução do problema através de um indivíduo (cromossomo).
- Definir como a avaliação dessa solução será realizada (função de aptidão).
- Determinar como a população inicial será gerada (na maioria dos Algoritmos Genéticos, sua inicialização é feita de forma aleatória).
- Determinar quais operadores genéticos serão aplicados nas soluções. Em geral são operadores de seleção, reprodução, cruzamento e mutação.
- Definir um conjunto de parâmetros, tais como tamanho da população, número de gerações e as probabilidades de aplicação dos operadores.

Na Figura 2.8 apresenta-se o diagrama de fluxo do algoritmo genético simples.

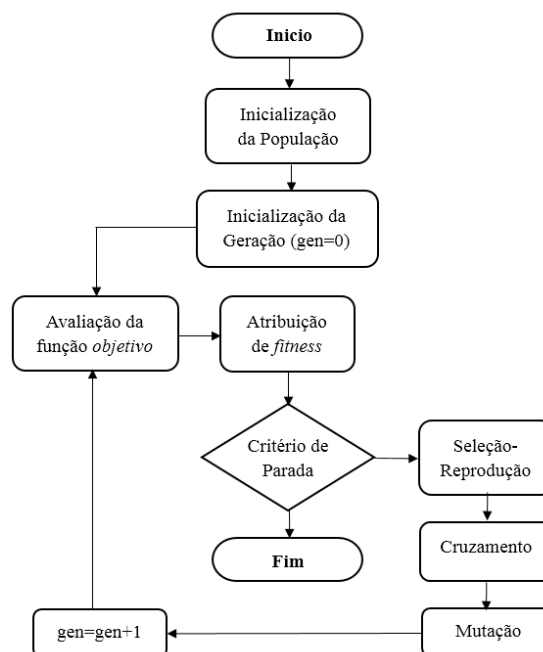


Figura 2.8: Diagrama de fluxo simples de algoritmo genético.

A seguir são apresentados os conceitos mais importantes referentes ao algoritmo genético, incluindo aqueles associados a problemas multi-objetivo, segundo Pappa (2002).

A. Indivíduo.

Os indivíduos representam uma solução candidata à resolução do problema em questão. É uma boa prática encontrar a melhor representação para uma solução candidata de acordo com o problema.

Dentro das formas mais comumente utilizadas para a representação dos indivíduos está a binária de tamanho fixo, em que um indivíduo é uma cadeia de bits que assumem valores 0 ou 1. Porém, essa representação torna-se problemática quando as variáveis a serem representadas

assumem valores contínuos. Em casos em que a representação binária não é a mais natural nem a mais apropriada, outros tipos de representação podem ser utilizados, como sugerido por Michalewicz (2013).

B. Função de aptidão (*fitness*).

A função de aptidão é utilizada para determinar o quão boa uma solução candidata é para resolução efetiva de um problema, através da atribuição de um valor numérico ponderado pelo resultado obtido da função objetivo para cada indivíduo. Sendo assim, a função de aptidão e a forma de representação do indivíduo são os únicos com relação direta ao domínio do problema.

C. Métodos de seleção.

Uma vez que os algoritmos genéticos baseiam-se no princípio da seleção natural, eles devem ser capazes de identificar os indivíduos mais aptos, para que permaneçam na população durante o processo de evolução, e os mais fracos, para que sejam excluídos do processo. Vários métodos podem ser utilizados para execução dessa tarefa, entre eles, a *seleção proporcional*, a *seleção por ranking* e a *seleção por torneio* (DEB, 2001).

D. Elitismo.

Visando preservar e utilizar as melhores soluções encontradas na geração atual nas próximas gerações, surgiu a estratégia de elitismo. Em sua versão mais simples, ela conserva os N_{elit} melhores indivíduos da população atual, copiando-os para a próxima geração sem nenhuma alteração. Os outros $N - N_{elit}$ indivíduos da população são gerados normalmente, através do método de seleção e posterior aplicação dos operadores genéticos. Assim, as melhores soluções não são apenas passadas de uma geração para outra, mas também participam da criação dos novos membros da nova geração.

E. Operadores genéticos.

Algoritmos genéticos tradicionais são normalmente constituídos de dois operadores: *cruzamento* e *mutação*.

O operador de cruzamento permite a troca de material genético entre dois indivíduos denominados pais, combinando informações de maneira que exista uma probabilidade razoável do que os novos indivíduos produzidos serem melhores que seus pais (HINTERDING, 2000). Outra forma de cruzamento muito utilizada é o cruzamento uniforme (FALKENAUER, 1999). De acordo com esse método, cada gene do cromossomo (indivíduo) pode ser trocado de acordo com uma probabilidade fixa p . Quanto maior o valor de p , maior o número de genes trocados entre dois pais. Na literatura, o valor padrão de p é 0.5.

O operador de mutação tem como objetivo substituir o valor de um gene por outro valor aleatoriamente gerado. No caso do indivíduo ser representado por um binário, ela consiste em escolher aleatoriamente um gene do cromossomo e inverter seu valor de 1 para 0 ou vice-versa. O propósito da mutação é manter a diversidade da população e assegurar que o cromossomo sempre cobrirá uma parte suficientemente grande do espaço de busca (HINTERDING, 2000).

F. Aninhamento (*Niching*).

A técnica de aninhamento (*niching*) consiste na divisão da população em espécies que reúnem indivíduos com características semelhantes, para reduzir a competição por recursos e criar subpopulações estáveis, cada uma delas concentrada em um nicho do espaço de busca. Métodos de *Aninhamento* são conhecidos por sua capacidade de criar e manter populações diversas. Dois tipos de métodos são normalmente utilizados na literatura: compartilhamento (*sharing*) e aglomeração (*crowding*).

Segundo Coello *et al.* (2007) no método de compartilhamento ou *sharing*, o tamanho de uma vizinhança (ou nicho) é controlado através do valor do raio de nicho (σ_{share}), dentro do qual, é preciso contar quantas soluções estão localizadas dentro do mesmo, para assim a aptidão seja diminuída proporcionalmente ao número de indivíduos que compartilham a mesma vizinhança. Isso visa promover a geração de soluções nas regiões menos povoadas do espaço de busca. Por exemplo fazendo uma análise dos pontos vermelhos 1, 2 e 3 na Figura 2.9 a aptidão do ponto 2 é menor que a aptidão do ponto 1, a vez que a aptidão do ponto 1 é menor que a aptidão do ponto 3. O compartilhamento da função de aptidão (*fitness sharing*) de um indivíduo, denominado F'_c , é igual a sua função de avaliação F_a dividida por seu contador de nichos (*niching count*). (Equação 2.30). O contador de nichos é a soma dos valores das funções de compartilhamento (*sh*) entre ele e os demais indivíduos da população (incluindo ele mesmo).

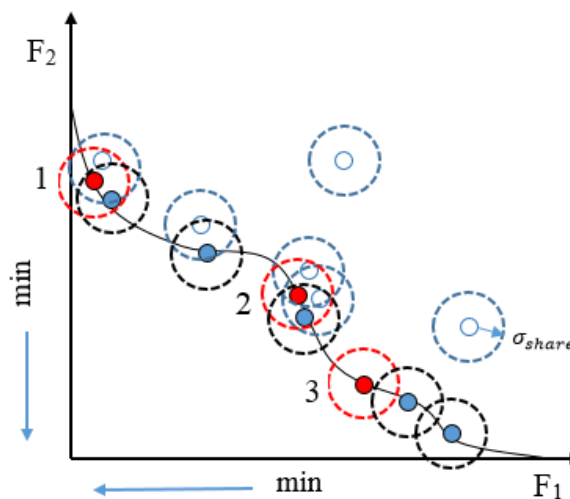


Figura 2.9: Ilustração gráfica de compartilhamento (*sharing*).
Adaptado de Coello *et al.* (2007)

A Equação 2.30 define formalmente o compartilhamento da função de aptidão de um indivíduo i , onde N_{ind} é o número de indivíduos da população.

$$F'_c(i) = \frac{F_a(i)}{\sum_{j=1}^{N_{ind}} sh(d(i,j))} \quad (2.30)$$

A função de compartilhamento é gerada em função de uma distância d entre dois elementos da população, e retorna 1 se os elementos são iguais, 0 se a diferença entre eles é maior que o raio de compartilhamento, e um valor intermediário entre 0 e 1 de acordo com seus níveis de dissimilaridade. Se a distância entre dois elementos da população for maior ou igual a σ_{share} , eles não afetam o compartilhamento da função de aptidão um do outro. Assim,

$$sh(d) = \begin{cases} 1 - (d - \sigma_{share})^\alpha & d < \sigma_{share} \\ 0 & d \geq \sigma_{share} \end{cases} \quad (2.31)$$

onde α é uma constante que regula a forma do compartilhamento da função de aptidão.

No método de aglomeração ou *crowding* são inseridos novos indivíduos na população substituindo indivíduos similares. Da mesma forma que o método de compartilhamento ou *sharing*, ele utiliza uma medida de distância, entre os pontos, para encontrar indivíduos similares. Neste caso, selecionam-se as soluções sobreviventes de acordo com uma métrica de aglomeração da região medida no espaço objetivo da função sobre o próprio conjunto não dominado, por exemplo, na Figura 2.10, para o caso particular de dois objetivos, a distância de aglomeração do ponto (i) , e estabelecida entre os pontos $(i - 1)$ e $(i + 1)$.

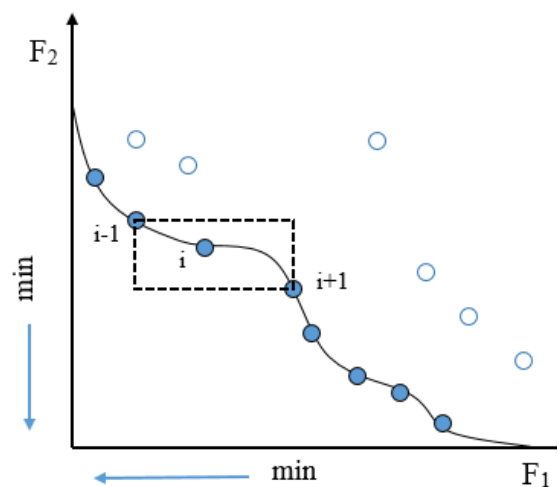


Figura 2.10: Ilustração gráfica de aglomeração (*crowding*). Adaptado de Coello et al. (2007)

A distância de aglomeração mede o perímetro do cuboide formado usando os vizinhos mais próximos como os vértices. Um exemplo é mostrado na Figura 2.10, a distância de aglomeração da solução não dominada é o comprimento médio do cuboide representado pela caixa tracejada. Assumindo que as soluções não dominadas i , $(i - 1)$ e $(i + 1)$, possuem os valores objetivos de $(f_1(i), f_2(i))$, $(f_1(i - 1), f_2(i - 1))$ e $(f_1(i + 1), f_2(i + 1))$ respectivamente, a distância de aglomeração $I[i]_{dis}$ da solução i no conjunto não dominado P^* é dada por (TANG *et al.* 2012):

$$I[i]_{dis} = |f_1(i - 1) - f_1(i)| + |f_1(i + 1) - f_1(i)| + |f_2(i + 1) - f_2(i)| \quad (2.32)$$

O algoritmo para o cálculo da distância de aglomeração é resumido nos seguintes passos:

Passo 1: Inicializar a distância de aglomeração $I[i]_{dis}$, de cada solução não dominada.

Passo 2: Inicializar o contador da função objetivo em, $k = 1$.

Passo 3: Ordenar a população em ordem de pior a melhor para cada função objetivo.

Passo 4: Para cada um dos extremos de cada função objetivo, atribua um valor relativamente grande.

Passo 5: Calcular $I[i]_{dis}$, para cada uma das soluções não dominadas restantes $i = 2, 3, \dots, (s - 1)$ usando a Equação 2.33.

$$I[i]_{dis} = I[i]_{dis} + \frac{I[i + 1]_k - I[i - 1]_k}{f_k^{max} - f_k^{min}} \quad (2.33)$$

em que, s é o número total de soluções no conjunto não-dominado P^* , $I[i]_k$ é o k -ésimo valor da função objetivo da i -ésima solução em P^* , e f_k^{min} e f_k^{max} são os valores mínimo e máximo da k -ésima função objetivo, respectivamente.

Passo 6: Incrementa-se k . Se $k \leq N_{obj}$, onde N_{obj} é o número total de funções objetivo, volta-se para o passo 3.

Um valor pequeno da distância de aglomeração num ponto do conjunto não dominado, implica que a solução não é diversificada próximo desse ponto (AKTHAR e SHOEMAKER, 2016), este critério é utilizado na literatura por alguns algoritmos de otimização multi-objetivo, para determinar o ponto mais isolado no espaço objetivo ou decisão no conjunto não dominado, no intuito de promover buscas globais durante o processo ou obter uma melhor diversidade de pontos na frente de Pareto.

2.4.1. Algoritmos genéticos

Não é novidade o sucesso do uso de algoritmos genéticos em problemas de otimização, incluindo a otimização multi-objetivo. Além das vantagens que os algoritmos genéticos oferecem na resolução de qualquer problema de busca ou otimização, em relação à otimização multi-objetivo, eles apresentam uma vantagem adicional. Nestes algoritmos pode-se perceber que, na busca da solução ótima do problema, uma população de soluções é processada em cada iteração ou geração. Esta característica dos Algoritmos Genéticos ou de outra técnica evolucionária faz com que eles sejam naturalmente adequados para a determinação de várias soluções. Sendo assim, pode-se afirmar que na resolução de problemas de otimização multi-objetivo, os Algoritmos Evolucionários são mais vantajosos em relação, por exemplo, aos métodos clássicos que não foram projetados para trabalharem com múltiplas soluções. O primeiro algoritmo multi-objetivo baseado em AGs só veio a ser implementado por Schaffer (1985), o VEGA (*Vector Evaluated Genetic Algorithm*). Schaffer modificou um algoritmo genético simples para que executasse ciclos independentes de seleção de acordo com cada objetivo. Depois desse primeiro trabalho de Schaffer, pouco avanço se verificaria, até que Goldberg e Holland (1988) esboçassem em dez linhas o que eles chamaram de um procedimento de ordenação (*ranking*) de indivíduos não dominados. Sua sugestão era utilizar o conceito de dominância para deixar várias cópias de indivíduos não dominados em uma população. Para que essas cópias não afetassem a diversidade da população, eles também sugeriram o uso de aninhamento entre pontos de uma classe de soluções não dominadas.

Partindo dessa sugestão de Goldberg e Holland (1988), vários pesquisadores desenvolveram diferentes versões de algoritmos genéticos multi-objetivos. Entre elas, destacam-se o MOGA (*Multi-Objective Genetic Algorithm*) (FONSECA e FLEMING, 1993), o NSGA (*Non-Dominated Sorting Genetic Algorithm*) (SRINIVAS e DEB, 1994), o NPGA (*Niched Pareto Genetic Algorithm*) (HORN *et al.* 1994) e o SPEA (*Strength Pareto genetic Algorithm*) (ZITZLER e THIELE, 1999).

Algoritmo genético multi-objetivo (MOGA)

Fonseca e Fleming (1993) foram pioneiros nessa linha, ao sugerir o MOGA, que é baseado na procura de soluções não dominadas mantendo ao mesmo tempo a diversidade entre elas. Este método é baseado num procedimento de classificação por níveis, ou *ranking* prévio da população. O método é realizado como se descreve a seguir: primeiramente, é feita a classificação por níveis dos indivíduos da população, considerando que, se um indivíduo x_i numa geração t é dominado por $p_i^{(t)}$ indivíduos na população atual, sua posição no *ranking* dos indivíduos é dada por:

$$\text{ranking}(x_i, t) = \begin{cases} 1 & x_i \in P^* \\ 1 + p_i^{(t)} & x_i \notin P^* \end{cases} \quad (2.34)$$

Quer dizer, indivíduos pertencentes ao conjunto não dominado P_i^* recebem *ranking* 1 e o *ranking* dos outros indivíduos é atribuído de acordo com o número de soluções k_d que dominam esses indivíduos, acrescido de um ($k_d = 1 + p_i^{(t)}$).

Depois de fazer o ranking da população, deve-se aplicar uma função de aptidão aos indivíduos. O procedimento pode ser resumido em 3 etapas:

- Ordena-se a população em ordem crescente de acordo com o ranking definido pela Equação 2.34.
- Aplica-se uma função de aptidão que é atribuída a cada solução de acordo com seu ranking. Isto pode ser através de uma função de mapeamento, a qual normalmente é escolhida de forma que os valores de aptidão atribuídos variem entre um número definido, para a solução de melhor ranking e 1 para a solução de pior ranking.
- Por último, é calculada a aptidão das soluções de um mesmo ranking. O procedimento de seleção usa então o valor de aptidão, para selecionar ou eliminar blocos de soluções. Durante a seleção, o procedimento de aninhamento é aplicado, para atingir uma distribuição uniforme de pontos na frente de Pareto.

O pseudocódigo do MOGA é mostrado na Tabela 2.1

Tabela 2.1: Pseudocódigo MOGA. Fonte: Coello *et al.* (2007).

MOGA: Pseudocódigo

- 10: Inicialização da População.
 20: Avaliação das Funções Objetivo.
 30: Atribuição dos *Ranking* Baseado na Dominância de Pareto.
 40: Cálculo do contador de Nichos (*Niching Count*).
 50: Atribuição de Função de aptidão *fitness* Linear.
 60: Atribuição de Função de Compartilhamento *Fitness* (*Sharing Fitness*).
 70: **For** $i=1$ **to** Número de Gerações.
 80: Seleção através de Amostragem Estocástico Universal.
 90: Cruzamento.
 100: Mutação.
 110: Avaliação das Funções Objetivo.
 120: Atribuição dos *Ranking* Baseado na Dominância de Pareto.
 130: Cálculo do contador de Nichos (*Niching Count*).
 140: Atribuição de Função de Avaliação *fitness* Linear.
 150: Atribuição de Função de Compartilhamento *Fitness* (*Sharing Fitness*).
 160: **End** do ciclo.
-

Pode-se observar na Tabela 2.1 que o MOGA utiliza uma função *fitness* de compartilhamento, ou *sharing* para garantir a diversidade da população.

Algoritmo genético multi-objetivo com ordenação de conjuntos não dominado (NSGA II)

Este algoritmo, baseado na ordenação de conjuntos não dominado (em inglês, *Non Dominated Sorting Genetic Algorithm* - NSGA), foi proposto por Srinivas e Deb (1994), como outra variação da abordagem de Goldberg e Holland (1988) para problemas multi-objetivo, depois do MOGA.

O NSGA é um método similar ao MOGA. Suas principais diferenças encontram-se na maneira como a função de aptidão é atribuída ao indivíduo, na estratégia de aninhamento (*niching*) e como se faz o ranking da população, neste sentido cada conjunto não dominado vai ter um valor de *ranking* crescente

$$ranking(x_i, P_q^*) = \begin{cases} 1 & x_i \in P_1^* \\ 1 + ranking(x_i, P_{q-1}^*) & x_i \in P_q^* \forall q > 1 \end{cases} \quad (2.35)$$

onde P_q^* representa qualquer conjunto não-dominado de pontos x_i com $q = 1, 2, 3, \dots, Q$, sendo Q é o número total de conjuntos não dominados da população, e i é o contador de indivíduos em cada conjunto não dominado.

Para garantir a diversidade da população durante o processo, o NSGA utiliza o método de compartilhamento (*sharing*) onde a cada indivíduo é atribuída uma função de aptidão (*shared fitness*). O menor valor da aptidão atribuído é guardado como Sh_{min} . Posteriormente, o segundo *ranking* de não dominados é encontrado, e a cada indivíduo é atribuído um valor de função de aptidão de compartilhamento igual a $(Sh_{min} - v)$, onde v é um número positivo pequeno e calcula-se de novo a função de aptidão. Esse processo é repetido até que as funções de aptidão de compartilhamento de todos os indivíduos tenham sido encontradas. Como resultado, pode-se pensar que esse algoritmo de otimização multi-objetivo converge rapidamente; no entanto, na execução do mecanismo de compartilhamento tem-se um alto custo computacional. O NSGA foi relativamente bem sucedido durante vários anos, embora vários estudos comparativos do tempo indicassem que ele era superado pelo MOGA. O NSGA também era um algoritmo altamente ineficiente devido à maneira como classificava os indivíduos. Neste sentido, Deb *et al.* (2002) criaram uma versão aprimorada, NSGA II com quatro peculiaridades que o distinguem da primeira versão: (i) a ordenação rápida dos não dominados, (ii) O uso da distância de aglomeração ou *crowding*, (iii) a classificação de aptidão é feita pela comparação da aglomeração e (iv) o uso estratégia elitista.

O pseudocódigo pode-se observar na Tabela 2.2

Tabela 2.2: Pseudo Código NSGA II. Fonte: Coello et al. (2007).

NSGA II: Pseudo-Código

10: Inicialização da população \mathbb{P} .
 20: Geração aleatória de uma população de tamanho \mathcal{N}
 30: Avaliação das funções objetivo.
 40: Atribuição dos valores de *Ranking* baseado na dominância de Pareto.
 50: Geração da população de filhos
 60: Método de seleção por torneio (binário)
 70: Recombinação e mutação
 80: **For** $i=1$ **to** número de gerações **do**
 90: **For** cada pai e filho da população **do**
 100: Assinação dos *Ranking* baseado na dominância de Pareto.
 110: Fazer um **loop** interno adicionando soluções na próxima geração começando no primeiro fronte até encontrar \mathcal{N} indivíduos, determinar a distância de aglomeração *Crowding* entre os pontos de cada fronte.
 120: **End For**.
 130: Selecionar os pontos de pior *Ranking* e os fora da distância de *Crowding*
 140: Criar uma nova geração.
 150: Método de seleção por torneio (binário).
 160: Recombinação e mutação.
 170: **End For**.

2.4.2. Algoritmos heurísticos não baseados em algoritmos genéticos.

Existem outros métodos heurísticos de otimização que têm sido adaptados de mono-objetivo a multi-objetivo, e que não são baseados em algoritmos genéticos. Alguns deles apoiam-se no comportamento social de grupos de animais, como as aves, os peixes, ou os insetos, ou em métodos físico-químicos usados para melhorar as propriedades mecânicas dos materiais, dentre outros. Dos mais usados, dentre estes, destacam-se o método de otimização multi-objetivo de enxame de partículas (*Multi-Objective Particle Swarm Optimization* - MOPSO) e o método de otimização multi-objetivo de recozimento simulado (*Multi-Objective Simulated Annealing* - MOSA).

Otimização multi-objetivo baseada em enxame de partículas (MOPSO)

O método populacional de enxame de partículas (em inglês: *Particle Swarm Optimization* - PSO), introduzido por Eberhart e Kennedy (1995) em meados da década de 1990, é baseado no comportamento social animal que segue uma estratégia metaheurística. Ele incorpora a distância, a velocidade e aceleração de cada indivíduo em relação a seu correspondente mais próximo. Uma diversidade de pesquisas tem usado este algoritmo, entre elas, Kumar e Minz

(2014) apresenta um glossário da terminologia e simbologia comum e suas definições seguem para maior clareza no caso da versão multi-objetivo (MOPSO):

- **Enxame (Swarm):** população.
- **Sub-Enxame:** subgrupos ou nichos da população.
- **Partícula (Particle):** um membro individual do enxame.
- **Melhor posição individual (p_{best}):** melhor posição alcançada por uma partícula.
- **Melhor posição local (l_{best}):** posição da melhor partícula dentro de um sub-enxame.
- **Melhor posição global (g_{best}):** posição da melhor partícula do enxame inteiro.
- **Líder:** partícula que guia ao enxame até as melhores regiões no espaço de busca.
- **Velocidade:** direção de movimento de cada partícula para melhorar sua posição.
- **Peso (w):** impacto da velocidade anterior, respeito da atual de cada partícula.
- **Fatores de Aprendizagem:** atração de uma partícula em relação aos seus valores anteriores ou de seus vizinhos. O algoritmo PSO adota dois fatores de aprendizagem: fator de aprendizagem cognitiva (c_1) e fator de aprendizagem social (c_2). O primeiro representa a atração pelo sucesso da própria partícula, e o segundo representa a atração pelos vizinhos. Ambos os fatores são constantes.

Com base nas definições anteriores, listam-se a seguir as equações que definem a topologia do algoritmo, baseada na posição \mathbf{x}_i e a velocidade \mathbf{V}_i da partícula em cada iteração. O parâmetro t representa o tempo ou a iteração do algoritmo

$$\mathbf{x}_i(t + 1) = \mathbf{x}_i(t) + \mathbf{V}_i(t + 1) \quad (2.36)$$

$$\mathbf{V}_i(t + 1) = \mathbf{V}_a + \mathbf{V}_l + \mathbf{V}_g \quad (2.37)$$

onde (\mathbf{V}_a), é a melhor velocidade obtida até o tempo t e as melhores velocidades, local e global (\mathbf{V}_l e \mathbf{V}_g) são calculadas pelas Equações 2.39 e 2.40

$$\mathbf{V}_a = \mathbf{V}_i(t) \quad (2.38)$$

$$\mathbf{V}_l = c_1 \mathbf{R}_{i,1}(t)(pbest_{i,t}(t) - \mathbf{x}_i(t)) \quad (2.39)$$

$$\mathbf{V}_g = c_2 \mathbf{R}_{i,2}(t)(gbest_{i,t}(t) - \mathbf{x}_i(t)) \quad (2.40)$$

$\mathbf{R}_{i,1}$ e $\mathbf{R}_{i,2}$ são vetores de valores aleatórios usados para dar diversidade à população em cada iteração “ t ”. O pseudocódigo é mostrado na Tabela 2.3

Tabela 2.3: Pseudo Código MOPSO. Fonte: Kumar e Minz, (2014).**MOPSO: Pseudo Código**

```

10: Inicialização das posições e velocidades da População (Aleatório)
20: Inicialização do arquivo ou vetor de líderes (Quality leader)
30: For  $i=1$  to número total de partículas do
40:     Selecionar um líder do vetor (Quality)
50:     Avaliação do líder na função de atitude (fitness function)
60:         if o valor da função fitness e melhor que  $p_{best}$ .
70:             o valor de fitness se atualiza como novo  $p_{best}$ 
80:         End if
90:     Atualizar a velocidade e posição da partícula
100: End for
110: Atualizar o Quality
120: Until o cumprimento do critério de parada

```

Um ponto importante no método é a seleção e atualização do líder além da criação de novas soluções (KUMAR e MINZ, 2014).

Otimização multi-objetivo baseada em recozimento simulado (MOSA)

A estrutura básica do algoritmo MOSA desenvolvido por Suppapitnarm *et al.*(2000) é uma estratégia montada sobre o mesmo método de recozimento simulado (*Simulated Annealing*) de otimização no caso tradicional mono-objetivo expandido para multi-objetivo. Uma função objetivo "composta", G , é formada entre cada um dos objetivos únicos e é definida como:

$$G = \sum_{i=1}^M \ln|f_i| \quad (2.41)$$

em que, f_i são as funções objetivo, $G(\mathbf{x}_n)$ é a avaliação da função composta em cada ponto \mathbf{x}_n do espaço de decisão. Depois da aplicação de uma perturbação aleatória a cada variável, o valor de $G(\mathbf{x}_{n+1})$ é computado, até que um critério de convergência seja atingido, o qual é testado entre iterações sucessivas com base na probabilidade p da Equação 2.42.

$$p = \exp\left(-\frac{1}{T}[G(\mathbf{x}_{n+1}) - G(\mathbf{x}_n)]\right) \quad (2.42)$$

onde T , é a temperatura do sistema. Usando as Equações. 2.41 e 2.42, a probabilidade p pode ser escrita como:

$$p = \exp\left(-\frac{1}{T} \sum_{i=1}^K \ln \left[\frac{f_i(\mathbf{x}_{n+1})}{f_i(\mathbf{x}_n)} \right]\right) \quad (2.43)$$

Tabela 2.4: Pseudo Código MOSA. Fonte: Suppaitmarm et al. (1999).

MOSA: Pseudo Codigo

10: Inicializar população inicial (\mathbf{x}_0), vetor de Funções objetivo $G(\mathbf{x}_0)$.
 20: Inicializar da temperatura de controle T .
 30: **Until** o critério de parada seja atingido **do**
 40: Perturbar a população randomicamente ($\mathbf{x}_n \rightarrow \mathbf{x}_{n+1}$)
 50: Avaliação do vetor de funções objetivo $G(\mathbf{x}_{n+1})$
 60: **if** (teste de probabilidade p) **do**
 70: Atualizar a população ($\mathbf{x}_{n+1} \rightarrow \mathbf{x}_n$)
 80: Armazenagem da população
 90: **else**
 100: Reduze periodicamente a Temperatura T
 110: **end if**
 120: **end until** o cumprimento do critério de parada

2.5. Métricas em Otimização Multi-objetivo

Esta seção cobre certas métricas de avaliação do desempenho dos problemas multi-objetivo (em inglês: *Multi-Objective Performance Metrics* - MOPMs), empregadas atualmente na literatura. O principal objetivo das MOPMs é quantificar o desempenho dos algoritmos de otimização multi-objetivo, tendo em conta um aspecto específico das soluções. O foco principal é avaliar o desempenho dos MOOAs em sua busca por aproximações mais precisas, com um número significativo de soluções ótimas de Pareto, distribuídas uniformemente entre todos os objetivos (ZITZLER *et al.* 2013). Em geral, as métricas de desempenho podem ser classificadas em quatro categorias principais: de convergência, de cobertura, de capacidade e de diversidade.

2.5.1. Métricas de convergência

Essa classe de métricas de desempenho quantifica a proximidade das soluções obtidas entre os conjuntos não dominados de cada iteração com respeito da frente de Pareto final (quando ela é conhecida), ou entre os conjuntos não dominados de iterações consecutivas. Dentre estas métricas de convergência uma das mais usada é a distância geracional normalizada.

Distância geracional normalizada (GD): Esta métrica foi proposta por Van Veldhuizen e Lamont, (2000). Ela é baseada no cálculo da distância Euclidiana d_i das soluções i dos

conjuntos não dominados de cada iteração, com respeito à frente de Pareto j , ou entre os conjuntos não dominados de duas iterações diferentes através das Equações 2.44 e 2.45.

$$GD^* = \frac{\sum_{i=1}^{|P^*|} d_i}{|P^*|} \quad (2.44)$$

$$d_i = \min_{j=1}^{|P^*|} \left(\sum_{k=1}^M \left(\frac{f_k(i) - f_k(j)}{f_k^{max} - f_k^{min}} \right)^2 \right)^{\frac{1}{2}} \quad (2.45)$$

em que, f_k^{max} e f_k^{min} são os valores máximo e mínimo de cada função objetivo e $|P^*|$ o número de pontos no conjunto não dominado. Para manter a métrica de convergência dentro de $[0,1]$, uma vez que os valores de GD^* são calculados em todas as iterações, eles são normalizados através do quociente do valor máximo GD_{max}^* , (DEB e JAIN, 2002).

$$GD = \frac{GD^*}{GD_{max}^*} \quad (2.46)$$

2.5.2. Métricas de cobertura

Essa classe de métricas define quão bem as soluções obtidas “cobrem” a faixa de cada um dos objetivos. O espaçamento é uma das métricas de cobertura mais usadas na literatura para determinar o nível de cobertura atingido pelos algoritmos de otimização multi-objetivo.

Espaçamento (SM): Esta métrica indica como os pontos são distribuídos uniformemente no espaço objetivo como uma variação, de acordo com a Equação 2.47

$$SM = \sqrt{\frac{1}{|P^*|} \sum_{i=1}^{|P^*|} (d_i - \bar{d})^2} \quad (2.47)$$

$$d_i = \min \left\{ \sum_{k=1}^K |f_k^i - f_k^m| \right\}_{m \in |P^*| \wedge m \neq i} \quad (2.48)$$

em que, \bar{d} é a média das normas unitárias d_i , entre pontos solução do conjunto não dominado ou a frente de Pareto (espaço objetivo). O valor zero dessa métrica indica que todos os membros da frente de Pareto estão espaçados uniformemente entre eles.

2.5.3. Métricas de capacidade

Estes grupos de métricas calculam o número ou proporção de soluções não dominadas no espaço objetivo, e que satisfazem determinados requisitos predefinidos, com relação ao número total de avaliações da função de aptidão.

Métrica da Taxa de Sucesso Perceptual (HRM%): se $|\mathcal{F}^*|$, é o número de pontos resultantes sobre o fronte de Pareto, e o parâmetro FC indica o número total de avaliações da função de aptidão, então a métrica HRM% é calculada de acordo com a seguinte equação:

$$HRM\% = \frac{|\mathcal{F}^*|}{FC} 100[\%] \quad (2.49)$$

Um grande valor de HRM% indica que o método consome menos tempo computacional no cálculo da função de aptidão para obter a solução ótima de Pareto numa porcentagem de avaliações da função de aptidão.

2.5.4. Métricas de diversidade

Essas métricas refletem a extensão da dispersão encontrada entre as soluções dos conjuntos não dominados de cada iteração. Note-se que é desejável obter um conjunto de soluções que cubra, ou represente, toda a frente de Pareto, sem soluções muito isoladas. Entre as métricas de diversidade mais usadas estão: a métrica de espalhamento Δ .

Métrica de espalhamento (*Spread*): Avalia a dispersão das soluções no conjunto \mathcal{F}^* ao longo da fronteira de Pareto, assim como a distribuição entre soluções contíguas (no espaço dos objetivos) de \mathcal{F}^* :

$$\Delta = \frac{\sum_{i=1}^m d_i^e + \sum_{i=1}^{|\mathcal{F}^*|} |d_i - \bar{d}|}{\sum_{i=1}^m d_i^e + |\mathcal{F}^*| \bar{d}} \quad (2.50)$$

onde d_i^e denota a distância Euclidiana entre os pontos extremos de \mathcal{F}^* ao longo da i -ésima coordenada, d_i representa a distância Euclidiana de cada ponto em \mathcal{F}^* para os vizinhos mais próximo contidos neste mesmo conjunto, e \bar{d} é a média de d_i , $i \in (1, \dots, |\mathcal{F}^*|)$. Assim sendo, quanto menor o valor de Δ , melhor é a distribuição das soluções.

2.5.5. Métricas de Sucesso

Uma forma de medir o sucesso comparativo dos resultados atingidos por diversos algoritmos de otimização multi-objetivo, é através de uma nova métrica estabelecida neste trabalho, denominada: Métrica Percentual de contribuição de Soluções na Frente das Frentes de Pareto (*MPSFF*). Esta métrica quantifica a porcentagem de pontos não dominados que resultam da aplicação de um algoritmo em particular dentro de um gráfico comparativo das frentes de Pareto dos algoritmos comparados (Frente das Frentes de Pareto). O cálculo é feito depois de se aplicar os critérios de dominância no conjunto de pontos contendo as soluções de todos os algoritmos comparados. Por exemplo: Sejam os algoritmos de otimização multi-objetivo A e B , em que \mathcal{F}_A^* e \mathcal{F}_B^* , representam o conjunto de pontos das frentes de Pareto geradas pelos algoritmos. \mathcal{F}_{pp}^* é o conjunto de pontos resultantes depois de aplicar os critérios de dominância sobre a união dos conjuntos das frentes de A e B ($\mathcal{F}_A^* \cup \mathcal{F}_B^*$).

Sendo $\mathcal{F}_A^{**} = \mathcal{F}_{pp}^* \cap \mathcal{F}_A^*$ a métrica percentual de melhores valores na frente de Pareto para o algoritmo A , pode-se calcular através da Equação 2.51

$$(\text{MPSFF})_A = \frac{|\mathcal{F}_A^{**}|}{|\mathcal{F}_{pp}^*|} \times 100\% \quad (2.51)$$

É importante ressaltar que todas as métricas devem ser avaliadas em conjunto para poder valorar a qualidade da frente obtida pelos algoritmos de otimização multi-objetivo.

Capítulo 3

Otimização Assistida por Metamodelos

Como já mencionado neste texto, a realização de projetos de engenharia envolve frequentemente a aplicação de métodos matemáticos complexos para resolver ou simular o modelo físico do problema, exigindo-se para isso o uso intensivo de computadores. Dependendo do nível de fidelidade e complexidade do modelo físico requerido pelo projetista, a avaliação das variáveis de projeto no modelo podem requerer tempos computacionais proporcionalmente elevados. Por exemplo, nos problemas de aerodinâmica, os modelos potenciais não viscosos tratados com o método dos painéis 3D (KATZ e PLOTKIN, 2001), são computacionalmente menos custosos que os modelos viscosos baseados em CFD, principalmente quando envolvem também a aplicação de modelos de turbulência. Nas mesmas metodologias de projeto baseadas em CFD, pode-se considerar ainda que um modelo 3D é mais fiáveis que um 2D, pois é capaz de representar mais fielmente a complexidade geométrica e física do problema à custa porém do aumento da discretização requerida e do tempo computacional.

A questão do custo computacional dos modelos torna-se ainda mais crítica ao se tratar de problemas de otimização. Aqui, a avaliação das funções objetivo envolve chamadas recorrentes dos solucionadores (*solvers*) dos modelos para diferentes valores das variáveis de projeto. Nesse contexto, surge o interesse no emprego de funções de aproximação denominadas metamodelos, modelos substitutos ou superfícies de resposta cuja construção pode ser feita através de métodos clássicos de regressão polinomial ou métodos de interpolação/regressão por funções de base radial, dentre outros. O uso criterioso de metamodelos pode se constituir numa ferramenta poderosa para aceleração de processos de otimização com funções objetivo computacionalmente custosas. Para isso, é importante orientar (assistir) o processo no sentido de um menor número possível de avaliações das funções custosas, de preferência usando

iterativamente as avaliações correntes para aprimorar os metamodelos nas regiões mais promissoras do espaço de projeto (de decisão). Os metamodelos construídos em certa iteração são então usados para ajudar a prever futuros pontos de busca, de modo que as regiões de interesse possam ser identificadas mais rapidamente.

3.1. Tipos de Otimização Assistida por Metamodelos

O desenvolvimento dos Algoritmos de Otimização Assistida por Metamodelos – AOAM, tem sido de grande interesse nas últimas décadas. A revisão da literatura indica que as técnicas de modelagem têm sido usadas de duas formas na resolução de problemas de otimização com funções objetivo computacionalmente custosas. A primeira delas é através da construção de metamodelos *estancos* baseado em um banco de dados com um conjunto fixo de pontos de avaliação custosa escolhidos a priori possivelmente seguindo algum critério de “erro mínimo”. A segunda forma de AOAM é através de um aprimoramento iterativo da qualidade do metamodelo nas regiões de interesse durante o processo de otimização. Sendo assim, neste trabalho, os AOAM são classificados em dois grupos: (i) Algoritmos baseados na Construção de um Metamodelo Único (ACMU), antes da execução do processo de otimização e (ii) Algoritmos baseados na Construção Dinâmica de Metamodelos (ACDM), aprimorados durante o processo de otimização, através da atualização de um banco de dados em que se incluem novos pontos avaliados nas funções custosas em cada iteração. Esses pontos são selecionados dentre um conjunto de pontos candidatos por meio de diversos critérios e técnicas de seleção especiais. Os pontos candidatos, por sua vez, são estabelecidos ou gerados previamente através da solução de problemas auxiliares de otimização dos metamodelos, procurando focar regiões de interesse e/ou garantir diversidade de busca.

O algoritmo MO-CORS proposto neste trabalho é de tipo ACDM. Nesse sentido, a seguir são definidos os aspectos básicos importantes sobre a estrutura destes algoritmos, como: as técnicas de construção do banco de dados inicial (planos de experiências), os métodos de construção de metamodelos e os métodos de busca e seleção de pontos de avaliação.

3.2. Algoritmos de otimização assistida por metamodelos (AOAM)

Um esquema geral da estrutura dos AOAM para problemas mono-objetivo foi apresentado por Knowles e Nakayama (2008), Embora situado no contexto de problemas com

objetivo único, ele é usado como referência nesta pesquisa, com as modificações pertinentes para abranger os problemas multi-objetivo.

A estrutura dos AOAM apresenta seis partes importantes: (i) a criação do banco de dados inicial (ii) a avaliação do banco de dados inicial na função objetivo custosa, (iii) a construção do metamodelo, (iv) a aplicação de um algoritmo de otimização sobre o metamodelo, para a obtenção dos pontos candidatos à próxima avaliação na função custosa, (v) a seleção do ponto de treino para atualização do banco de dados e (vi) verificação do critério de parada para a obtenção do ótimo do problema. A Figura 3.1 mostra um fluxograma dos AOAM geral, onde se pode observar o processo de otimização e cada parte da estrutura.

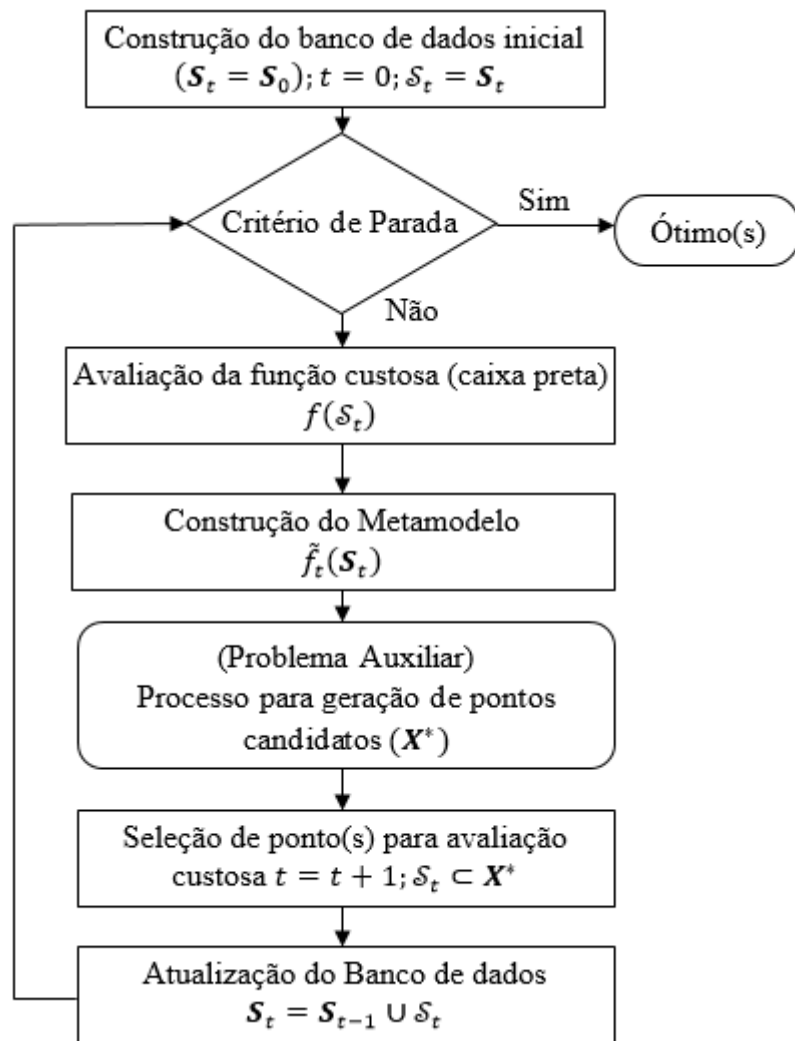


Figura 3.1: Esquema geral de processo de otimização baseado em metamodelos.
Adaptado de Knowles E Nakayama (2008)

O procedimento de otimização iterativa apresentado na Figura 3.1, alterna entre fazer avaliações no modelo de alta fidelidade (f) e no metamodelo de baixa fidelidade (\tilde{f}), usando o modelo apenas para atualizar ou retrainar o metamodelo nas regiões de interesse em cada

iteração (t). Desta forma, o número de avaliações do modelo computacionalmente custoso pode ser reduzido substancialmente.

Para problemas mono-objetivo, o procedimento geral dos AOAM da Figura 3.1 pode ser descrito no algoritmo a seguir:

Passo 1: *Construção do Banco de Dados inicial do espaço de decisão.* Consiste em construir uma amostragem de pontos inicial \mathbf{S}_0 usando técnicas de projeto de experimentos.

$\mathbf{S}_0 = \{\mathbf{X}_1, \dots, \mathbf{X}_i\}; i = 1, \dots, N_{exp}$, em que N_{exp} é o número de experiências e $\mathbf{X}_j = (x_1, x_2, \dots, x_{n_{var}}); j = 1, \dots, n_{var}$ é n_{var} é o número de variáveis.

Passo 2: *Inicialização.* Inicializa-se o contador de iterações ($t = 0$), o banco de dados inicial ($\mathcal{S}_t = \mathbf{S}_0$) e o conjunto de pontos selecionados \mathcal{S}_t . Para a iteração 0, \mathcal{S}_t representa o próprio conjunto de pontos igual ao conjunto de pontos do banco de dados inicial, ou seja, $\mathcal{S}_0 = \mathbf{S}_0$ e para as iterações seguintes, \mathcal{S}_t irá representar o único ponto ótimo resultante da resolução do problema de otimização auxiliar no **Passo 5**.

Início do ciclo iterativo (Critério de parada)

Passo 3: *Avaliação na função custosa.* Avaliação dos pontos selecionados usando a função objetivo custosa (caixa preta), $f(\mathcal{S}_t)$. Esse valor é adicionado ao banco de dados.

Passo 4: *Construção do metamodelo.* Consiste na construção do metamodelo baseado no banco de dados da iteração, usando funções de aproximação, $\tilde{f}_t(\mathcal{S}_t, f(\mathcal{S}_t))$.

Passo 5: *Problema auxiliar.* Consiste na resolução de um problema de otimização auxiliar (através de um algoritmo de otimização mono-objetivo) em que a função objetivo é o metamodelo \tilde{f}_t . A solução \mathbf{X}^* desse problema corresponde ao ponto de avaliação da função custosa na seguinte iteração.

Passo 6: *Atualização.* Atualiza-se o conjunto de avaliação da função custosa, $\mathcal{S}_{t+1} = \{\mathbf{X}^*\}$. Atualiza-se o banco de dados, ($\mathcal{S}_{t+1} = \mathcal{S}_t \cup \{\mathbf{X}^*\}$). Atualiza-se o contador de iterações, $t \leftarrow t + 1$.

Fim do ciclo iterativo.

Na extensão do algoritmo anterior para problemas multi-objetivo de tipo direto, deve-se ter em conta certos aspectos teóricos próprios destes problemas, como: a multiplicidade das atividades executadas, a introdução de técnicas de seleção de pontos de avaliação, a realização de uma busca ampla nos espaços de decisão e objetivo e o cumprimento de aspectos relacionados a qualidade da frente de Pareto, como: cobertura, convergência, capacidade e diversidade. A seguir, apresenta-se o algoritmo geral AOAM para problemas multi-objetivo.

Passo 1: *Construção do Banco de Dados inicial do espaço de decisão.* Consiste em construir uma amostragem de pontos inicial \mathbf{S}_0 usando técnicas de projeto de experimentos.

$$\mathbf{S}_0 = \{\mathbf{X}_1, \dots, \mathbf{X}_i\}; i = 1, \dots, N_{exp} \text{ e } \mathbf{X}_j = (x_1, x_2, \dots, x_{n_{var}}); j = 1, \dots, n_{var}.$$

Passo 2: *Inicialização.* Inicializa-se o contador de iterações ($t = 0$), do banco de dados inicial ($\mathbf{S}_t = \mathbf{S}_0$) e o conjunto de pontos selecionados \mathcal{S}_t . Para a iteração 0, \mathcal{S}_t representa o proprio conjunto de pontos igual ao conjunto de pontos do banco de dados inicial, ou seja, $\mathcal{S}_0 = \mathbf{S}_0$; para as iterações seguintes \mathcal{S}_t irá representar o conjunto dos N_{ps} pontos selecionados de acordo com o *Passo 7*.

Início do ciclo iterativo (Critério de parada)

Passo 3: *Avaliação nas funções custosas.* Avaliação do(s) N_{ps} ponto(s) selecionado(s) no espaço de decisão usando as funções objetivo (caixa preta), para obter o conjunto de imagens no espaço objetivo: $\mathbf{\Omega}_t(\mathcal{S}_t) = \{\mathbf{f}_1, \dots, \mathbf{f}_{N_{ps}}\}$, em que $\mathbf{f}_i = (f_1 \dots f_{N_{obj}})$, e o vetor de objetivos resultante da avaliação de cada ponto selecionado \mathbf{X}_i em cada função objetivo $f_k, k = 1, \dots, N_{obj}$, sendo N_{obj} o número de objetivos do problema. A matriz de ordem $N_{ps} \times N_{obj}$ dos valores correspondentes a $\mathbf{\Omega}_t(\mathcal{S}_t)$ é adicionada ao banco de dados.

Passo 4: *Determinação do conjunto não dominado.* Determina-se o conjunto não dominado de pontos P_t^* entre os pontos do banco de dados na presente iteração, \mathbf{S}_t .

Passo 5: *Construção dos metamodelos.* Consiste na construção de um metamodelo $\tilde{f}_k(\mathbf{S}_t, \mathbf{\Omega}_t(\mathbf{S}_t))$ para cada função objetivo custosa $f_k, k = 1, \dots, N_{obj}$ baseada no banco de dados da presente iteração.

Passo 6: *Problemas auxiliares.* Otimização do conjunto de metamodelos através da aplicação de um algoritmo de otimização multi-objetivo, para obtenção dos pontos candidatos $\tilde{\mathcal{F}}^*$, (frentes de Pareto ou conjunto de pontos obtidos dos problemas de otimização auxiliar) para avaliação custosa na próxima iteração.

Passo 7: *Seleção dos pontos para avaliação custosa.* Consiste na aplicação de critérios e técnicas que permitam selecionar, entre os candidatos, os N_{ps} pontos que serão avaliados na função custosa na próxima iteração $\mathcal{S}_{t+1} \subseteq \tilde{\mathcal{F}}^*$. A seleção deve ser feita buscando atingir uma boa cobertura, capacidade e diversidade da frente de Pareto final assim como a convergencia mais rápida possível.

Passo 8: *Atualização.* Atualiza-se o conjunto de avaliação da função custosa, $\mathcal{S}_{t+1} = \{\mathbf{X}^*\}$. Atualiza-se o banco de dados, ($\mathbf{S}_{t+1} = \mathbf{S}_t \cup \mathcal{S}_{t+1}$). Atualiza-se o contador de iterações, $t \leftarrow t + 1$.

Fim do ciclo iterativo.

É muito importante abordar cada parte do procedimento descrito para detectar os pontos-chaves para a construção do algoritmo proposto nesta pesquisa, bem como para seu aprimoramento em futuros trabalhos. Neste sentido, apresenta-se a seguir uma descrição mais detalhada dos aspectos mais importantes do algoritmo geral baseado em metamodelos.

3.3. Plano de Experiências

Segundo Silva (2011), “A qualidade e a capacidade de previsão de um metamodelo podem ser fortemente dependentes da distribuição dos pontos de dados do modelo dos quais uma aproximação será construída”. Nesse sentido, distribuições adequadamente projetadas são essenciais para uma análise de metamodelos de forma efetiva e permite adquirir uma boa avaliação do espaço de solução. Frequentemente as técnicas de construção de Plano de Experiências (DOE – *Design Of Experiment*) são empregadas para a construção da amostragem do banco de dados inicial do espaço de decisão nos AOAM. As técnicas DOE mais importantes empregadas pelos desenvolvedores de algoritmos de otimização na construção de planos de experiências são: (i) Amostragem Aleatória, (ii) Plano Fatorial Completo (PFC), (iii) Plano Fatorial Incompleto (PFI) e (iv) Hipercubo Latino (HCL). A seguir, apresenta-se uma descrição das técnicas PFC, PFI e HLC a serem aplicadas nesta pesquisa.

PFC: Esta técnica consiste em dividir uniformemente cada dimensão do espaço de decisão com o mesmo número de níveis (N_{niv}) iguais. O número de experiências (N_{exp}) ou pontos no espaço de decisão resultantes da aplicação deste DOE é determinado pela Equação 3.1.

$$N_{exp} = (N_{niv})^{nvar} \quad (3.1)$$

No contexto dos AOAM, a aplicação desta técnica na inicialização dos dados referentes ao espaço de decisão apresenta vantagens para a construção do metamodelo devido à uniformidade na distribuição de pontos na região viável. No entanto, ela torna-se computacionalmente inviável quando o número de variáveis do problema é relativamente alto.

PFI: A base de construção empregada nesta técnica é similar à da técnica PFC, com a diferença que, neste caso, os número de níveis de cada variável em geral pode ser diferente. Sendo, N_{niv}^i , o número de níveis da variável i , $i = 1, 2, \dots, n_{var}$, então o número de experiências necessárias pode ser calculado através da Equação 3.2.

$$N_{exp} = \prod_{i=1}^{n_{var}} N_{niv}^i \quad (3.2)$$

Esta técnica é usada distribuindo os níveis de maior a menor em cada variável dependendo do nível de importância de cada uma delas dentro do problema analisado. De forma geral, o algoritmo utilizado para a construção de bancos de dados iniciais baseadas em PFC prévias à resolução dos problemas de otimização, resolve a construção de um plano de experiências para o caso PFI, transformando-se no caso particular PFC com níveis distintos para cada variável de projeto.

HCL: A técnica de plano de experiências Hipercubo Latino (HCL) opera dividindo o subespaço de cada componente vetorial $S_i, i = 1, 2, \dots, N$ em M subconjuntos disjuntos de igual probabilidade $p_{ik}, i = 1, 2, \dots, N; k = 1, 2, \dots, M$. Amostras de cada componente vetorial são extraídas de cada subconjunto específico de acordo com a Equação 3.3, (SHIELDS e ZHANG, 2016).

$$\mathbf{x}_{ik} = D_{x_i}^{-1}(U_{ik}); \quad i = 1, 2, \dots, N; \quad k = 1, 2, \dots, M \quad (3.3)$$

em que, U_{ik} são amostras uniformemente distribuídas no intervalo de limites laterais $[\mathbf{x}_k^{LI}, \mathbf{x}_k^{LS}]$ com $\mathbf{x}_k^{LI} = (k - 1)/M$ e $\mathbf{x}_k^{LS} = k/M$, e D_{x_i} é uma distribuição aleatória acumulativa de probabilidade de extração.

A principal vantagem do uso deste método dentro dos AOAM para a geração de conjuntos de amostragem iniciais dos espaços de decisão, é que ele é muito útil para trabalhar com um grande número de variáveis, além de prover uma uniformidade nas distribuições de pontos aceitável, Neste sentido esta técnica é utilizada na literatura em algoritmos similares ao proposto, neste trabalho, tais como os de Akhtar e Shoemaker (2016) e de Müller (2017).

3.4. Construção do Metamodelo

A base fundamental do algoritmo proposto neste trabalho, é precisamente a construção de *metamodelos*. Segundo Knowles e Nacayama (2008) os métodos principais para a construção dos metamodelos podem ser classificados segundo a técnica usada, tais como: (i) regressão Gaussiana (*Kriging*), (ii) inteligência artificial, (iii) ferramenta de suporte vetorial, (iv) regressão de suporte vetorial e (v) representação através de Funções de Base Radial (RBF). No algoritmo desenvolvido neste trabalho, a construção dos metamodelos é feita através de um

esquema numérico baseado na utilização de funções de base radial. Os algoritmos mono-objetivo (CORS), desenvolvido por Regis e Shoemaker (2005) e o algoritmo OGM desenvolvido por Silva (2011), foram bem sucedidos no uso de RBF. Como o algoritmo MO-CORS proposto nesse trabalho é uma extensão deles, optou-se por manter a mesma técnica. A seguir, são apresentadas generalidades para a construção de metamodelos dentro do contexto dos AOAMs, usando funções de base radial.

3.4.1. Funções de Base Radial.

As Funções de Base Radial (RBF) são funções de aproximação que dependem unicamente da distância da origem $\phi(\mathbf{X}) = \phi(\|\mathbf{X}\|)$, ou alternativamente da distância de outro ponto qualquer \mathbf{X}_i chamado centro, $\phi(\mathbf{X}) = \phi(\|\mathbf{X} - \mathbf{X}_i\|)$, onde \mathbf{X} e $\mathbf{X}_i \in \mathbb{R}^{n_{var}}$, (SANTOS *et al.*, (2018).

As RBF evocadas aqui são usadas para construir metamodelos para uma função $f(\mathbf{X})$ via interpolações a partir de um conjunto de pontos distintos $\mathbf{X}_i, i = 1, \dots, N_{exp}$, com valores funcionais conhecidos, $f(\mathbf{X}_i) = f_i$. Uma função de aproximação interpolante $s(\mathbf{X})$ é dada pela soma ponderada das RBF associadas aos diferentes centros \mathbf{X}_i , segundo a Equação 3.4.

$$s(\mathbf{X}) = \sum_{i=1}^{N_{exp}} \alpha_i \phi(\|\mathbf{X} - \mathbf{X}_i\|) \quad (3.4)$$

A aplicação das condições de interpolação $s(\mathbf{X}_i) = f_i, i = 1, \dots, N_{exp}$, mediante a Equação 3.4 conduz a um sistema de equações lineares ($[\varphi]\{\alpha\} = \{F\}$) donde são calculados os coeficientes α_i . Algumas RBF dependem de um parâmetro de forma c em sua definição, outras não. Exemplos dos dois tipos estão mostrados nas Tabelas 3.1 e 3.2, sendo r a distância do centro. O valor do parâmetro de forma define a suavidade na forma da aproximação feita. O aumento de c torna a RBF “mais plana” e aumenta a precisão da aproximação, porém à custa de uma piora no seu condicionamento.

Tabela 3.1: RBF dependentes do parâmetro de forma c . Fonte: Silva (2011).

Nome da RBF	$\phi(r)$
Gaussiana	$\exp(-r^2/c^2)$
Multiquádrica	$\sqrt{r^2 + c^2}$
Multiquádrica inversa	$1/\sqrt{r^2 + c^2}$

Tabela 3.2: RBF independentes do parâmetro de forma c . Fonte: Santos *et al.* (2018)

Nome da RBF	$\phi(r)$
Poliharmônicas	$r^{2\beta-1} \quad \beta \in \mathbb{N}$
Splines Poliharmônicas	$r^{2\beta} \log r \quad \beta \in \mathbb{N}$
<i>Thin Plates Spline</i>	$r^2 \log r$

Com a finalidade de melhorar a precisão da aproximação e a suavidade das RBF independentes do parâmetro de forma, é necessário em alguns problemas adicionar um polinômio de extensão à aproximação da Equação. 3.4. A Equação 3.5 ilustra a situação no caso particular de um polinômio de extensão linear.

$$s(\mathbf{X}) = \sum_{i=1}^{N_{exp}} \alpha_i \phi(\|\mathbf{X} - \mathbf{X}_i\|) + \mu_0 + \sum_{i=1}^{n_{var}} \mu_i x_i \quad (3.5)$$

em que x_i são as variáveis de decisão no presente contexto ou as componentes do vetor de decisão $\mathbf{X} = (x_1 \ x_2 \ \dots \ x_{n_{var}})$ e μ_i são os coeficientes do polinômio de extensão linear. A determinação dos coeficientes α_i e μ_i é feita impondo as N_{exp} condições de interpolação e $n_{var} + 1$ condições de compatibilidade para o polinômio de extensão. Resulta daí o sistema de ordem $N_{exp} + n_{var} + 1$ da Equação 3.6 escrito em forma matricial, com matrizes componentes apresentadas na Tabela 3.3. O vetor nulo $\hat{\mathbf{0}}$ e a matriz \mathbf{P} refletem a inserção das condições de compatibilidade: o vetor dos valores de cada base polinomial $1, x_1, x_2, \dots, x_{n_{var}}$, calculados nos pontos de interpolação, deve ser ortogonal ao vetor $\boldsymbol{\alpha}$ dos coeficientes das RBF's (SCHABACK, 1995).

$$\begin{bmatrix} \boldsymbol{\varphi} & \mathbf{P} \\ \mathbf{P}^T & \hat{\mathbf{0}} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\mu} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ \hat{\mathbf{0}} \end{Bmatrix} \quad (3.6)$$

Tabela 3.3: Especificação das matrizes componentes da Equação 3.6.

$\boldsymbol{\varphi} \in \mathbb{R}^{N_{exp} \times N_{exp}}$	$\boldsymbol{\varphi} = [\varphi_{ij}] = \phi(\ \mathbf{X}_i - \mathbf{X}_j\); \quad i, j = 1, 2, \dots, N_{exp}$
$\mathbf{P} \in \mathbb{R}^{N_{exp} \times (n_{var}+1)}$	$\mathbf{P} = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{n_{var},1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{1,N_{exp}} & \dots & x_{n_{var},N_{exp}} \end{bmatrix}$
$\mathbf{F} \in \mathbb{R}^{N_{exp} \times 1}$	$\mathbf{F} = \left(f(\mathbf{X}_1) \ f(\mathbf{X}_2) \ \dots \ f(\mathbf{X}_{N_{exp}}) \right)^T$
$\boldsymbol{\alpha} \in \mathbb{R}^{N_{exp} \times 1}$	$\boldsymbol{\alpha} = \left(\alpha_1 \ \alpha_2 \ \dots \ \alpha_{N_{exp}} \right)^T$
$\boldsymbol{\mu} \in \mathbb{R}^{(n_{var}+1) \times 1}$	$\boldsymbol{\mu} = \left(\mu_0 \ \mu_1 \ \mu_2 \ \dots \ \mu_{n_{var}} \right)^T$

No apêndice A, descreve-se o algoritmo para a construção de metamodelos baseada em RBF, no contexto da presente pesquisa utilizado, seguido de um exemplo literal baseado num banco de dados com 4 pontos ($N_{exp} = 4$), num espaço de decisão de duas variáveis ($n_{var} = 2$), e uma só função objetivo ($N_{obj} = 1$). O intuito é ilustrar de forma sistemática e didática a aplicação do algoritmo desenvolvido.

Segundo as características da interpolação das RBF, estas em geral podem envolver uma estimativa do erro entre a aproximação e a função real. Este erro, conhecido como Erro Médio Quadrático (em inglês: *Root Mean Square Error* – RMSE), é calculado através da raiz quadrada da média do quadrado das diferenças, entre os valores da função real f e a função aproximada \tilde{f} , avaliadas nos pontos de uma amostragem (banco de dados), \mathbf{S} . Segundo a Equação 3.7.

$$\text{RMSE} = \left[\frac{1}{N_p} \sum_{i=1}^{N_p} (f(\mathbf{S}) - \tilde{f}(\mathbf{S}))^2 \right]^{\frac{1}{2}} \quad (3.7)$$

em que, N_p é o número de pontos no banco de dados. Este erro é frequentemente usado para determinar a qualidade na construção de metamodelos, mas ele requer de um banco de dados com uma grande quantidade de pontos, o qual resulta inviável quando a função a aproximar é computacionalmente custosa. Nesse sentido, Rippla (1999) desenvolveu um algoritmo eficiente para simular o cálculo de RMSE, o qual foi utilizado por Silva (2011) e incorporado também nesta pesquisa, para a determinação automática do parâmetro de forma c das RBF.

3.4.2. Determinação automática do parâmetro de forma c .

No contexto dos AOAM de tipo ACMU, uma das técnicas utilizadas para a construção de metamodelos neste contexto, consiste na minimização de uma função que imita o erro médio quadrático através da técnica LOOCV – *Leave One Out Cross Validation*. Rippla (1999) desenvolveu um algoritmo eficiente em termos de tempo computacional baseado em LOOCV para estimar o erro E_i associado à retirada de cada ponto i do banco de dados usando as Equações 3.8 e 3.9.

$$[\boldsymbol{\varphi}]\{\boldsymbol{\alpha}\} = \{\mathbf{F}\} \Rightarrow [\varphi_{ij}]\{\alpha_i\} = \{F_i\} \quad (3.8)$$

$$E_i = \frac{\alpha_i}{[\varphi_{ii}]^{-1}} \quad (3.9)$$

O erro total E é então calculado através da norma euclidiana ou da norma unitária do vetor de componentes E_i . A Equação 3.8 deriva da Equação 3.6 sem polinômio de extensão linear. Na Equação 3.9, $[\varphi_{ii}]^{-1}$, representa o componente i da diagonal da inversa da matriz $[\boldsymbol{\varphi}]$. Para determinar o valor ótimo de c , o algoritmo de Rippa resolve um problema de otimização mono-objetivo, em que o erro $E(c)$, é minimizado com uma amostragem significativa de valores. Para cada valor de c , Rippa não calcula de fato a inversa de $\boldsymbol{\varphi}(c)$, mas diretamente os valores $[\varphi_{ii}]^{-1}$, com um esquema de complexidade computacional também de $O(N^3)$. Note-se que o esquema LOOCV padrão exigiria a solução de N sistemas do tipo (Equação 3.8), um para cada centro retirado, com $N - 1$ equações ($O(N^3)$), implicando numa complexidade computacional de $O(N^4)$.

A aplicação de estratégias para a automatização do parâmetro de forma dentro do contexto dos AOAM tipo ACDM, têm sido pouco testada. Silva (2011) testou a implementação de um algoritmo para a automatização da escolha do parâmetro de forma dentro do contexto do algoritmo OGM (OGM-LOOCV), baseado em duas formas de aplicação chamadas de modos: (Modo 1) otimizando o valor de c apenas para o banco de dados inicial e (Modo 2) obtendo um valor de c para cada metamodelo dentro do problema auxiliar. Depois dos testes nas funções de Dixon Szêgo, com a RBF inversa multiquádrica, Silva (2011) recomenda o uso do Modo 2.

O algoritmo MO-CORS, inclui um algoritmo auxiliar baseado no Modo 2 de Silva (2011) para automatização da escolha do parâmetro de forma c dentro de uma região de confiança limitada pelo número de condição da matriz $[\varphi_{ij}]$, em cada metamodelo de cada iteração. Por enquanto, ele está implementado apenas para problemas mono-objetivo. Os resultados dos estudos prévios para a implementação e teste deste algoritmo dentro do contexto dos algoritmos de construção dinâmica de metamodelos como MO-CORS, são apresentados no apêndice B. No capítulo 4 são apresentados os resultados comparativos do uso de este algoritmo em funções de teste mono-objetivo.

3.5. Problema de Otimização Auxiliar.

O problema de otimização auxiliar é aquele que é usado para a obtenção de conjuntos de pontos candidatos para avaliação da função custosa em cada iteração. Como foi expressado nos algoritmos da seção 3.2, para o caso dos problemas mono-objetivo os AOAM-ACDM em geral têm um único problema de otimização auxiliar, devido a que se procura um único ponto solução. No caso dos AOAM-ACDM multi-objetivo, além de se procurar múltiplas soluções (frente de Pareto), procura-se também cobertura, diversidade e exploração dos extremos da frente, o que é de grande interesse na otimização de projetos de engenharia. Nesse sentido, os

desenvolvedores de AOAM-ACDM multi-objetivo costumam usar mais um problema de otimização auxiliar inclusive com metamodelos diferentes. O algoritmo MO-CORS emprega três problemas de otimização auxiliares que serão explicados no capítulo 4.

Nos AOAM em geral, a escolha do algoritmo de otimização para a resolução dos problemas auxiliares, não depende fortemente dos mesmos, é dizer, o usuário ou pesquisador pode escolher o algoritmo de otimização que melhor se adequa as aplicações pretendidas. No caso do MO-CORS foi preciso escolher um algoritmo de otimização para cada problema de otimização auxiliar, tendo em conta o foco dos pontos desejados de cada um deles.

3.6. Seleção dos Novos Pontos de Avaliação.

No início deste capítulo foi dito que um dos critérios básicos para avaliar o desempenho de algoritmos de otimização baseados em metamodelos é a redução do número de avaliações na função objetivo custosa. Tendo em consideração que a solução dos problemas multi-objetivo do tipo direto é a frente de Pareto, é muito importante o uso ou desenvolvimento de técnicas de seleção que orientem a escolha da menor quantidade possível de pontos de avaliação custosa. Concomitantemente, é importante que a escolha atenda a certos requisitos próprios dos problemas multi-objetivo: a globalidade ou localidade da busca no espaço de decisão, a diversificação das soluções na frente de Pareto, a exploração na vizinhança dos pontos do conjunto não dominado e, em particular, dos extremos da frente em cada iteração. Neste sentido, os trabalhos de Müller (2017) e Akhtar e Shoemaker, (2016), desenvolvem um conjunto de técnicas de seleção baseadas em buscas locais e globais nos espaços de decisão e objetivo. Algumas delas são discutidas a seguir.

3.6.1. Amostragem aleatória dos pontos não dominados.

Este é um procedimento de busca local que consiste em selecionar ao menos um ponto de forma aleatória dentro do espaço de decisão, de modo a garantir a exploração nas proximidades dos pontos não dominados durante uma iteração. A Figura 3.2 ilustra um exemplo de espaço de decisão com 8 pontos dos quais 6 são não dominados (pretos\vermelho). Ao redor de cada ponto são gerados aleatoriamente outros 6 pontos (azuis) dentro de um raio de seleção pequeno. Segundo Müller (2017) a quantidade de pontos gerados ao redor de cada ponto não dominado pode ser qualquer, entanto maior melhor, para ter mais opções candidatas dentre dos quais unicamente um deles é selecionado para avaliação da função custosa. Nesta pesquisa esta metodologia foi utilizada para gerar um conjunto de pontos candidatos auxiliares que são selecionados segundo as necessidades do algoritmo, como será explicado no capítulo 4.

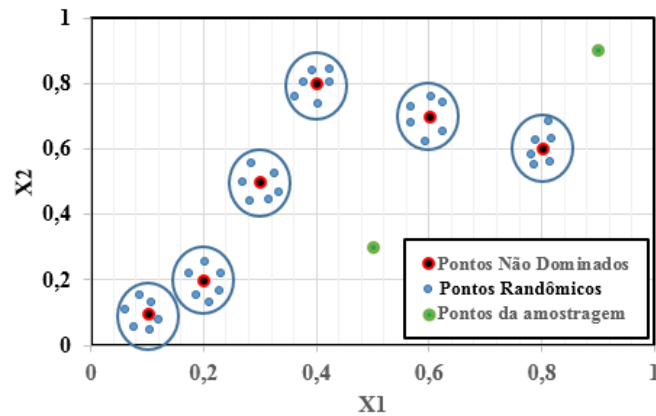


Figura 3.2: Amostragem aleatória de pontos no espaço de decisão.

O algoritmo para a seleção de pontos através da amostragem aleatória do conjunto não dominado descreve-se a seguir:

Passo 1: Selecionam-se os N pontos do conjunto não dominado P^* , no espaço de decisão.

Passo 2: Ao redor de cada um deles, selecionam-se aleatoriamente uma quantidade M de pontos, $\chi_1, \chi_2, \dots, \chi_N$, que estejam próximos de cada ponto não dominado ($\chi_l^{C_q}$), (em que $l = 1, \dots, N$), formando N nuvens (círculos azuis) $C_q = C_1, C_2, \dots, C_N$. Têm-se então $N \times M$ pontos aleatórios gerados.

Passo 3: Em cada ponto aleatório é calculado o valor aproximado das funções objetivo usando o metamodelo correspondente a cada uma delas $\tilde{f}_k(\chi_l^{C_q})$. As funções \tilde{f}_k , são escaladas de modo a mapear seus valores no intervalo $[0,1]$, com os piores valores mapeados em 0 e os melhores em 1, obtendo-se assim um vetor f_s , para cada nuvem. Este vetor f_s , é chamado vetor escalado do metamodelo.

Passo 4: (*Função peso dos valores do metamodelo*) Em cada nuvem é calculada a média de f_s , (\bar{f}_s), obtendo-se uma pontuação para cada nuvem, $\bar{f}_s \in [0,1]$.

Passo 5: (*Função peso das distâncias*) Em cada ponto aleatório é calculada a distância (f_d) de cada um deles até o ponto não dominado mais próximo $\|X_1 - \chi_1^{C_1}\|, \|X_1 - \chi_2^{C_1}\|, \dots, \|X_N - \chi_{N \times N}^{C_N}\|$. Esta função é escalada (mapeada) no intervalo $[0,1]$, tal que ao menor valor de distância corresponda 1 e ao maior 0, $f_d \in [0,1]$.

Passo 6: Soma-se para cada nuvem, cada uma das funções peso F_{AR} (Equação 3.10)

$$F_{AR} = \bar{f}_s + f_d \quad (3.10)$$

e escolhe-se o menor valor de F_{AR} para cada nuvem, obtendo-se N pontos candidatos para avaliação custosa.

Passo 7: Para cada ponto candidato é calculada a distância de aglomeração (Equação 2.33) e é aplicado o critério de dominância sobre seus valores \tilde{f} calculados no passo 3, descartando aqueles com uma distância de aglomeração baixa e que sejam dominados por outros.

3.6.2. Amostragem aleatória no espaço de decisão.

Esta técnica de amostragem constitui um procedimento de buscas locais, semelhante ao descrito na seção 3.6.1, mas neste caso é gerada uma amostragem aleatória de pontos ao redor de todos os pontos no espaço de decisão numa iteração. As funções de peso e distância escaladas dos metamodelos são calculadas da mesma forma que nos passos 4 e 5 do algoritmo anterior. Calculam-se os valores da soma dos pesos usando a Equação 3.10, e é escolhido o menor valor, para posteriormente selecionar um conjunto de pontos segundo o passo 7.

3.6.3. Busca dos extremos da Frente de Pareto.

Müller (2017) define este como um procedimento de seleção que orienta uma busca global, cuja finalidade é procurar as extremidades da frente aproximada de Pareto pelos metamodelos de cada função objetivo. Neste sentido, para cada metamodelo \hat{f} , é resolvido um problema de otimização de busca global, pois o metamodelo (e as funções custosas) não são necessariamente unimodais. A ideia é que, minimizando cada objetivo separadamente, sem levar em conta os outros objetivos, podem ser explorados os pontos cujas imagens correspondem aos extremos da frente de Pareto. Após encontrá-los, são descartados aqueles que estejam muito próximos uns dos outros ou muito próximos dos pontos já avaliados na população. Caso as funções objetivo tenham atingido a convergência e os pontos sejam todos descartados, quer dizer, quando os mínimos dos metamodelos pouco se alteram entre as iterações, Müller (2017) sugere selecionar o ponto que maximiza a distância mínima (no espaço objetivo) entre ele e os pontos do banco de dados S_t avaliado até então (Equação 3.11). A imagem desse ponto estaria em uma região inexplorada do espaço objetivo e poderia contribuir para ajustar globalmente o metamodelo. Logo, o ponto \mathbf{X} selecionado no espaço de decisão S seria aquele que resolve o seguinte problema:

$$\max_{\mathbf{X} \in S} \left[\min_{\mathbf{X}_j \in S_t} \|\tilde{f}(\mathbf{X}_i) - f(\mathbf{X}_j)\| \right] \quad (3.11)$$

A técnica proposta por Müller (2017) resulta numa complexidade computacional alta devido à resolução de um problema de otimização global para cada função objetivo. O

algoritmo MO-CORS aproveita a garantia de convergência de CORS nos problemas mono-objetivo e gera pontos candidatos orientados a preencher os espaços nas proximidades dos extremos de cada conjunto não dominado até a frente de Pareto final.

3.6.4. Máxima das mínimas distâncias no espaço de decisão.

Este é um procedimento de escolha global baseado no cálculo das distâncias Euclidianas no espaço de decisão. Consiste em maximizar as distâncias mínimas entre os pontos candidatos para avaliação das funções custosas (conjunto S_c) e os pontos do banco de dados S_t . O ponto X selecionado será então o ponto candidato mais distante dos pontos já avaliados, pois resolve o seguinte problema:

$$\max_{X \in S_c} \left[\min_{X_j \in S_t} \|X - X_j\| \right] \quad (3.12)$$

3.6.5. Máxima das mínimas distâncias no espaço objetivo.

Este é um procedimento é análogo ao anterior, com a diferença que as distâncias agora são calculadas no espaço objetivo. Sejam $f(X)$ e $\tilde{f}(X)$, respectivamente, as imagens de X mediante as funções custosas e do metamodelo. O ponto X selecionado será então o ponto candidato que tende a induzir a menor aglomeração no espaço objetivo, pois resolve o seguinte problema:

$$\max_{X \in S_c} \left[\min_{X_j \in S_t} \|\tilde{f}(X) - f(X_j)\| \right] \quad (3.13)$$

3.6.6. Máxima distância de aglomeração.

Este método consiste numa busca orientada ao preenchimento dos espaços vazios na frente de Pareto em cada iteração. Este se baseia no cálculo da distância de aglomeração do conjunto não dominado P^* segundo o procedimento descrito no capítulo 2, e identifica-se aquele com menor valor, do qual é subtraída sua contra imagem no espaço de decisão, X^{crow} . É importante ressaltar também que, centrado neste ponto pode ser resolvido o problema de otimização auxiliar que explora localmente as vizinhanças do mesmo, (AKHTAR e SHOEMAKER, 2016).

3.7. Otimização Restrita Assistida Por Metamodelos (CORS).

O algoritmo de otimização global restrita baseada em superfícies de resposta (em inglês: *Constrained Optimization using Response Surface - CORS*), foi desenvolvido por Regis e Shoemaker (2005) para acelerar a resolução de problemas de otimização mono-objetivo que envolvem funções custosas tipo caixa preta, baseado em metamodelos construídos com RBF, especificamente a *Thin Plate Spline*. CORS, é essencialmente um algoritmo assistido por metamodelos para problemas mono-objetivo, que segue a estrutura base dos AOAM-ACDM mostrado na Figura 3.1 – com algumas particularidades na forma de exploração do espaço de decisão, através da construção iterativa de restrições baseadas nas componentes de um vetor de busca padrão. Além disso, Regis e Shoemaker (2005) utilizaram o algoritmo determinístico *DiRect (Dividing Rectangles)*, para a resolução do problema de otimização auxiliar. Silva (2010) e Silva *et al.* (2011) apresentam o algoritmo de Otimização Global baseada em Metamodelos (OGM) adaptado do CORS e introduzindo algumas variantes como: a utilização das RBF multiquádricas; incorporando um algoritmo para a determinação automática do parâmetro de forma c através da técnica da validação cruzada LOOCV; utilização de uma versão do algoritmo de otimização populacional de busca aleatória controlada (em inglês: *Controlled Random Search Algorithm – CRSA*), no problema de otimização auxiliar.

O algoritmo MO-CORS é uma extensão do CORS que envolve também variantes introduzidas no OGM. Espera-se que a contribuição principal de MO-CORS seja aproveitar a garantia de convergência do CORS para pontos ótimos globais demonstrada por Regis e Shoemaker (2005) em problemas mono-objetivo, em conjunto com técnicas de exploração e de seleção próprias de problemas multi-objetivo. A seguir, apresenta-se uma descrição do algoritmo base (CORS) para a resolução de problemas mono-objetivo.

Algoritmo CORS (REGIS e SHOMAKER, 2005):

Início do Algoritmo

Passo 1: *Seleção dos pontos iniciais.* Configurar $t = 1$ e selecionar um conjunto inicial de pontos $S_t = \{X_1 X_2 \dots X_{N_{exp}}\} \in \mathbb{R}^n; n = N_{exp}$ para avaliação da função computacionalmente custosa.

Passo 2: *Avaliação da função custosa.* Avaliar a função f nos pontos do conjunto S_i e preencher o banco de dados $D_t = \{(X, f(X)): X \in S_t\}$.

Início do processo iterativo: Enquanto o critério de parada não for satisfeito

Passo 3: *Construção do metamodelo.* Construir o metamodelo \tilde{f}_t com a informação do banco de dados D_t .

Passo 4: *Problema de otimização auxiliar.* Selecionar o novo ponto \mathbf{X}^* para próxima avaliação da função custosa f , que resulta da resolução de um problema de otimização auxiliar, tomando o metamodelo como função objetivo e restrições baseadas nos valores das componentes do vetor de busca padrão β e a máxima das mínimas distâncias Δ_t entre os pontos que discretizam o espaço de decisão (Matriz de cobertura; $\mathbf{X}_c \in \mathbf{M}_c$) e os pontos já avaliados X_j :

$$\begin{aligned} & \text{Minimizar } \tilde{f}(\mathbf{X}) \\ \text{sujeito a:} & \quad \|\mathbf{X}^* - \mathbf{X}_j\| \geq \beta_i \Delta_t, \quad j = 1, \dots, N_{exp} + i - 1 \\ \text{onde} & \quad \Delta_t = \max_{\mathbf{X}_c \in \mathbf{M}_c} \min_{1 \leq j \leq N_{exp} + t - 1} \|\mathbf{X}_c - \mathbf{X}_j\| \end{aligned}$$

Passo 5: *Avaliação da função custosa.* Avaliação da função no ponto $X^* = X_{N_{exp}+t}$.

Passo 6: *Atualização do banco de dados.*

$$S_{t+1} = S_i \cup \{X_{N_{exp}+t}\}; \quad D_{t+1} = D_t \cup \{(X_{N_{exp}+t}, f(X_{N_{exp}+t}))\}$$

incrementar o contador de iterações $t \leftarrow t + 1$

Fim do processo iterativo. Fim do algoritmo.

Note-se que as restrições fazem com que as buscas sejam moduladas ciclicamente pelas componentes β_i de um dado vetor de busca padrão $\beta = [\beta_1 \beta_2 \dots \beta_{N_{pb}}]$, cujos valores são todos escolhidos no intervalo $[0 \ 1]$. As componentes β_i são aplicados em ciclos de tamanho N_{pb} ao longo das iterações. Antes de se iniciar o processo iterativo, configurara-se $i = 0$. Antes do *Passo 3.2*, o valor de i é incrementado de 1. Após o *Passo 3.2*, verifica-se se $i = N_{pb}$ e em caso positivo, faz-se $i = 0$ novamente. Desse modo, o vetor de busca padrão é percorrido ciclicamente e, em cada ciclo de N_{pb} iterações, o valor de β_i se repete. Os valores maiores de β_i promovem buscas mais globais (a busca se afasta dos pontos já avaliados) e os menores valores intensificam buscas mais locais (a busca se aproxima dos pontos já avaliados). Regis e Shoemaker (2005) demonstraram que, para funções contínuas, basta um único valor β_i , não nulo para garantir a convergência do algoritmo a um ótimo global. Esse resultado matemático nem sequer depende do tipo de metamodelo ou otimizador empregado. Na prática, porém, a escolha desses elementos, além do próprio vetor de busca padrão, constituem aspectos que influem sobremaneira no desempenho computacional do algoritmo CORS.

Capítulo 4

Otimização Multi-Objetivo Restrita Assistida por Metamodelos

Este capítulo apresenta o desenvolvimento e testes do algoritmo de otimização multi-objetivo restrito assistido por metamodelos MO-CORS, mantendo a estrutura base do CORS mono-objetivo, de busca global baseada em restrições de distância controlada pelos valores de um vetor de busca padrão. A adaptação foi feita aproveitando a capacidade de convergência do algoritmo CORS nos problemas mono-objetivo para a exploração dos extremos da frente de Pareto. Além disso, serão introduzidos os aspectos envolvidos próprios da natureza dos problemas de otimização multi-objetivo, tais como: dominância, espaço objetivo, conjunto não dominado e frente de Pareto. Além dos aspectos mencionados, estão envolvidas no algoritmo MO-CORS técnicas orientadas na busca da qualidade da solução de Pareto, tais como: diversidade, capacidade e taxa de sucesso, todos esses aspectos tratados no capítulo 2. Eles foram introduzidos dentro do algoritmo MO-CORS através da implementação de três problemas auxiliares de otimização, de técnicas de seleção de pontos de avaliação da função custosa e de critérios de exploração e busca nos espaços de decisão e objetivo.

Uma vez desenvolvido e implementado o algoritmo MO-CORS, ele foi testado inicialmente num grupo de funções de teste mono-objetivo de Dixon e Szegö (1978), (Branin, Goldstein-Price, Hartmann³ e Hartmann⁶) para garantir o funcionamento do algoritmo neste tipo de problemas e, ao mesmo tempo, verificar que o algoritmo CORS é um caso particular de MO-CORS, em que o número de funções objetivo é um, $N_{obj} = 1$. Neste mesmo caso foi introduzido também um algoritmo para a determinação automática do parâmetro de forma c das Funções de Base Radial dentro do contexto do algoritmo MO-CORS. Depois, neste capítulo, o algoritmo foi testado num grupo selecionado de problemas de teste multi-objetivo obtidos de Huband *et al.* 2006, foi avaliado seu desempenho usando as métricas definidas no capítulo 2.

4.1. Definição Geral do Algoritmo MO-CORS

Em termos gerais o MO-CORS, é um algoritmo assistido por metamodelos, que pode ser usado para gerenciar o processo de otimização de problemas mono-objetivo e multi-objetivo tipo direto, em que as funções de avaliação são computacionalmente caras. Isto é alcançado através do aprimoramento dinâmico de metamodelos, usados em três problemas de otimização auxiliares para geração de pontos candidatos, de onde serão selecionados os pontos para avaliação custosa em cada iteração, baseado em seis critérios de seleção local e global.

O algoritmo é projetado essencialmente para problemas irrestritos, mas aplica restrições de distância próprias do algoritmo CORS, utilizadas como técnica iterativa de exploração local e global do espaço de decisão. Estas restrições são introduzidas dentro do problema de otimização auxiliar 1, em que os pontos obtidos tenderiam a ser aqueles que exploram os extremos da frente de Pareto, tendo em conta a garantia de convergência a ótimos globais de cada função objetivo demonstrada para CORS, (REGIS e SHOEMAKER, 2005). Na estratégia, são agregados os problemas de otimização auxiliares 2 e 3, orientados ao preenchimento dos espaços vazios do conjunto não dominado de cada iteração.

A seguir, são descritos os passos do algoritmo MO-CORS com base no diagrama de fluxo geral dos AOAM-ACDM (Capítulo 3, Figura 3.1), bem como alguns procedimentos dedicados a manter a filosofia básica dos AOAM e aprimorar sua eficácia e desempenho. A meta principal é a diminuição do custo computacional, através da redução do número de avaliações da função cara, mantendo ou melhorando as características de diversidade, distribuição, taxa de sucesso e convergência dos resultados do processo em relação aos dos obtidos pelo algoritmo de otimização NSGAI.

4.2. Descrição dos aspectos relevantes dos AOAM dentro do algoritmo MO-CORS

4.2.1. Entrada de dados

Os dados de entrada do algoritmo MO-CORS são aqueles parâmetros mantidos constantes durante todo processo. A seguir apresenta-se uma lista com a identificação dos dados de entrada:

- Número de variáveis dos pontos no espaço de decisão (n_{var}).
- Números de níveis para a construção do plano de experiências do banco de dados inicial (N_{niv}), (dados requeridos para aplicação das técnicas PFC e PFI).

- Número de pontos para a construção do plano de experiências do banco de dados inicial (N_{hcl}), (dado requerido para aplicação da técnica HCL).
- Número de níveis para a construção do plano de experiências da matriz de cobertura (N_{cover}). (Ver seção 4.2.4)
- Vetor de busca padrão β .
- Número de funções objetivo (N_{obj}).
- Limites superiores e inferiores de cada variável (x_{max}^i e x_{min}^i).
- Número identificador da RBF de cada função objetivo (n_{RBF}).
- Número identificador de minimização [0] ou maximização [1]

4.2.2. Construção do banco de dados inicial.

O algoritmo MO-CORS tem a versatilidade de poder usar três técnicas para a execução de planos de experiências, como: Plano Fatorial Completo (PFC), Plano Fatorial Incompleto (PFI) e Hipercubo Latino (HCL), definidas na seção 3.3 para a construção do banco de dados inicial no espaço de decisão. Em princípio, o critério de seleção de uso de alguma delas se volta para a resolução dos problemas de teste e aplicações desta pesquisa, e depende do número de variáveis de projeto envolvidas no problema (n_{var}). Sendo assim, a técnica HLC pode ser usada em forma irrestrita, com o número de pontos totais no banco de dados inicial seguindo a regra $N_{hcl} = 2(n_{var} + 1)$, (MÜLLER, 2017). No caso em que $n_{var} \leq 5$, pode-se utilizar também as técnicas PFC ou PFI. Elas foram codificadas em uma função computacional (f_{BD}), baseada nos procedimentos descritos no capítulo 3. Para a técnica HCL, foi utilizada uma ferramenta própria da linguagem de programação adotada para a codificação do algoritmo (Matlab). A função f_{BD} , recebe uma matriz com os valores dos limites laterais x_j^{min} e x_j^{max} e o número de níveis N_{niv}^j de cada variável de projeto j , e retorna uma matriz que contém os pontos do banco de dados inicial S_0 , de ordem $N_{exp} \times n_{var}$, em que N_{exp} é o número de pontos no banco de dados:

$$S_0 = f_{BD}(x_j^{min}, x_j^{max}, N_{niv}^j, n_{var}); j = 1, \dots, n_{var} \quad (4.1)$$

No caso do uso da técnica HCL, foi utilizada a mesma função (Equação 4.1), com a única diferença do uso do número de pontos desejados (N_{hcl}) em substituição do número de níveis de cada variável N_{niv}^i .

As técnicas aqui empregadas oferecem uma distribuição uniforme, partindo de uma quantidade reduzida de pontos, que é um requerimento fundamental de todos os AOAM.

4.2.3. Critério de parada.

O MO-CORS tem a capacidade direta de testar dois tipos de critério segundo o uso pretendido do algoritmo. Para o caso de aplicação em problemas reais, tanto para problemas mono-objetivo como multi-objetivos da literatura, é comum nos AOAM, usar como critério de parada o número máximo de avaliações da função objetivo custosa ($N_{maxeval}$) (SILVA, 2011), (DATTA e REGIS, 2016), (MÜLLER, 2017) e (AKHTAR e SHOEMAKER, 2016). Isso se explica naturalmente pelo desconhecimento dos valores das soluções ótimas por parte do projetista. Nesse sentido se escreve:

$$N_{eval} \leq N_{maxeval} \quad (4.2)$$

No caso de aplicação em funções de teste, procurou-se estabelecer um critério de parada para problemas multi-objetivo como uma espécie de extensão daquele usado em problemas mono-objetivo. Primeiramente, para funções de teste em problemas mono-objetivo, o algoritmo MO-CORS, adota o mesmo critério usado por Silva, (2011), chamado critério da diferença percentual D_p . Consiste em interromper o algoritmo assim que o melhor valor obtido numa certa iteração, f_t^* , não ultrapasse a diferença (em módulo) de $D_p\%$ em relação ao valor ótimo conhecido da função de teste, f^* , conforme a Equação 4.3. Silva, (2011) usou $D_p = 1$, em seus testes.

$$|f_t^* - f^*| \leq \frac{D_p}{100} \quad (4.3)$$

Para o caso de aplicação em problemas de teste multi-objetivo, foi aplicado um critério baseado na métrica da distância geracional, (capítulo 2 seção 2.4.1) entre dois conjuntos de soluções não dominadas contíguas. Sendo conhecidos os pontos do conjunto não dominado entre duas iterações P_{t-1}^* e P_t^* , calcula-se em cada iteração t a média das mínimas distâncias \bar{d}_{min} entre os conjuntos não dominados contíguos. O processo termina quando o critério da expressão 4.4 for atingido.

$$\bar{d}_{min} \leq \varepsilon_d \quad (4.4)$$

em que, \bar{d}_{min} é calculado segundo a Equação 4.5 e ε_d , é o erro de aproximação máximo permitido para a convergência na distância geracional.

Note-se que esse critério de parada também é de convergência.

$$(\bar{d}_{min})_t = \frac{1}{|P_t^*|} \sum_{i=1}^{|P_t^*|} \min \|P_{t-1}^* - P_t^*\| \quad (4.5)$$

4.2.4. Determinação da matriz de cobertura.

O algoritmo MO-CORS preserva as bases do algoritmo CORS, onde em cada iteração é resolvido um problema de otimização auxiliar, com o metamodelo como função objetivo e restrições baseadas na máxima das mínimas distâncias Δ_t entre pontos do espaço de decisão e os pontos já avaliados no banco de dados. Torna-se então necessária uma discretização fina para cobrir amplamente o espaço de decisão (região viável), chamada de matriz de cobertura M_c , a qual se constrói no início do algoritmo usando uma das técnicas de construção de planos de experiência DOE, com um número de níveis bem maior de pontos ao utilizado no banco de dados inicial. Objetiva-se, assim garantir a maior cobertura possível da região viável durante a busca (SILVA, 2011). É importante ressaltar, que os pontos da matriz de cobertura não são avaliados nas funções custosas, eles são unicamente utilizados para o cálculo de Δ_t , que junto com as componentes do vetor de busca padrão, modulam as buscas locais e globais em cada iteração.

A Figura 4.1 representa um exemplo de matriz de cobertura num espaço de decisão bidimensional, onde os pontos pretos com borda vermelha são da matriz de cobertura, os pontos azuis 1, 2 e 3 representam os pontos do banco de dados e Δ_t , a máxima das mínimas distâncias numa iteração t .

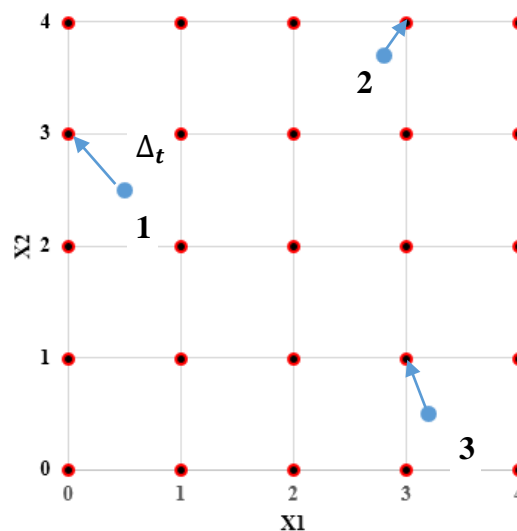


Figura 4.1: Ilustração da matriz de cobertura, pontos do banco de dados e distância Δ_t .

4.2.5. Avaliação no modelo real (funções objetivo).

Esta é a parte do algoritmo que tem maior custo computacional, já que aqui são avaliados nas funções objetivo reais (custosas), os pontos do banco de dados inicial e os pontos selecionados com o intuito de aprimorar a construção dos metamodelos em regiões de interesse.

Cada ponto \mathbf{X}_i , pertencente ao conjunto de pontos do banco de dados ($\mathcal{S}_t = \mathcal{S}_0$), na primeira iteração e ao conjunto de pontos selecionados nas iterações sucessivas $\mathcal{S}_t = \mathcal{S}_N$, é avaliado no modelo real para obter os valores das componentes dos vetores objetivo $\mathbf{f}_i(\mathbf{X}_i)$, no espaço de objetivos Ω_t . As Equações 4.6 e 4.7 (em que $i = 1, \dots, N_{exp}$) representam os vetores de decisão e objetivo em cada espaço multidimensional.

$$\mathbf{X}_i \in \mathcal{S}_t / \mathbf{X}_i = [x_1 \dots x_{n_{var}}] \quad (4.6)$$

$$\mathbf{f}_i \in \Omega_t / \mathbf{f}_i = [f_1 \dots f_{N_{obj}}] \quad (4.7)$$

Em geral, o modelo real $\Omega(\mathcal{S})$, representa as funções objetivo do problema de otimização. Esses modelos podem ser chamados de duas formas:

- *Função explícita*: define-se quando os valores das funções f_i , são obtidos algébrica ou analiticamente. As funções em problemas de teste desenvolvidos na literatura, em geral são dadas como funções explícitas.
- *Função implícita e função caixa preta*: define-se quando a função é obtida através da solução numérica de certo problema em que não se tem a solução algébrica ou analítica. Além disso, a função de caixa preta pode ser entendida como aquela no processo de otimização em que não se tem controle de todos os parâmetros e variáveis dentro da função. A solução *caixa preta* é encontrada quando a resposta é oriunda de um processo numérico como, por exemplo, aquele envolvido na obtenção de uma solução por CFD.

Nesta parte do algoritmo, é construída a matriz de dados necessários para a construção dos metamodelos M_P , de ordem $N_{exp} \times (n_{var} + N_{obj})$. Em cada iteração t , esta matriz é gerada através da união dos espaços de decisão e objetivo, como se expressa na Equação 4.8.

$$(M_P)_t = \mathcal{S}_t \cup \Omega_t \text{ ou } (M_P)_t = \begin{bmatrix} x_1^1 & \dots & x_{n_{var}}^1 & f_1^1 & \dots & f_{N_{obj}}^1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1^{N_{exp}} & \dots & x_{n_{var}}^{N_{exp}} & f_1^{N_{exp}} & \dots & f_{N_{obj}}^{N_{exp}} \end{bmatrix} \quad (4.8)$$

4.2.6. Determinação do conjunto não dominado.

Parte essencial dos AOAM para aceleração de problemas multi-objetivo baseados em métodos diretos é a busca do conjunto não dominado ótimo e de sua imagem, frente de Pareto \mathcal{F}^* , como foi discutido na seção 2.3.2. Essa busca é feita ao longo do processo através da construção iterativa dos conjuntos não dominados P^* , baseado no conceito de dominância. Para isso, foram desenvolvidas duas funções computacionais: (i) função de dominância, e (ii) a função de determinação do conjunto não dominado.

A função de dominância ($D_{A,B}$): determina a dominância ou não entre dois pontos no espaço n-dimensional, baseada na *definição 5* da seção 2.2.4, que estabelece que um ponto X_1 domina à outro X_2 , se: (i) a solução $f_k(X_1)$, não é pior que a solução $f_k(X_2)$, para todas as funções objetivo ($k = 1, \dots, N_{obj}$), e (ii) a solução de $f_k(X_1)$, é estritamente melhor que a solução de $f_k(X_2)$, em ao menos uma das funções objetivo. Uma representação esquemática da função de dominância, é mostrada na Figura 4.2.

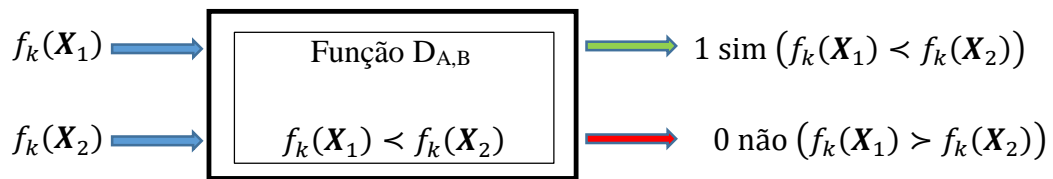


Figura 4.2: Função dominância ($D_{A,B}$).

A função conjunto não dominado (ND_{set}): Esta função foi codificada utilizando o método de atualização contínua definido na seção 2.2.6.1. Esta função recebe a matriz $(M_P)_t$, da Equação 4.8, e determina o conjunto de pontos dominados e não dominados P_t^* em cada iteração. $P_t^* \leftarrow ND_{set}[(M_P)_t]$.

É importante ressaltar que nestas funções necessárias para a determinação do conjunto não dominado, é usado o número de identificação de maximização [1] ou minimização [0], de cada função objetivo, introduzidos a priori segundo o problema analisado.

4.2.7. Normalização dos espaços de decisão e objetivo.

A fim de evitar complicações que poderiam advir do uso de diferentes unidades físicas relacionadas a cada variável e a cada função objetivo, é importante que tanto o espaço de decisão como o espaço objetivo sejam normalizados. Com isso, torna-se possível realizar operações matemáticas em ambos os espaços de forma adimensional, além de estabelecer um

equilíbrio entre grandezas de diferentes ordens de magnitude no momento da construção dos metamodelos. As componentes de cada vetor \mathbf{X} no espaço de decisão e cada vetor \mathbf{f} no espaço objetivo (Equações 4.6 e 4.7), são normalizados através das Equações 4.9 e 4.10, respectivamente, para os espaços de decisão e objetivo respectivamente. sendo x_j^{max} e x_j^{min} , os limites superiores e inferiores da cada uma das variáveis do problema e f_k^{max} e f_k^{min} os valores máximos e mínimos de cada função objetivo. Os limites máximo e mínimo das variáveis de projeto do problema permanecem fixos durante todo o processo já que eles representam as fronteiras da região viável, enquanto f_k^{max} e f_k^{min} , podem variar em cada iteração pela própria natureza dos processos de otimização.

$$\hat{x}_j = \frac{x_j - x_j^{min}}{x_j^{max} - x_j^{min}}; j = 1, \dots, n_{var} \quad (4.9)$$

$$\hat{f}_k = \frac{f_k - f_k^{min}}{f_k^{max} - f_k^{min}}; k = 1, \dots, N_{obj} \quad (4.10)$$

Assim é construída a matriz de dados normalizados em cada iteração $(\hat{M}_P)_t$. Sendo que, $(\hat{M}_P)_t = \hat{\mathbf{S}}_t \cup \hat{\mathbf{\Omega}}_t$, onde $\hat{\mathbf{S}}_t$ e $\hat{\mathbf{\Omega}}_t$, são os espaços de decisão e objetivo normalizados.

4.2.8. Construção do metamodelo.

Uma vez normalizado o banco de dados, são construídos os metamodelos de cada uma das funções objetivo. No algoritmo MO-CORS pode ser usada qualquer uma das RBF, listadas na Tabela 4.1, com ou sem polinômio de extensão linear, para a construção dos metamodelos segundo o procedimento estabelecido no Apêndice A. Além disso, o algoritmo MO-CORS permite a utilização de uma RBF diferente para a modelagem de cada função objetivo, isto é feito indicando um número de identificação de cada RBF (n_{RBF}) para cada função objetivo. Essa versatilidade do algoritmo foi implementada a fim de pesquisar a influência do tipo de RBF nas aplicações de MO-CORS.

Outro aspecto importante na versão atual de MO-CORS foi a implementação de um algoritmo para a escolha automática do parâmetro de forma c (Apêndice B), em caso do uso da RBF Gaussiana, multiquádrica ou multiquádrica inversa. Essa opção está disponível unicamente para problemas mono-objetivo. Ela foi motivada pelos resultados obtidos por Diaz *et al*, 2018 onde se verificou o melhor desempenho das RBF dependentes do parâmetro de forma, em relação às aquelas não dependentes dele.

Para os testes e aplicações multi-objetivo realizadas nesta pesquisa foi utilizada a RBF cúbica com polinômio de extensão linear para todas as funções objetivo.

Tabela 4.1: Número de identificação da RBF.

Nome da RBF	n_{RBF}
Gaussiana	1
Multiquádrica	2
Multiquádrica inversa	3
Linear	4
Cúbica	5
<i>Thin Plates Spline</i>	6

4.2.9. Problemas de otimização auxiliar.

Embora o esquema clássico dos AOAM se refira a um único problema auxiliar para indicar pontos candidatos à avaliação custosa, verificam-se tendências na literatura referente aos AOAM no sentido de se gerenciar problemas de otimização multi-objetivo com o uso de vários problemas de otimização auxiliar. Por exemplo, Akhtar e Shomaker, (2016) e Müller, (2017), utilizam dois e três problemas de otimização auxiliares respectivamente, com o objetivo de explorar os espaços de decisão e objetivo além de tentar promover diversidade e cobertura na frente de Pareto final. Neste sentido, no desenvolvimento do algoritmo MO-CORS foi adotada uma metodologia que consiste na resolução de três problemas de otimização auxiliar. O primeiro é o próprio problema auxiliar usado no algoritmo CORS, com restrições de distância aplicadas ao metamodelo de cada função objetivo, contudo, tornou-se necessário um algoritmo de otimização essencialmente multi-objetivo do tipo direto e que se reduzisse a um algoritmo mono-objetivo, quando fosse o caso. Nesse sentido, o algoritmo selecionado foi o NSGA II, considerando também o bom desempenho atingido por ele em problemas de aerodinâmica de asas (DIAZ *et al*, 2017) e (DIAZ *et al*, 2018.a), cujos resultados são mostrados no Apêndice C. O segundo problema de otimização auxiliar, chamado de “*gap*”, (AKHTAR e SHOMAKER, 2016) faz uma busca local nas vizinhanças dos pontos do conjunto não dominado de cada iteração. Para este problema foi utilizado também o NSGA II como algoritmo de otimização. O terceiro problema auxiliar, analogamente ao segundo busca preencher os espaços vazios da imagem do conjunto não dominado (frente) presente no banco de dados em cada iteração. Isso se faz por meio de uma técnica baseada na construção de outro metamodelo no espaço objetivo, buscando aproximar o formato da frente. Através de um processo de otimização, procura-se então um vetor objetivo com potencial para preencher os espaços maiores entre pontos dessa frente.

A seguir se descrevem os problemas de otimização auxiliar que serão utilizados dentro do algoritmo MO-CORS:

Problema de otimização auxiliar 1 (Busca CORS).

Este problema segue a mesma busca global desenvolvida por Regis e Shomaker (2005) no CORS, baseada em restrições de distância que modulam buscas locais e globais, ao se percorrer ciclicamente o vetor de busca padrão β . A diferença é o contexto multi-objetivo: o otimizador é o NSGA II e as restrições são agora aplicadas a cada metamodelo de cada função objetivo. O objetivo principal deste problema de otimização é aportar uma busca dos extremos da frente de Pareto, além de assegurar que o algoritmo MO-CORS convirja a ótimos globais no caso de aplicações mono-objetivo. Assim é calculado o primeiro conjunto de pontos candidatos P_{can1} , em que $P_{can1} \leftarrow (P_t^*)_1$ e a frente de Pareto resultante da resolução do problema seguinte através do algoritmo NSGA II.

$$\begin{aligned} & \text{Otimizar } \tilde{f}_k(\mathbf{X}); k = 1, \dots, N_{obj} \\ \text{sujeito a:} & \quad \|\mathbf{X} - \mathbf{X}_i\| \geq \beta_c \Delta_t \quad \forall \mathbf{X} \in D, \quad i = 1, 2, \dots, N_{exp} + t - 1 \\ \text{onde:} & \quad \Delta_t = \max_{\mathbf{X} \in D} \min_{1 \leq i \leq N_{exp} + t - 1} \|\mathbf{X}_c - \mathbf{X}_i\| \end{aligned}$$

Problema de otimização auxiliar 2 (Busca Local).

Akhtar e Shoemaker (2016) apresentam em seu trabalho este problema de otimização auxiliar com a finalidade de fazer uma busca nas proximidades dos pontos mais isolados do conjunto não dominado presente no banco de dados em cada iteração. Este tipo de busca é considerada como local e utiliza a métrica da distância de aglomeração para estabelecer os limites laterais do problema de otimização. A formulação matemática é mostrada a seguir:

$$\begin{aligned} & \text{Otimizar } \tilde{f}_k(\mathbf{X}) \\ \text{sujeito a:} & \quad (\mathbf{X}^{crow} - \varepsilon_c) \leq \mathbf{X} \leq (\mathbf{X}^{crow} + \varepsilon_c) \end{aligned}$$

Este problema é conhecido como “gap” onde ε_c é um valor na ordem de 10^{-3} , chamado o raio de “gap” e \mathbf{X}^{crow} é o ponto de maior distância de aglomeração. Assim é calculado um segundo conjunto de pontos candidatos P_{can2} , em que $P_{can2} \leftarrow (P_t^*)_2$, é a frente de Pareto resultante da resolução do problema através do algoritmo NSGA II.

Problema de otimização auxiliar 3 (Busca de vetor alvo).

A estratégia aplicada neste problema consiste numa busca local, similar à aplicada por Müller (2017) com o nome de “estratégia do vetor alvo”. Esta estratégia é composta por duas partes. Na **primeira parte** é construído um novo metamodelo \tilde{f}_{P^*} que aproxima a imagem do conjunto não dominado (frente) em cada iteração usando uma RBF linear com polinômio de

extensão. O domínio S_{P^*} o conjunto formado pela matriz de valores de todas as funções objetivo normalizadas menos uma $\hat{f}_{q,l}$, em que $q = 1, \dots, |P^*|$ e $l = 1, \dots, N_{obj} - 1$. A imagem do domínio Ω_{P^*} , formado pelo vetor que contém os valores da última função objetivo $\hat{f}_{q,N_{obj}}$. Isto é expressado nas Equações 4.11 e 4.12

$$S_{P^*} = \hat{f}_{q,l} \text{ sendo } \hat{f}_{q,l} = \begin{bmatrix} \hat{f}_{1,1} & \cdots & \hat{f}_{1,(N_{obj}-1)} \\ \vdots & \ddots & \vdots \\ \hat{f}_{|P^*|,1} & \cdots & \hat{f}_{|P^*|,(N_{obj}-1)} \end{bmatrix} \quad (4.11)$$

$$\Omega_{P^*} = \hat{f}_{q,N_{obj}} \text{ sendo } \hat{f}_{q,N_{obj}} = \begin{bmatrix} \hat{f}_{1,N_{obj}} \\ \vdots \\ \hat{f}_{|P^*|,N_{obj}} \end{bmatrix} \quad (4.12)$$

O metamodelo \tilde{f}_{P^*} , define qualquer vetor $\boldsymbol{\tau}$ do espaço objetivo $\boldsymbol{\tau} = \tilde{f}_{P^*}(S_{P^*}, \Omega_{P^*})$, que este contido no conjunto não dominado P^* . Depois de construído o metamodelo, é definida uma função $\delta_{\boldsymbol{\tau}}$ que calcula a distância mínima entre o vetor variável $\boldsymbol{\tau}$ e os pontos do conjunto não dominado no espaço objetivo $F^* = f(P^*)$, quer dizer, $\delta_{\boldsymbol{\tau}}(\boldsymbol{\tau}, F^*)$. Resolve-se um problema de otimização onde é maximizada a função $\delta_{\boldsymbol{\tau}}(\boldsymbol{\tau}, F^*)$, sujeita a restrições laterais formados pelos valores máximos e mínimos das funções normalizadas, $(\hat{f}_k^{max})^*$ e $(\hat{f}_k^{min})^*$ do conjunto não dominado e uma restrição de tolerância $\varepsilon_{\boldsymbol{\tau}}$ que descarta soluções já avaliadas em iterações anteriores.

$$\begin{aligned} & \max \delta_{\boldsymbol{\tau}}(\boldsymbol{\tau}, F^*) \\ \text{Sujeito a:} & \\ & \|\boldsymbol{\tau}_t - \Gamma_t\| > \varepsilon_{\boldsymbol{\tau}} \\ & (\hat{f}_k^{min})^* \leq \boldsymbol{\tau}_t \leq (\hat{f}_k^{max})^*; k = 1, \dots, N_{obj} \end{aligned}$$

em que, $\varepsilon_{\boldsymbol{\tau}} \ll 1$ e Γ_t é o conjunto de vetores $\boldsymbol{\tau}_t$, já avaliados em iterações anteriores.

Obtida a solução do problema de maximização anterior ($\boldsymbol{\tau}_t = \boldsymbol{\tau}^*$), o vetor obtido é chamado de *vetor alvo* e pertence ao espaço de objetivos normalizado da iteração ($\boldsymbol{\tau}^* \in \Omega_t$). Em seguida esta solução é agregada ao conjunto Γ_t , visando que a solução não se repita em iterações seguintes. O vetor $\boldsymbol{\tau}_t$ pode ser então definido como um ponto que preenche os espaços vazios da frente durante uma iteração qualquer.

A **segunda parte** da estratégia consiste em obter o vetor $\mathbf{X}_{\boldsymbol{\tau}^*}$ que minimiza a distância euclidiana $\Delta_{\boldsymbol{\tau}} = \|\tilde{f}(\mathbf{X}) - \boldsymbol{\tau}^*\|$, resolvendo um segundo problema de otimização mono-objetivo que se mostra a seguir:

$$\min \Delta_{\boldsymbol{\tau}}(\mathbf{X}, \boldsymbol{\tau}^*)$$

Sujeito a:

$$x_j^{min} \leq \mathbf{X} \leq x_j^{max}; j = 1, \dots, n_{var}$$

Neste problema, é selecionado um único ponto \mathbf{X}_τ^* o qual é chamado de P_3 . Nos dois problemas de otimização foi utilizado um algoritmo de otimização global mono-objetivo baseado em programação não linear.

Eliminação de pontos próximos aos contidos no banco de dados

Os conjuntos candidatos P_{can1} e P_{can2} obtidos nos problemas de otimização auxiliar 1 e 2, são depurados através de um processo de **filtragem** inicial, onde são eliminados dos conjuntos os pontos que estejam muito próximos dos pontos já avaliados do banco de dados no intuito de evitar o mal condicionamento da matriz de coeficientes durante a construção dos metamodelos.

4.2.10. Seleção de pontos para avaliação na função custosa.

Antes de fazer referência à metodologia de seleção de pontos é importante ressaltar que no caso de problemas mono-objetivo em que $N_{obj} = 1$, não faria sentido a aplicação dos problemas de otimização auxiliares 2 e 3, já que o foco deles é procurar atingir a cobertura e a diversidade da frente de Pareto, que são aspectos próprios dos problemas multi-objetivo. Então, nesse caso, os conjuntos P_{can2} e P_3 são vazios, ($P_{can2} = \{P_3\} = \{\emptyset\}$), e o único ponto candidato \mathbf{X}^* é obtido através da solução do problema de otimização auxiliar 1. Por isso a metodologia a seguir neste caso é muito simples: o conjunto de pontos selecionados \mathcal{S}_t para avaliação custosa é formado pelo único ponto candidato \mathbf{X}^* , ($\mathcal{S}_t \leftarrow \{\mathbf{X}^*\}; |\mathcal{S}_t| = 1$).

No caso dos problemas multi-objetivo a metodologia de seleção de pontos consiste no estudo dos critérios que permitam a obtenção de um conjunto de pontos \mathcal{S}_t para a próxima avaliação na função custosa. O bom desenvolvimento destes métodos é de vital importância para garantir a abordagem dos aspectos próprios destes problemas, como: a exploração abrangente de pontos nos espaços de decisão e objetivo, a diversificação dos pontos sobre a frente de Pareto, a exploração dos extremos dessa frente e a convergência rápida do algoritmo. Isto, com um número consideravelmente menor de avaliações da função computacionalmente custosa em comparação com um algoritmo de otimização multi-objetivo normal (tal como o NSGA II aplicado diretamente). Neste sentido, a metodologia de seleção desenvolvida nesta pesquisa ressalta a utilização de três critérios principais de seleção aplicados aos conjuntos de pontos candidatos dos problemas de otimização auxiliares 1 e 2 (P_{can1} e P_{can2}), que abordam os aspectos nomeados, além do uso do problema de otimização auxiliar 3 para a seleção de um único ponto P_3 , orientado na busca de preencher espaços não explorados da frente de Pareto.

Em geral para os conjuntos de pontos candidatos, foram utilizados três critérios de seleção principal e um critério único de geração de pontos auxiliares.

Critérios de seleção principal

- Seleção dos extremos do conjunto não dominado (P_{ex})
- Seleção do ponto baseada na máxima das mínimas distâncias no espaço de decisão (P_{de})
- Seleção do ponto baseada na máxima das mínimas distâncias no espaço de objetivos (P_{ob})

Ao mesmo tempo, contempla-se a eventual necessidade de outros pontos para construção dos metamodelos durante alguma iteração, mantendo uma quantidade fixa de pontos a serem selecionados, é gerado um conjunto de pontos auxiliares (P_{aux}), onde eventualmente são selecionados a quantidade de pontos necessária que complete a quantidade preestabelecida em cada caso, mostrada na Tabela 4.2. O **critério único de geração do conjunto de pontos auxiliares** é baseado na amostragem aleatória no espaço de decisão do conjunto de pontos não dominados, descrito no capítulo 3 seção 3.6.1, mas tendo em conta que neste caso é utilizado para geração de um conjunto de pontos, em vez de seleção de um ponto único.

Tabela 4.2: Número máximo de pontos selecionados.

Conjunto de pontos candidatos	Número máximo de pontos selecionados de cada critério e número de pontos auxiliares				Número máximo de pontos totais
	$ P_{ex} _{max}$	$ P_{de} _{max}$	$ P_{ob} _{max}$	$ P_{aux} $	
P_{can1}	3	0	1	$4 - (P_{ex} + P_{ob})$	4
P_{can2}	0	1	1	$2 - (P_{ex} + P_{ob})$	2
P_3	0	0	0	$1 - (P_3)$	1

No total, a quantidade de pontos selecionados em cada iteração é de sete (07), ($|\mathcal{S}_t| = 7$), em que, há que se considerar que em algum momento os conjuntos P_{can1} e P_{can2} poderiam ficar vazios ou com quantidade insuficiente de pontos para atingir o número de pontos preestabelecido, quando então é usado o conjunto auxiliar P_{aux} .

Seleção dos extremos do conjunto não dominado.

Este critério é aplicado unicamente aos pontos candidatos do problema de otimização auxiliar 1, o qual consiste na aplicação do algoritmo CORS original de Regis e Shoemaker (2005), estendido em MO-CORS para problemas multi-objetivo. Conceitualmente, cada ponto extremo do conjunto não dominado cumpre com que ao menos um dos valores de seu vetor objetivo tem o melhor valor de todos os pontos nessa função. Em princípio, então, o número de pontos selecionados segundo esta premissa deveria ser igual ao número de objetivos

($|P_{ex}| = N_{obj}$). Isto aumentaria consideravelmente o número de pontos de avaliação custosa nos problemas com um grande número de funções objetivo. Para evitar isso, para $N_{obj} > 1$ foi estabelecida como referência uma quantidade fixa de 3 pontos selecionados com este critério. Então são possíveis, 4 diferente situações no MO-CORS para a tomada de decisões através deste critério:

- Caso $N_{obj} = 1$, o problema é mono-objetivo e, por tanto, é aplicado unicamente o problema auxiliar 1 para obter o único ponto candidato através do algoritmo CORS original. Nesse caso então, a aplicação deste critério é equivalente à obter o melhor ponto da iteração. Isto garante que CORS é um caso particular de MO-CORS quando $N_{obj} = 1$
- Caso $N_{obj} = 2$, são selecionados em primeiro lugar os pontos extremos de cada uma das duas funções objetivo e um terceiro ponto com a maior distância de aglomeração, obtido do conjunto de pontos candidatos auxiliares P_{aux} .
- Caso $N_{obj} = 3$, são selecionados os pontos extremos de cada uma das três funções objetivo.
- Caso $N_{obj} > 3$, são selecionados os três pontos extremos com a maior distância de aglomeração dentre os pontos extremos de cada função objetivo. Esta distância é calculada segundo o procedimento descrito no capítulo 2, Equação 2.33.
- Caso o número de pontos extremos seja menor que o número de funções objetivo, em qualquer caso, são selecionados três pontos do conjunto de candidatos auxiliares P_{aux} .

A seguir descreve-se o algoritmo do critério.

Passo 1: Extrair os vetores objetivo dos pontos de P_{can1} , e alocar na matriz $[f_{can1}]_{m,n}$ de ordem $m \times n$, em que $m = |P_{can1}|$ e $n = N_{obj}$

Passo 2: Determinar o valor do índice m que identifica a posição do melhor valor de cada coluna n , $(I_{m,n})$ da matriz $[f_{can1}]_{m,n}$.

Passo 3: Extrair cada elemento de posição $I_{m,n}$, fora da matriz P_{can1} , e alocar eles numa matriz auxiliar P_{ex}^{aux} , ou $P_{ex}^{aux} \rightarrow P_{can1}(I_{m,n})$.

Passo 4: No caso que $N_{obj} = 1$, $P_{ex} \rightarrow P_{ex}^{aux}$ e vai ao passo 10

Passo 5: No caso que $N_{obj} = 2$ e $|P_{ex}^{aux}| = 2$ Completa-se a matriz P_{ex}^{aux} com um (01) ponto da matriz auxiliar P_{aux} e fazer $P_{ex} \rightarrow P_{ex}^{aux}$; vai ao passo 10

Passo 6: No caso que $N_{obj} = 2$ e $|P_{ex}^{aux}| < 2$ Completa-se a matriz P_{ex}^{aux} com pontos da matriz auxiliar P_{aux} ate que $|P_{ex}^{aux}| = 3$ e fazer $P_{ex} \rightarrow P_{ex}^{aux}$; vai ao passo 10

Passo 7: No caso que $N_{obj} = 3$ e $|P_{ex}^{aux}| = 3$ $P_{ex} \rightarrow P_{ex}^{aux}$ e vai ao passo 10

Passo 8: No caso que $N_{obj} \geq 3$ e $|P_{ex}^{aux}| < 3$ Completa-se a matriz P_{ex}^{aux} com pontos da matriz auxiliar P_{aux} ate que $|P_{ex}^{aux}| = 3$ e fazer $P_{ex} \rightarrow P_{ex}^{aux}$; vai ao passo 10

Passo 9: No caso que $N_{obj} > 3$ e $|P_{ex}^{aux}| > 3$, calcula-se a distância de aglomeração d_{ag} de todos os pontos do P_{ex}^{aux} , e são selecionados os três com maior d_{ag} . $P_{ex} \rightarrow P_{ex}^{aux} \rightarrow P_{mag}$

Passo 10: Fim.

Seleção do ponto baseada na máxima das mínimas distâncias no espaço de decisão (P_{de}) e do espaço objetivo (P_{ob}).

Nestes critérios, a seleção é baseada na máxima das mínimas distâncias entre os pontos do conjunto não dominado (candidatos) e os pontos do banco de dados, no espaço de decisão e no espaço objetivo. A seguir descreve-se o algoritmo para aplicação do critério de seleção baseado na máxima das mínimas distâncias no espaço de decisão, o qual é aplicado unicamente para a seleção de 1 ponto dos candidatos do problema de otimização auxiliar 2, seguindo o procedimento explicado no capítulo 3 seção 3.6.4.

Algoritmo para seleção no espaço de decisão:

Passo 1: Extrair os vetores de decisão dos pontos de P_{can2} e alocar na matriz X_{can2} , de ordem $m \times n$, em que $m = |P_{can2}|$ e $n = n_{var}$

Passo 2: Extrair os vetores de decisão X_{BD} da matriz do banco de dados normalizada \hat{M}_P .

Passo 3: Calcular o vetor de distâncias mínimas entre os pontos candidatos X_{can2} e os pontos X_{BD} do banco de dados, $d_{min}^2 = \|X_{BD} - X_{can2}\|$

Passo 4: Determinar a posição $I_{dmaxmin}^2$ dos pontos da máxima das mínimas distâncias do conjunto de candidatos P_{can2} onde são cumpridas as condições $\max(d_{min}^2)$.

Passo 5: $P_{de}^2 \rightarrow P_{can2}(I_{dmaxmin}^2)$.

Passo 6: Fim

A seguir, descreve-se o algoritmo para aplicação do critério de seleção baseado na máxima das mínimas distâncias no espaço de objetivos, o qual é aplicado para a seleção de 1 ponto dos candidatos do problema de otimização auxiliar 1, e 1 ponto dos candidatos do problema de otimização auxiliar 2 seguindo o procedimento explicado no capítulo 3 seção 3.6.5.

Algoritmo para seleção no espaço objetivo:

Passo 1: Extrair os vetores objetivo dos pontos de P_{can1} e P_{can2} e alocar na matriz f_{can1} e f_{can2} , de ordem $m \times n$, em que $n = N_{obj}$ e $m = |P_{can1}|$, para a seleção dos pontos do problema auxiliar 1 e $m = |P_{can2}|$ para a seleção dos pontos do problema auxiliar 2.

Passo 2: Extrair os vetores objetivo f_{BD} , da matriz do banco de dados normalizada \hat{M}_P

Passo 3: Calcular o vetor de distâncias mínimas entre os pontos candidatos f_{can1} e f_{can2} e os pontos f_{BD} , do banco de dados, $\mathbf{d}_{min}^1 = \|f_{BD} - f_{can1}\|$ e $\mathbf{d}_{min}^2 = \|f_{BD} - f_{can2}\|$

Passo 4: Determinar as posição $I_{dmaxmin}^1$ e $I_{dmaxmin}^2$, dos pontos da máxima das mínimas distâncias no espaço objetivo dos conjuntos de candidatos P_{can1} e P_{can2} , onde são cumpridas as condições $\max(\mathbf{d}_{min}^1)$ e $\max(\mathbf{d}_{min}^2)$.

Passo 5: $P_{ob}^1 \rightarrow P_{can1}(I_{dmaxmin}^1)$ e $P_{ob}^2 \rightarrow P_{can2}(I_{dmaxmin}^2)$.

Passo 6: Fim

É importante ressaltar que para todos os algoritmos os conjuntos P_{de} , P_{ob} e P_{ex} estão formados unicamente pelas coordenadas no espaço de decisão dos pontos selecionados segundo cada critério. Da mesma forma que que no critério dos pontos extremos, na medida em que se atinge convergência, as quantidades de pontos em cada conjunto podem ficar insuficientes, sendo então completada a seleção com pontos da matriz auxiliar P_{aux} .

Conjunto de pontos selecionados no espaço normalizado (\mathcal{S}_N).

Aqui se juntam os pontos selecionados após a aplicação de cada critério de seleção em cada conjunto de pontos candidatos e o ponto selecionado no problema de otimização auxiliar 3, (Equações 4.13, 4.14 e 4.15)

$$\mathcal{S}_N^1 = P_{ex}^1 \cup P_{ob}^1 \quad (4.13)$$

$$\mathcal{S}_N^2 = P_{de}^2 \cup P_{ob}^2 \quad (4.14)$$

$$\mathcal{S}_N^3 = P_3 \quad (4.15)$$

Finalmente o conjunto de pontos selecionados normalizados é:

$$\mathcal{S}_N = \mathcal{S}_N^1 \cup \mathcal{S}_N^2 \cup \mathcal{S}_N^3 \quad (4.16)$$

4.2.11. Avaliação do conjunto de pontos selecionados na função custosa.

Antes de avaliar o conjunto de pontos selecionados na função custosa, é preciso desfazer a normalização das variáveis de decisão. Os valores das variáveis de decisão dos pontos selecionados são transformados da forma normalizada \mathcal{S}_N à forma sem normalização \mathcal{S}_R (ordem de grandeza real) através da Equação 4.17.

$$(x_j)_{\mathcal{S}_R} = (x_j)_{\mathcal{S}_N} (x_j^{max} - x_j^{min}) + x_j^{min}; j = 1, \dots, n_{var} \quad (4.17)$$

Uma vez representados em termos das dimensões originais das variáveis de decisão, os pontos selecionados são avaliados nas funções objetivo, obtendo a matriz de pontos selecionados $\mathcal{S} = [\mathcal{S}_R, f_k(\mathcal{S}_R)]$, com; $k = 1, \dots, N_{obj}$; $|\mathcal{S}|$ representa o número de pontos selecionados durante a iteração.

4.2.12. Atualização do banco de dados.

Nesta parte do algoritmo, o banco de dados $(M_P)_t$, é atualizado com o conjunto de pontos selecionados no procedimento anterior (Equação 4.18).

$$(M_P)_t = (M_P)_t \cup (\mathcal{S})_t \quad (4.18)$$

Após a atualização do banco de dados o número de avaliações N_{eval} , efetuadas até a iteração t , é calculado pela Equação 4.19.

$$(N_{eval})_t = N_{exp} = |(M_P)_t| \quad (4.19)$$

4.3. Algoritmo MO-CORS

Como visto até aqui, MO-CORS é um algoritmo de otimização geral, mono e multi-objetivo, assistido por técnicas de metamodelagem para o tratamento de funções computacionalmente custosas. Embora os testes até agora realizados indiquem que o MO-CORS possa ser aplicado em problemas de engenharia com até 30 variáveis de decisão, seu melhor desempenho quanto ao número de avaliações da função objetivo custosa, tem se manifestado em situações envolvendo até 12 variáveis e 3 funções objetivos. Felizmente, essa faixa contempla a grande maioria das aplicações práticas em aerodinâmica e máquinas de fluxo reportadas na literatura recente em particular aquelas realizadas no LHV da UNIFEI, (por exemplo: BOTAN, 2019; REZEK, 2019), e que constituem a principal motivação do presente trabalho.

A seguir mostra-se os passos do algoritmo de otimização assistido por metamodelagem (MO-CORS), que foi a base para o desenvolvimento do código computacional produto desta pesquisa.

Algoritmo MO-CORS:

Passo 1: *Dados de entrada.* São introduzidos os parâmetros fixos do problema de otimização e outros necessários para o funcionamento do algoritmo.

Passo 2: *Construção do banco de dados inicial do espaço de decisão.* Gera-se a matriz de banco de dados iniciais, através de técnicas de plano de experiências, fatorial completo, fatorial incompleto ou hipercubo latino. Inicializa-se o contador de iterações $t = 1$. O número de experiências ou pontos no banco de dados inicial N_{exp} é determinado dependendo da técnica utilizada ou

$$S_t = \{ \mathbf{X}_1, \dots, \mathbf{X}_{N_{exp}} \}; \forall \mathbf{X}_i \in \mathbb{R}^d; i = 1, \dots, N_{exp}; d = n_{var}$$

$$\mathbf{X} = (x_1, \dots, x_{n_{var}}); \forall x_j \in \mathbb{R}; j = 1, \dots, n_{var}$$

Passo 3: *Construção da matriz de cobertura normalizada.* Constrói-se a matriz de cobertura, através de uma das técnicas para a construção de planos de experiências (Ver seção 4.2.4). Esta matriz deve cobrir a maior quantidade possível de pontos no espaço de decisão, porém no caso do uso de PFC, o número de níveis N_{niv} de cada variável deve de ser tal que o número de pontos discretizados esteja na ordem de $8000 \times (n_{var} + 1)$.

Passo 4: *Avaliação da função custosa no banco de dados inicial.* Os pontos do banco de dados inicial, são avaliados nas funções objetivo custosas, $\Omega_t = \mathbf{f}(S_t)$

$$\Omega_t = \{ \mathbf{f}_1, \dots, \mathbf{f}_{N_{exp}} \}; \forall \mathbf{f}_i \in \mathbb{R}^K; i = 1, \dots, N_{exp}; K = N_{obj}$$

$$\mathbf{f}_i = (f_1, \dots, f_{N_{obj}}); \forall f_k \in \mathbb{R}; k = 1, \dots, N_{obj}$$

Matriz de banco de dados: $(M_p)_t = \{S_t, \Omega_t\}$

Passo 5: *Determinação do conjunto não dominado do banco de dados inicial.* Aplicação dos critérios de dominância ao banco de dados durante a iteração. $P_t^* = ND_{set} \left((M_p)_t \right)$.

Início do ciclo iterativo (Critério de parada)

Inicia-se um processo iterativo contínuo até cumprir com o *critério de parada*. O algoritmo continua em execução, até que se cumpram alguns dos seguintes critérios dependendo do requerimento do usuário:

Critérios de parada para funções de teste, caso $N_{obj} = 1$:

- $(N_{eval} \leq N_{maxeval})$
- $|f_t^* - f^*| \leq \frac{D_p}{100}$

Critério de parada para aplicações de engenharia, caso $N_{obj} = 1$:

- $(N_{eval} \leq N_{maxeval})$

Cr terios de parada, tanto para problemas de teste, como para aplica es de engenharia, sim $N_{obj} > 1$:

- $(N_{eval} \leq N_{maxeval})$
- $\bar{d}_{min} \leq \varepsilon_d$

Passo 6: *Normaliza o dos espa os de decis o e objetivo do banco de dados.* S o normalizadas todas as vari veis de decis o e os valores das fun es objetivo. $(\widehat{M}_p)_t = \{\widehat{S}_t, \widehat{\Omega}_t\}$

$$\widehat{S}_t = \{\widehat{X}_1, \dots, \widehat{X}_{N_{exp}}\} \text{ com } \widehat{X}_i = \widehat{x}_j$$

$$\widehat{\Omega}_t = \{\widehat{f}_1, \dots, \widehat{f}_{N_{exp}}\} \text{ com } \widehat{f}_i = \widehat{f}_k.$$

$$\widehat{x}_j = \frac{x_j - x_j^{min}}{x_j^{max} - x_j^{min}}; j = 1, \dots, n_{var}$$

$$\widehat{f}_k = \frac{f_k - f_k^{min}}{f_k^{max} - f_k^{min}}; k = 1, \dots, N_{obj}$$

Passo 7: *Constru o dos metamodelos para os problemas de otimiza o auxiliares 1 e 2.*

Utilizando a matriz $(\widehat{M}_p)_t = \{\widehat{S}_t, \widehat{\Omega}_t\}$, s o constru dos metamodelos para cada fun o objetivo, usando a RBF requerida pelo usu rio, atrav s do n mero de identifica o n_{RBF} da Tabela 4.1. Nesta vers o de MO-CORS, unicamente se $N_{obj} = 1$,   poss vel segundo seja requerido pelo usu rio, a otimiza o do par metro de forma c sempre que a RBF utilizada precise dele ($n_{RBF} \leq 3$), Ver Ap ndice B.

No caso de n o se requerer uso de polin mio linear:

Determina o do vetor de coeficientes de influ ncia $\{\alpha\}^T$, atrav s da resolu o do seguinte sistema:

$$[\varphi]\{\alpha\} = \{\widehat{\Omega}_t\}$$

Constru o do vetor de fun es metamodeladas:

$$\text{onde: } \tilde{f}(X) = \alpha_i \varphi_{X,i}^T$$

No caso de se requerer uso de polin mio linear:

A extens o com polin mio linear requer que   matriz de base de dados no espa o de decis o seja acrescentada uma coluna com uns, e dizer: $\widehat{S}_t^1 = \{\widehat{S}_t, 1\}$.

Determina o do vetor de coeficientes de influ ncia $\{\alpha, \mu\}^T$, atrav s da resolu o do seguinte sistema:

$$\begin{bmatrix} \varphi & \widehat{S}_t^1 \\ (\widehat{S}_t^1)^T & \widehat{\Omega} \end{bmatrix} \begin{Bmatrix} \alpha \\ \mu \end{Bmatrix} = \begin{Bmatrix} \widehat{\Omega}_t \\ \widehat{\Omega}_{n_{var}+1} \end{Bmatrix}$$

Construção do vetor de funções metamodeladas:

$$\text{onde: } \tilde{f}(\mathbf{X}) = \alpha_i \boldsymbol{\varphi}_{X,i}^T + \mu_0 + \mu_1 x_1 + \mu_2 x_2 + \dots + \mu_{n_{var}} x_{n_{var}}$$

Passo 8: Problemas de otimização auxiliares. Seja \tilde{f}_k o vetor de metamodelos que aproximam as funções objetivo custosas do banco de dados. Baseado nestes metamodelos resolvem-se três problemas de otimização auxiliar para a determinação dos *pontos candidatos* para avaliação da função custosa. P_{can1} , P_{can2} . Um terceiro problema de otimização auxiliar é usado para a seleção direta de um único ponto P_3 .

Problema de otimização auxiliar 1

$$\text{Otimizar } \tilde{f}_k(\mathbf{X}); k = 1, \dots, N_{obj}$$

Sujeito a:

$$\|\mathbf{X} - \mathbf{X}_i\| \geq \beta_c \Delta_t \quad \forall \mathbf{X} \in D, \quad i = 1, 2, \dots, N_{exp} + t - 1$$

Em que:

$$\Delta_t = \max_{\mathbf{X} \in S} \min_{1 \leq i \leq N_{exp} + t - 1} \|\mathbf{X}_c - \mathbf{X}_i\|$$

O resultado obtido na resolução deste problema é o conjunto não dominado da iteração t . $(P_t^*)_1$. Os pontos pertencentes a $(P_t^*)_1$ que estão muito próximos aos pontos do banco de dados são eliminados do conjunto de pontos candidatos, obtendo-se assim $P_{can1} = (P_t^*)_1^f$.

Problema de otimização auxiliar 2:

$$\text{Otimizar } \tilde{f}_k(\mathbf{X}); k = 1, \dots, N_{obj}$$

Sujeito a:

$$(\mathbf{X}^{crow} - \varepsilon_c) \leq \mathbf{X} \leq (\mathbf{X}^{crow} + \varepsilon_c)$$

O resultado obtido na resolução deste problema é o conjunto não dominado da iteração t . $(P_t^*)_2$. Os pontos pertencentes à $(P_t^*)_2$ que estão muito próximos aos pontos do banco de dados são eliminados do conjunto de pontos candidatos, obtendo-se assim $P_{can2} = (P_t^*)_2^f$.

Problema de otimização auxiliar 3:

Este problema é resolvido em duas partes, na primeira, consiste em resolver o seguinte problema de otimização mono-objetivo, para obter o vetor $\boldsymbol{\tau}$ que preenche um espaço vazio dentro do conjunto não dominado

$$\max \min \delta_{\boldsymbol{\tau}}(\boldsymbol{\tau}, F^*)$$

Sujeito a:

$$\begin{aligned} & \|\boldsymbol{\tau}_t - \Gamma_t\| > \varepsilon_t; \\ & (\hat{f}_k^{min})^* \leq \boldsymbol{\tau}_t \leq (\hat{f}_k^{max})^*; k = 1, \dots, N_{obj} \end{aligned}$$

Na segunda parte, é resolvido o seguinte problema de otimização mono-objetivo que minimiza a distância entre um ponto $\tilde{f}(\mathbf{X})$ do espaço de objetivos e o vetor alvo

τ^* resultante da primeira parte, $\Delta_\tau = \|\tilde{f}(\mathbf{X}) - \tau^*\|$, onde é obtido o vetor \mathbf{X}_τ^* chamado de “vetor alvo”.

$$\min \Delta_\tau(\mathbf{X}, \tau^*)$$

Sujeito a:

$$x_j^{\min} \leq \mathbf{X} \leq x_j^{\max}; j = 1, \dots, n_{var}$$

A solução deste problema $P_3 = \mathbf{X}_\tau^*$, é um ponto de seleção direta para avaliação custosa.

$$\mathcal{S}_N^3 = P_3$$

Passo 9: *Seleção de pontos para a próxima avaliação custosa.* É selecionado um conjunto de pontos no espaço normalizado para cada conjunto de pontos candidatos P_{can1} e P_{can2} , segundo os critérios de seleção para exploração de extremos dos conjuntos não dominados P_{ext} , exploração do espaço de decisão P_{de} e exploração do espaço objetivo P_{ob} .

$$\mathcal{S}_N^1 = P_{ex}^1 \cup P_{ob}^1$$

$$\mathcal{S}_N^2 = P_{de}^2 \cup P_{ob}^2$$

Sendo finalmente o conjunto pontos selecionados \mathcal{S}_N :

$$\mathcal{S}_N = \mathcal{S}_N^1 \cup \mathcal{S}_N^2 \cup \mathcal{S}_N^3$$

Passo 10: *Desnormalização dos pontos selecionados para avaliação custosa.* Retorna-se às dimensões e ordens de grandeza originais das variáveis de decisão, \mathcal{S}_R :

$$(x_j)_{\mathcal{S}_R} = (x_j)_{\mathcal{S}_N} (x_j^{\max} - x_j^{\min}) + x_j^{\min}; j = 1, \dots, n_{var}$$

Passo 11: *Avaliação dos pontos selecionados na função custosa.* O conjunto de pontos selecionados \mathcal{S}_R é avaliado nas funções custosas $\mathcal{F}_S = f_k(\mathcal{S}_R)$, obtendo-se a matriz de pontos selecionados $\mathcal{S}_t = \{\mathcal{S}_R, \mathcal{F}_S\}$.

Passo 12: *Atualização do banco de dados.* O banco de dados é atualizado com os novos pontos selecionados. $(M_p)_t = (M_p)_t \cup \mathcal{S}_t$.

Passo 13: *Determinação do conjunto não dominado.* Aplicação dos critérios de dominância ao banco de dados durante a iteração. $P_t^* = ND_{set}(M_p)_t$.

Passo 14: *Cálculo do número de experiências:* $N_{exp} = |(M_p)_t|$

Passo 15: *Cálculo do número de avaliações:* $N_{eval} = N_{exp}$. Acréscimo do número de iterações, $t \leftarrow t + 1$.

Fim do ciclo Iterativo.

Passo 16: *Fim.* Saída do conjunto solução de pontos. Frente de Pareto $\mathcal{F}^* \leftarrow P_t^*$.

4.4. Problemas de Teste

Com a finalidade de testar o desempenho do algoritmo MO-CORS tanto em problemas mono-objetivo como em multi-objetivo, foram selecionadas quatro (04) funções de teste para algoritmos de otimização mono-objetivo e onze (11) problemas de otimização multi-objetivo. A seguir, são apresentados os critérios de seleção de cada uma das funções e problemas de teste com a metodologia para elaboração dos experimentos computacionais em cada caso.

4.4.1. Experimentos computacionais mono-objetivo.

Para os testes de desempenho do MO-CORS em problemas mono-objetivo foram selecionadas as funções: Branin, Hartmann 3 e Hartmann 6 de Dixon e Szegö (1978), além da Goldstein Price, todas elas utilizadas por Silva (2011) no teste de seu algoritmo OGM. A Tabela 4.3 mostra as dimensões de cada uma das funções de teste usadas e os valores ótimos (mínimos) das mesmas.

Tabela 4.3: Funções de teste mono-objetivo. Fonte: Silva *et al.* 2011

Função de teste	Dimension	Limites laterais	Ótimo da Função
Branin	2	$[-5,10] \times [0,15]$	0,398
Goldstein-Price	2	$[-2,2]^2$	3
Hartman3	3	$[0,1]^3$	-3,86
Hartman6	6	$[0,1]^6$	-3,32

Para cada função da Tabela 4.3, foram realizados experimentos computacionais do MO-CORS com duas RBF diferentes: a inversa multiquádrica (IMq), com o valor do parâmetro de forma c fixado em 0,5 (SILVA, 2011), e a Thin Plate Spline com polinômio de extensão linear (TPS-pol). Foram também testados dois algoritmos diferentes para otimização dos metamodelos do MO-CORS: o algoritmo mono-objetivo *CRSI-Controlled Random Search with quadratic Interpolation* (MANZANARES-FILHO *et al.* 2018), também usado por Silva (2011), e o algoritmo NSGA II, base deste trabalho. Os resultados principais dos experimentos estão apresentados nas Tabelas 4.4 e 4.5. Um asterisco (*) indica o uso da IMq e dois asteriscos (**) indicam a TPS-pol. Os resultados obtidos por Silva (2011) com o seu OGM e com o uso direto do otimizador CRSI também estão mostrados. Aparecem ainda os resultados do CORS original (REGIS e SHOEMAKER, 2005), que se baseia no otimizador determinístico DiRect e no uso de TPS-pol. O critério de parada (convergência) da diferença de 1% foi aplicado em todos os testes de MO-CORS, OGM e CRSI. Com exceção do CORS original, todas as entradas das Tabelas 4.4 e 4.5 indicam médias do número de avaliações da função de teste (“custosa”)

necessárias para convergência entre vinte (20) repetições do experimento ou seja, 20 rodadas do algoritmo usando uma semente de números aleatórios diferente em cada rodada.

Tabela 4.4: Número médio de avaliações necessárias para OGM, MO-CORS com CRSI.

Função de teste	OGM*	OGM**	CORS	MO-CORS*	MO-CORS**	CRSI
Branin	19,8	51,7	34	20,1	51,4	138
Goldstein-Price	46,5	59,7	49	40,9	62,6	257
Hartman3	36,3	46,7	25	33,6	58,9	113
Hartman6	102,0	82,6	104	96,2	147,1	427

MO-CORS* e OGM* Resultado para RBF inversa multiquádrica ($c=0,5$). Otimizador CRSI.

MO-CORS** e OGM** Resultado para RBF *Thin Plate Spline – TPS* com polinômio linear. Otimizador CRSI.

A Tabela 4.4 reporta os resultados referentes ao uso do CRSI como otimizador do MO-CORS. A similitude entre os resultados obtidos para os algoritmos MO-CORS e OGM, no caso do uso da RBF inversa multiquádrica, mostra que ambos os algoritmos têm um desempenho similar nos problemas mono-objetivo, se observando uma diferença máxima na média das avaliações na função Hartman6 de 8,2 equivalente a 8,04% do máximo obtido por OGM. Igualmente esses resultados mostram que MO-CORS tem um bom desempenho nos problemas de otimização mono-objetivo, já que para todas as funções testadas são obtidas acelerações significativamente altas com respeito ao uso direto de CRSI, na ordem de: 6,87 vezes para a Branin, 6,28 para a Goldstein Price, 3,36 para a Hartmann 3 e 4,42 para a Hartmann 6, A aceleração aqui é definida pela razão entre o número de avaliações demandadas pelo uso direto do otimizador (CRSI) e o número de avaliações demandadas pelo MO-CORS. Outro aspecto importante tido em conta previamente à elaboração do algoritmo, foi a necessidade de introduzir um otimizador multi-objetivo, com capacidade de operar eficientemente também com problemas mono-objetivo. Como já destacado na seção 4.2.9, o algoritmo MO-CORS sempre utiliza o NSGA II para otimização dos metamodelos, em particular no problema auxiliar 1, que é o único a ser atacado no caso mono-objetivo. Sendo assim, os mesmos experimentos computacionais que originaram a Tabela 4.4 foram realizados, mas substituindo o CRSI pelo NSGA II como otimizador de metamodelos. Também foram obtidos os resultados da aplicação direta de NSGA II. Estes resultados estão reportados na Tabela 4.5.

Tabela 4.5: Número médio de avaliações necessárias para OGM, MO-CORS com NSGA II.

Função de teste	OGM* (Tab.4.4)	OGM** (Tab.4.4)	CORS (Tab.4.4)	MO-CORS*	MO-CORS**	NSGA II
Branin	19,8	51,7	34	17,2	58,7	87
Goldstein-Price	46,5	59,7	49	37	97,8	116
Hartman3	36,3	46,7	25	29,3	47,6	77
Hartman6	102,0	82,6	104	100,65	135,3	409

MO-CORS* Resultado para RBF inversa multiquádrica ($c=0,5$). Otimizador NSGA II.

MO-CORS** Resultado para RBF *Thin Plate Spline – TPS* com polinômio linear. Otimizador NSGA II.

Os resultados da Tabela 4.5 mostram que o uso do NSGA II para otimização dos metamodelos construídos com a RBF IMq, gera melhoras significativas no desempenho com respeito a OGM nas funções Branin, Goldstein Price, Hartmann 3 e 6. As acelerações de MO-CORS com esta RBF em relação ao NSGA II foram de 5.1, 3.14, 2.63 e 4.06, respectivamente. No caso da função Hartmann 6, praticamente não houve aceleração (1.01). Por outro lado, no caso da TPS-pol, o OGM apresentou sempre melhor desempenho que o MO-CORS, ressaltando o caso da função Hartmann 6, em que o OGM foi 1.64 vezes mais rápido que o MO-CORS.

Otimização do parâmetro de forma (c).

Observe-se que nos resultados mostrados nas Tabelas 4.4 e 4.5, tanto com o otimizador CRSI quanto com o NSGA II, a RBF IMq com parâmetro de forma $c = 1/2$, sempre resultou em melhor desempenho que a TPS-pol. No entanto, que o uso de outros valores de c pode resultar em desempenhos bem piores. Além disso, um valor adequado para uma função pode não ser para outra. Sendo assim, a utilização de RBF dependentes do parâmetro de forma traz ao projetista a desvantagem do desconhecimento *a priori* de um valor adequado de c em cada problema específico. Neste sentido, Silva *et al.*, (2019), (artigo em submissão) elaboraram um procedimento para obtenção automática de c , empregando o algoritmo eficiente proposto por Rippa (1999) para implementar a técnica de validação cruzada tipo LOOCV (em inglês: *Leave One Out Cross Validation*). Busca-se por meio dessa técnica determinar o valor de c que minimiza uma função de erro estimado. MO-CORS inclui esse procedimento baseado em LOOCV para otimizar o valor do parâmetro de forma da RBF IMq em cada iteração. Foram realizados 2 testes com aplicação de LOOCV no processo de otimização iterativa do metamodelo dentro de MOCORS, usando os algoritmos CRSI e NSGA II.

Tabela 4.6: Número médio de avaliações comparativas com MO-CORS-LOOCV.

Função de teste	MO-CORS CRSI (MO-CRSI)	MO-CORS NSGA II (MO-NSGA)	MO-CORS LOOCV CRSI (LO-CRSI)	MO-CORS LOOCV NSGA II (LO-NSGA)
Branin	20,1	17,2	20,9	15,8
Goldstein-Price	40,9	37	57,3	51,6
Hartman3	33,6	29,3	35,5	25,5
Hartman6	96,2	100,7	124,3	103,8

MO-CORS Resultado para RBF inversa multiquádrica ($c=0,5$)

MO-CORS-LOOCV Resultado para RBF inversa multiquádrica.

A Tabela 4.6 mostra que o desempenho de MO-CORS com o otimizador CRSI se mantém praticamente igual para as funções Branin e Hartman3, enquanto que para a Goldstein-Price e Hartman6 o uso de LOOCV piora o desempenho de MO-CORS com uma diferença percentual ($\Delta D\%$), de mais de um 20% segundo a Equação 4.20. O valor de c obtido por LOOCV, não é único durante o processo, e incluso pode não convergir a um único valor ao final do processo.

$$\Delta D\% = \frac{|NE_{MOCORS} - NE_{MO-LOOCV}|}{\max(NE_{MOCORS}, NE_{MO-LOOCV})} \times 100\% \quad (4.20)$$

Com a utilização do algoritmo NSGA II, os resultados mostram que para as funções: Branin e Hartman3, o desempenho de MO-CORS apresenta melhoras consideráveis com o emprego de LOOCV, para otimizar o parâmetro de forma, nestes casos obteve-se uma melhoria de 8,14% e 12,97% respectivamente. Por outro lado para as funções Goldstein-Price e Hartman6, o desempenho piora em 28,43% e 2,99% respectivamente. É importante considerar que, para as funções de teste usadas neste trabalho, um valor de c em torno de 0,5 foi avaliado como adequado no trabalho de Silva (2011), mediante um grande número de testes para as RBF multiquádrica e inversa multiquádrica em cada função de teste. Para outras funções, os valores adequados de c resultaram bem diferentes. Além disso, o valor de c otimizado por LOOCV, não é único durante o processo, e incluso pode não convergir a um único valor ao final do mesmo. Outro aspecto importante que pode marcar diferenças nos resultados, é que o algoritmo eficiente para aplicação de LOOCV no cálculo do parâmetro de forma, foi desenvolvido e testado por Rippa (1999) para a construção de metamodelos estanques partindo de um grande número de pontos; aparentemente, ele não havia ainda sido testado no escopo da construção iterativa de metamodelos durante um processo de otimização, como aquele do MO-CORS. Os resultados obtidos depois de 20 repetições de cada experimento são ilustrados graficamente nos diagramas de caixa e bigode, ou de extremos e quartis, da Figura 4.3, afim de fornecer uma representação esclarecedora sobre a forma como os dados se distribuem, nomeadamente quanto a: maior ou menor concentração, simetria e a existência de valores atípicos.

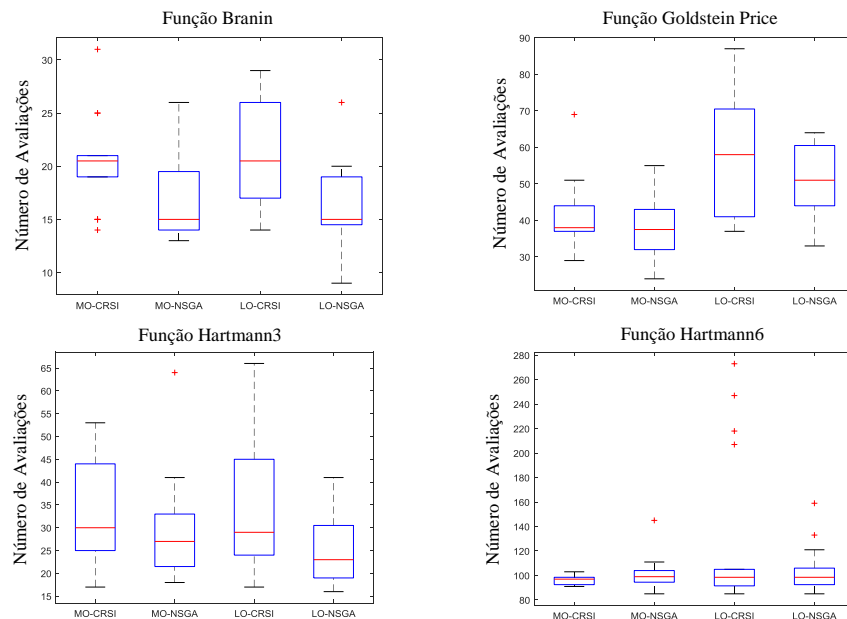


Figura 4.3: Diagramas para a análise da influência do otimizador e LOOCV em problemas mono-objetivo.

Os diagramas da Figura 4.3 mostram uma distribuição uniforme, com poucos valores atípicos, exceto para o caso do uso do MO-CORS sem LOOCV com algoritmo de otimização do problema auxiliar CRSI (MO-CRSI) na função Branin, e o MO-CORS com LOOCV com algoritmo de otimização auxiliar CRSI (LO-CRSI) na função Hartman6, em que ocorreram aproximadamente 20% de soluções atípicas. Nesses casos, as médias das avaliações da função custosa estiveram muito próximas (com e sem a aplicação de LOOCV), sendo 21 para a Branin e 98 para a Hartman6. Em geral, as médias para a aplicação de MO-CORS com e sem LOOCV resultaram aproximadamente iguais para cada otimizador do metamodelo, exceto no caso da função Goldstein-Price, em que o número avaliações foi de 58 com LOOCV e 38 sem LOOCV, com o otimizador CRSI, e de 52 com LOOCV e 37 sem LOOCV, com o otimizador NSGA II.

4.4.2. Experimentos computacionais multi-objetivo.

Em princípio, visando as aplicações multi-objetivo pretendidas com MO-CORS em problemas de aerodinâmica e máquinas de fluxo com até 12 variáveis e 3 funções objetivos, foi selecionado o conjunto de problemas de teste “MOP” (VAN VELDHUIZEN e LAMONT, 2000) e “ZDT” de (ZITZTLER *et al.*, 2001), definidos nas Tabelas 4.7 e 4.8 respectivamente.

Estes conjuntos de problemas de teste, além de ter a quantidade de variáveis e funções objetivo desejadas, capturam coletivamente uma ampla variedade de características de teste desejáveis, tais como aquelas mencionadas em Huband *et al.* (2006), dentre outras: possuem ótimos que não são os extremos do domínio, apresentam multiplicidade na ordem de magnitude dos parâmetros do espaço de decisão e do espaço objetivo; as soluções ótimas de Pareto são conhecidas; apresentam diversidade nas geometrias das frentes de Pareto e suas contra-imagens; e podem ser unimodais ou multimodais.

Foi assim realizado um total de 9 experimentos computacionais aplicando o algoritmo MO-CORS e 9 experimentos aplicando diretamente o algoritmo NSGA II. Em cada experimento, os algoritmos foram aplicados 20 vezes ininterruptamente, mantendo o mesmo estado randômico inicial para cada problema de teste. Isso no intuito de diminuir o efeito da aleatoriedade dos algoritmos durante a análise de resultados. Para a aplicação do algoritmo MO-CORS em cada experimento, foi utilizado o algoritmo NSGA II como motor de busca para a resolução dos problemas de otimização auxiliares 1 e 2, com uma população de 80 indivíduos, 150 gerações e uma fração de Pareto igual a 1. A intenção aqui foi permitir que o NSGA II pudesse atingir uma convergência na otimização dos metamodelos com a obtenção de até 80 pontos candidatos nas frentes finais de cada iteração. Os metamodelos de cada função objetivo foram construídos usando a RBF cúbica (r^3) com polinômio de extensão linear, mesma utilizada nas referências principais (AKHTAR e SHOEMAKER, 2016), (MÜLLER, 2017). Todos

os bancos de dados iniciais foram construídos usando as técnicas HCL ou PFC segundo o número de variáveis de cada problema. Ou seja, para um número de variáveis menor ou igual à 3 ($n_{var} \leq 3$), a técnica utilizada foi o PFC, com um número de níveis igual à 2 ($N_{niv} = 2$); para um número de variáveis maior que 3 ($n_{var} > 3$) a técnica usada foi o HCL com número total de pontos de $N_{exp} = 3n_{var} + 2$.

Tabela 4.7: Funções de teste de Van Veldhuizen (MOP's). Fonte: Huband *et al.* (2006).

Nome	Problema	Limites laterais
MOP1 <i>min-min</i>	$f_1(x) = x^2$ $f_2(x) = (x - 2)^2$	$[-10^5, 10^5]$
MOP2 <i>min-min</i>	$f_1(x_1, \dots, x_n) = 1 - \exp\left(-\sum_{i=1}^n (x_i - 1/\sqrt{n})^2\right)$ $f_2(x_1, \dots, x_n) = 1 - \exp\left(-\sum_{i=1}^n (x_i + 1/\sqrt{n})^2\right)$	$[-4, 4]$
MOP3 <i>max-max</i>	$f_1(x_1, x_2) = -1 - (A_1 - B_1)^2 - (A_2 - B_2)^2$ $f_2(x_1, x_2) = -(x_1 - 3)^2 - (x_2 + 1)^2$ $A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2$ $A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2$ $B_1 = 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \cos x_2$ $B_2 = 1.5 \sin x_1 - \cos x_1 + 2 \sin x_2 - 0.5 \cos x_2$	$[-\pi, \pi]$
MOP4 <i>min-min</i>	$f_1(x_1, x_2, x_3) = \sum_{i=1}^2 -10 \exp^{-0.2 \sqrt{x_i^2 + x_{i+1}^2}}$ $f_2(x_1, x_2, x_3) = \sum_{i=1}^3 x_i ^{0.8} + 5 \sin(x_i^3)$	$[-5, 5]$
MOP5 <i>min-min-min</i>	$f_1(x_1, x_2) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2)$ $f_2(x_1, x_2) = \frac{(3x_1 - 2x_2 + 4)^2}{8} - \frac{(x_1 - x_2 + 1)^2}{27} + 15$ $f_3(x_1, x_2) = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1 \exp(-x_1^2 - x_2^2)$	$[-30, 30]$
MOP6 <i>min-min</i>	$f_1(x_1) = x_1$ $f_2(x_1, x_2) = (1 + 10x_2) \left(1 - \left(\frac{x_1}{1 + 10x_2}\right)^2 - \frac{x_1}{1 + 10x_2} \sin(8\pi x_1)\right)$	$[0, 1]$
MOP7 <i>min-min-min</i>	$f_1(x_1, x_2) = \frac{(x_1 - 2)^2}{2} + \frac{(x_2 + 1)^2}{13} + 3$ $f_2(x_1, x_2) = \frac{(x_1 + x_2 - 3)^2}{36} + \frac{(-x_1 + x_2 + 2)^2}{8} - 17$ $f_3(x_1, x_2) = \frac{(x_1 + 2x_2 - 1)^2}{175} + \frac{(-x_1 + 2x_2)^2}{17} - 13$	$[-400, 400]$

Tabela 4.8: Funções de teste de Zitzler (ZDT's). Fonte: Huband *et al.* (2006).

Nome	Problema	Limites laterais
ZDT1 <i>min-min</i>	$f_1 = x_1$ $g(x) = 1 + 9 \sum_{i=2}^n x_i / (n - 1)$ $h = 1 - \sqrt{f_1/g}$	$[0, 1]$
ZDT2 <i>min-min</i>	Como ZDT1 exceto $h = 1 - (f_1/g)^2$	$[0, 1]$

Os problemas ZDT1 e ZDT2 são min-min e: $f_2(x, g) = g(x)h(f_1, g(x))$

A fim de estabelecer uma comparação justa dos algoritmos MO-CORS e NSGA II aplicado diretamente, em cada ciclo de aplicação, foi seguida uma *metodologia de comparação equilibrada*, a saber:

- Aplica-se primeiramente MO-CORS até que seja atingido o critério de parada da diferença da distância geracional, $\varepsilon_d \leq 10^{-4}$.
- Aplica-se em seguida NSGA II iterativamente, aumentando o número de gerações uma por vez em cada iteração, até que duas condições sejam cumpridas: (i) o número de pontos obtidos na frente atinja ao menos o número de pontos na frente de MO-CORS; (ii) a métrica de espalhamento de NSGA II atinja valor igual ou menor que a métrica de espalhamento de MO-CORS.

Resultados para os problemas de teste multi-objetivo.

A seguir são apresentadas para cada problema de teste as seguintes ilustrações:

- Seis (06) diagramas de caixa e bigode, para cada problema de teste, com os resultados comparativos entre os algoritmos MO-CORS e NSGA II para as métricas de *espalhamento* (Δ), *espaçamento* (SM), *taxa de sucesso* (HRM) e a de contribuição de soluções na *frente das frentes de Pareto* ($MPSFF$ ou BPV), Dois dos diagramas apresentam os resultados referentes ao número de avaliações da função custosa e a aceleração (Figuras 4.4, 4.6, 4.8, 4.10, 4.12, 4.14, 4.16, 4.18 e 4.20).
- Um ou dois (02) gráficos ilustrando o espaço de decisão (só para as MOP) e o espaço objetivo de um dos resultados representativos das 20 rodadas realizadas para a obtenção dos diagramas anteriores. Isto no intuito de mostrar sempre que for possível, a aproximação comparativa dos espaços de decisão e as frentes de Pareto (Figuras 4.5, 4.7, 4.9, 4.11, 4.13, 4.15, 4.17, 4.19 e 4.21).

Problema de teste MOP1:

MOP1 é um problema de teste de minimização de ambas funções objetivo (min-min), considerado aqui de baixa complexidade quanto ao número de variáveis de decisão (uma) e de funções objetivo (duas) (Tabela 4.7). Apresenta uma frente de Pareto contínua e convexa (COELLO *et al.*, 2007) e assim se verifica na Figura 4.5.

O foco do teste deste problema foi verificar a capacidade de convergência do algoritmo MO-CORS em espaços de decisão e objetivo com uma grande amplitude de busca, como se verifica nos limites laterais mostrados na Tabela 4.7

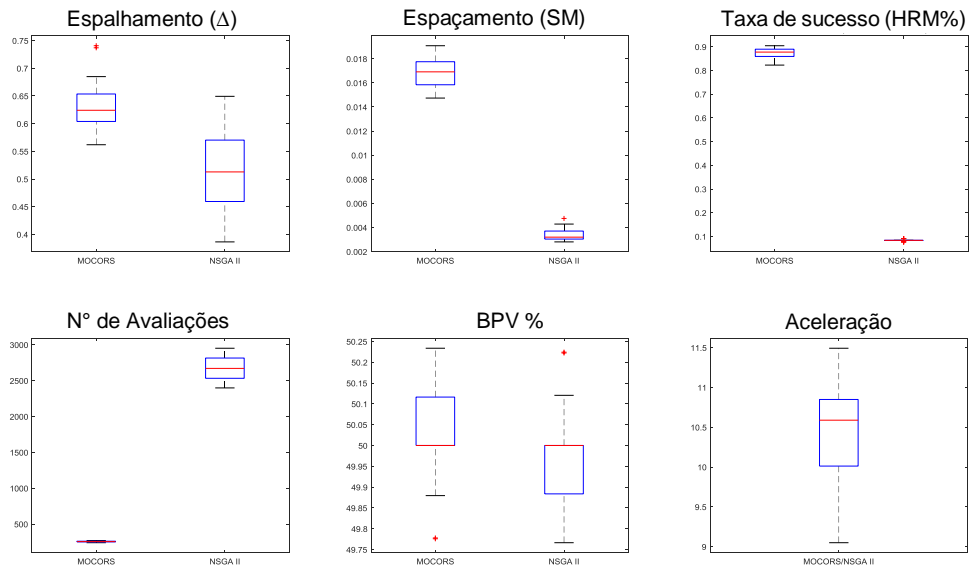


Figura 4.4: Métricas e aceleração para o problema MOP1

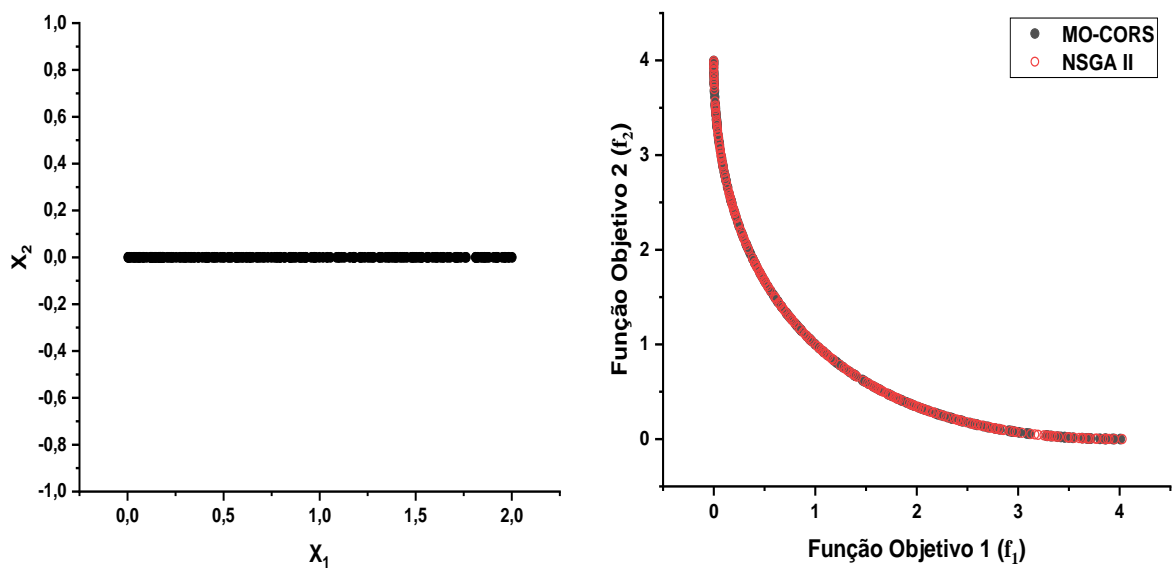


Figura 4.5: Espaço de decisão e frente de Pareto MOP1.

Problema de teste MOP2:

MOP2 é um problema de teste de minimização de duas (02) funções objetivo (min-min), considerado aqui de complexidade variável quanto ao número de variáveis de decisão, já que é possível aplicá-lo até trinta (30) variáveis (Tabela 4.7). Apresenta uma frente de Pareto contínua e não convexa (COELLO *et al.*, 2007), assim verifica-se na Figura 4.7. O foco do teste deste problema foi verificar a capacidade do algoritmo MO-CORS na obtenção de frentes não convexas com diferentes números de variáveis de decisão. Embora este problema tenha sido testado com três, seis e doze variáveis, os resultados apresentados correspondem a $n_{var} = 12$.

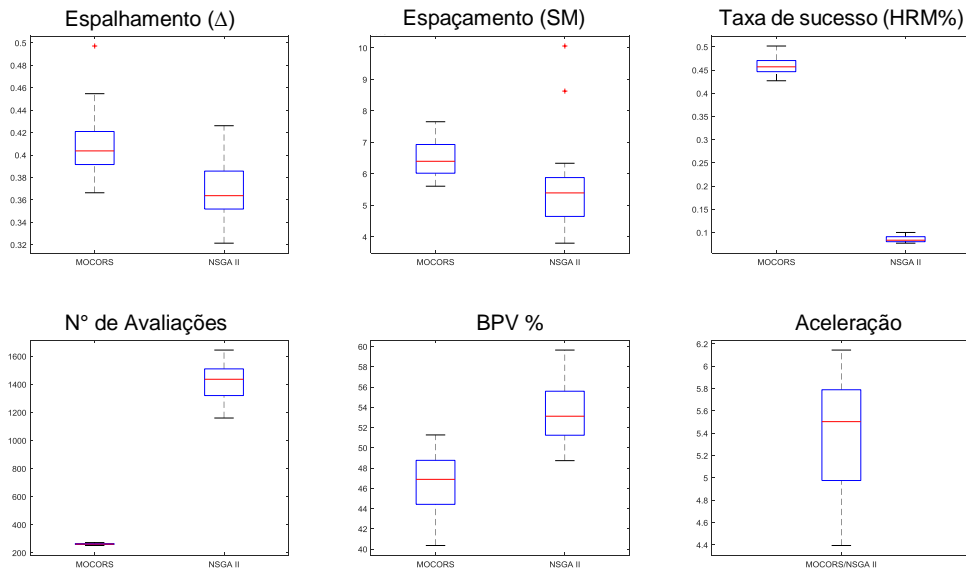


Figura 4.6: Métricas e aceleração para o problema MOP2 (12 Variáveis).

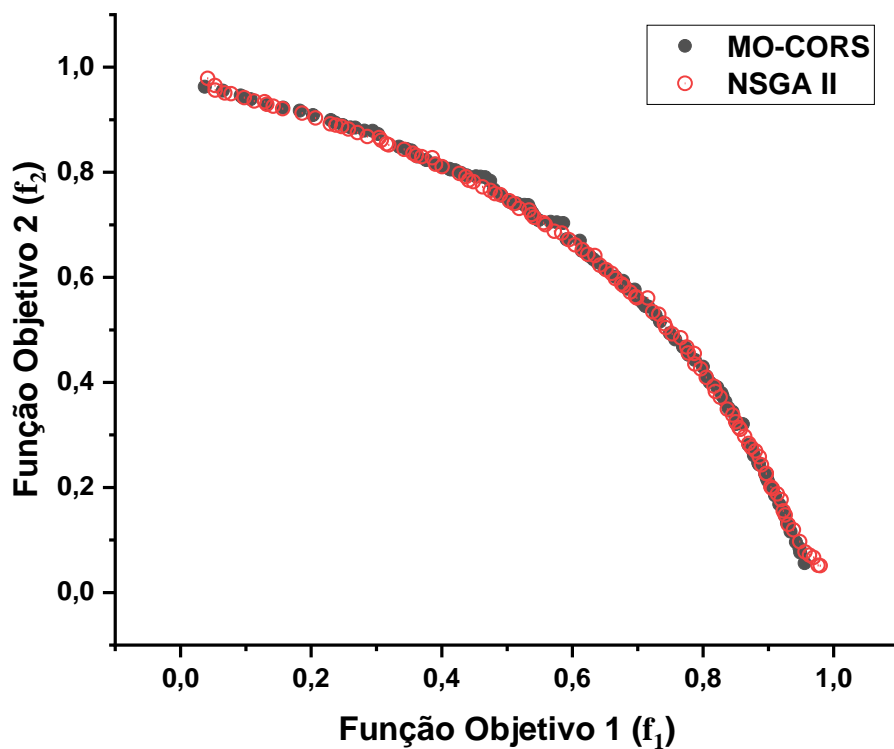


Figura 4.7: Frente de Pareto MOP2 (12 Variáveis).

Problema de teste MOP3:

MOP3 é um problema de teste de maximização de ambas funções objetivo (max-max), considerado aqui de baixa complexidade quanto ao número de variáveis de decisão (duas) e de funções objetivo (duas) (Tabela 4.7). Apresenta uma frente de Pareto descontínua e convexa (COELLO *et al.*, 2007), assim verifica-se na Figura 4.9.

O foco do teste deste problema foi verificar a capacidade do algoritmo MO-CORS na obtenção de frentes descontínuas convexas de maximização das funções objetivo.

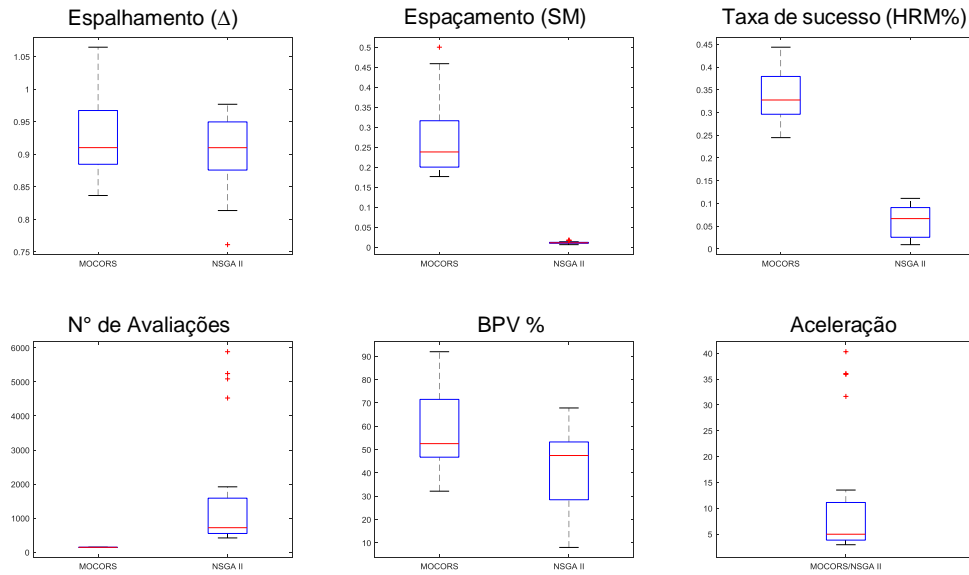


Figura 4.8: Médias das métricas e aceleração para o problema MOP3

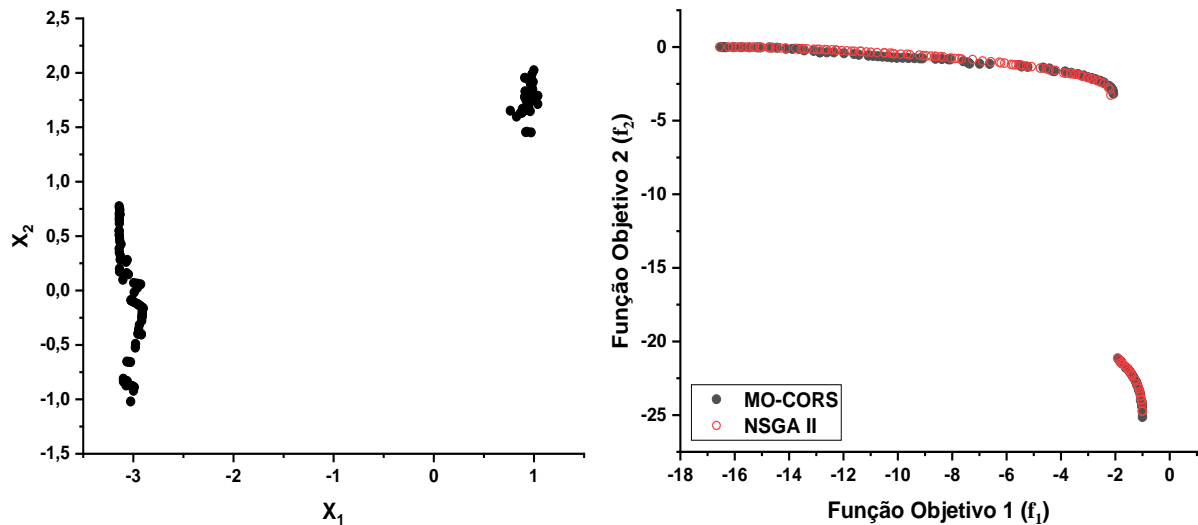


Figura 4.9: Espaço de decisão e frente de Pareto MOP3.

Problema de teste MOP4:

MOP4 é um problema de teste de minimização de ambas funções objetivo (min-min), considerado aqui de complexidade média na exploração do espaço de decisão (três (03) variáveis) e duas funções objetivo (Tabela 4.7). Apresenta uma frente de Pareto descontínua e não convexa (COELLO *et al.*, 2007), assim verifica-se na Figura 4.11. A complexidade da MOP4, é devida, além da exploração do espaço de decisão, visto que as contra-imagens da frente de Pareto estão aglomerados em 4 nichos.

O foco do teste deste problema foi avaliar o desempenho do algoritmo MO-CORS na exploração de frentes descontínuas não convexas.

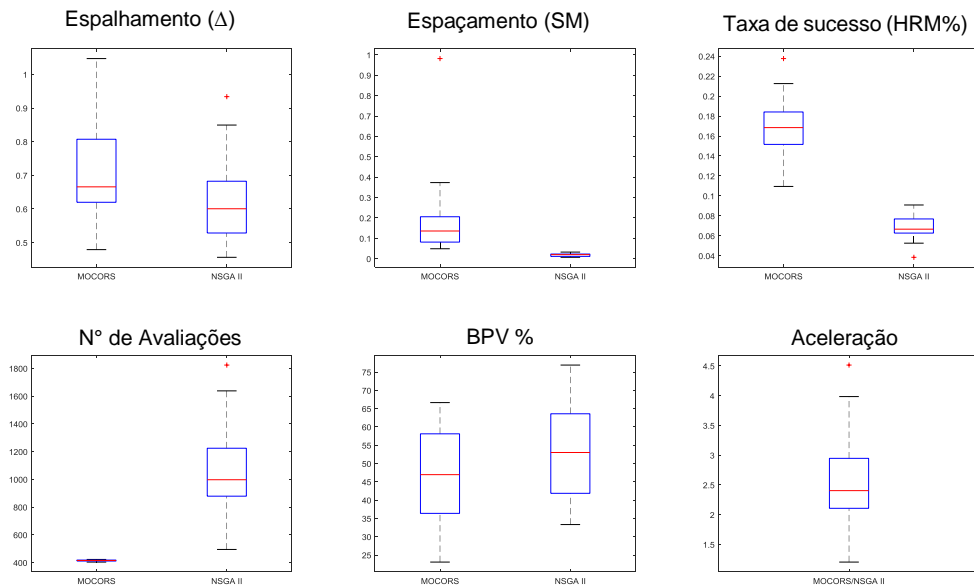


Figura 4.10: Métricas e aceleração para o problema MOP4

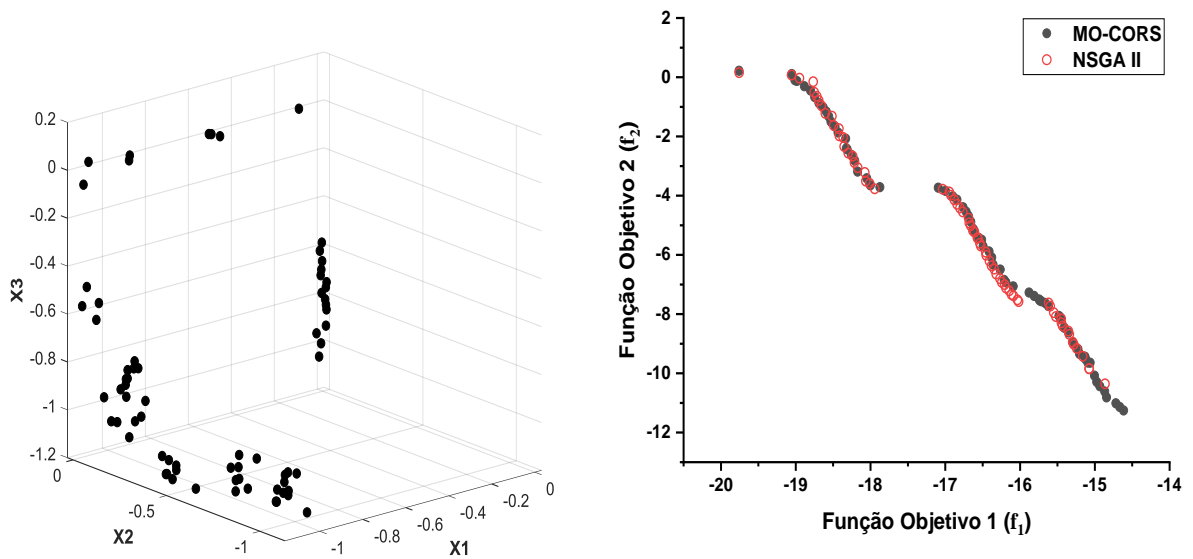


Figura 4.11: Espaço de decisão e frente de Pareto MOP4.

Problema de teste MOP5:

MOP5 é um problema de teste de minimização de três (03) funções objetivo (min-min), considerado aqui de complexidade baixa na exploração do espaço de decisão (dois (02) variáveis) (Tabela 4.7). Apresenta uma frente de Pareto contínua tridimensional (COELLO *et al.*, 2007), assim verifica-se na Figura 4.13.

O foco do teste deste problema foi avaliar o desempenho do algoritmo MO-CORS na determinação de espaços objetivos tridimensionais com pontos alinhados numa curva contínua.

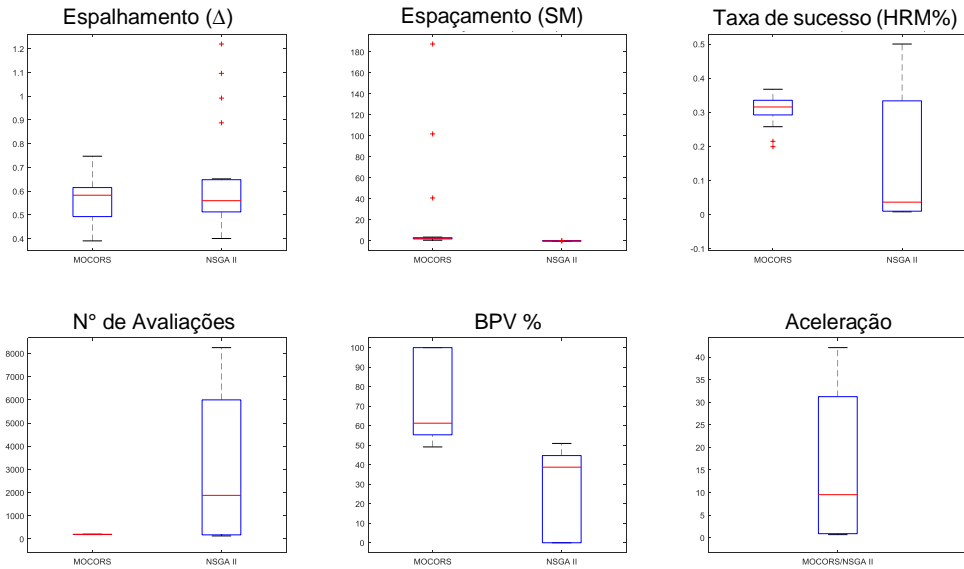


Figura 4.12: Métricas e aceleração para o problema MOP5

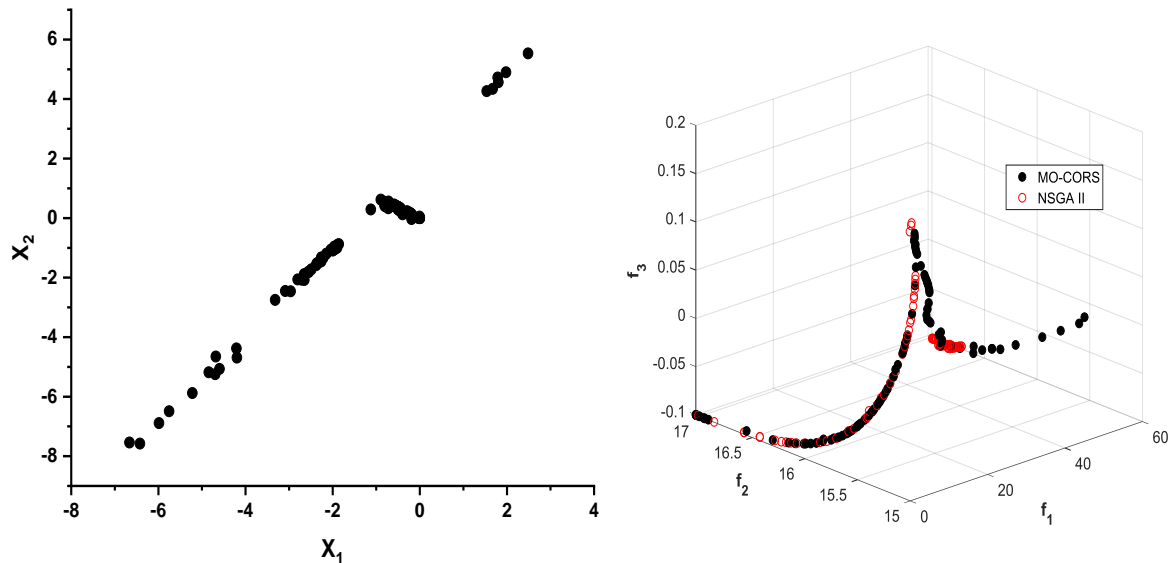


Figura 4.13: Espaço de decisão e frente de Pareto MOP5.

Problema de teste MOP6:

MOP6 é um problema de teste de minimização de ambas funções objetivo (min-min), considerado aqui de complexidade baixa na exploração do espaço de decisão (dois (02) variáveis) e duas funções objetivo (Tabela 4.7). Apresenta uma frente de Pareto descontínua e convexa de 4 nichos (COELLO *et al.*, 2007), assim verifica-se na Figura 4.15. A contra imagem de cada seção da frente é também descontínua. O foco do teste deste problema foi verificar a

capacidade do algoritmo MO-CORS na obtenção de frentes descontínuas não convexas de 4 nichos.

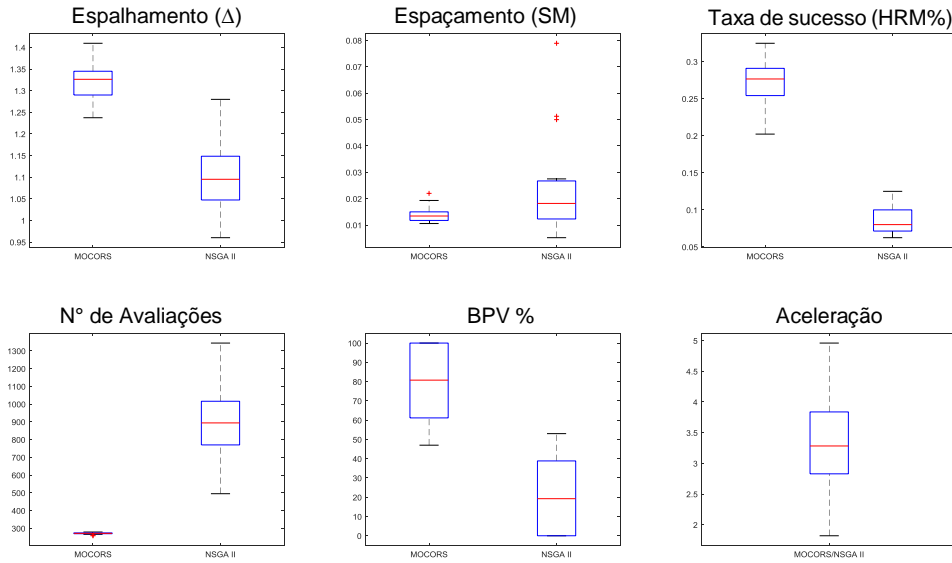


Figura 4.14: Métricas e aceleração para o problema MOP6

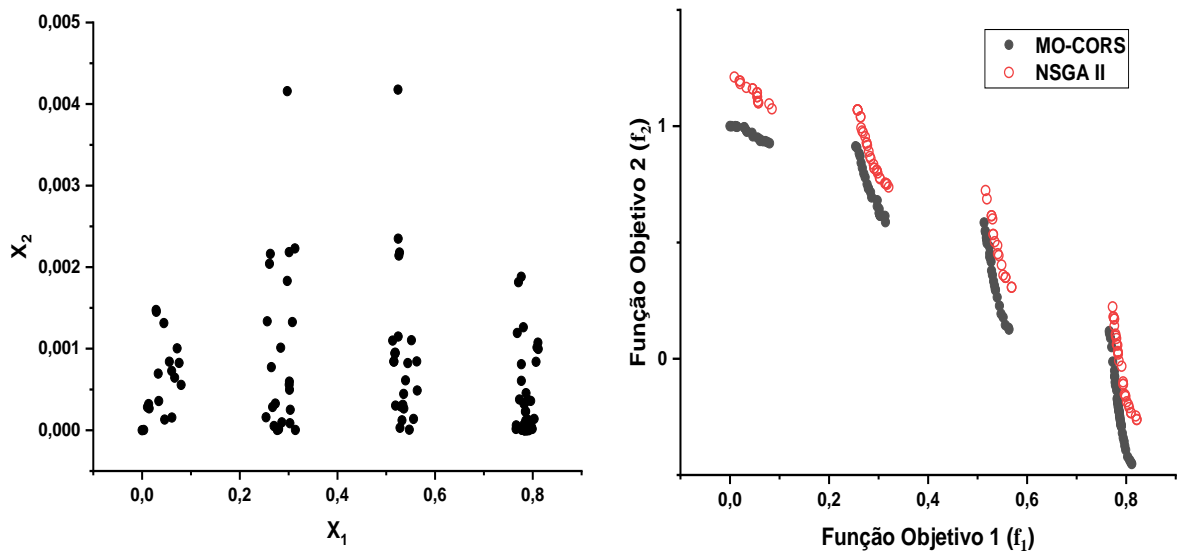


Figura 4.15: Espaço de decisão e frentes de Pareto MOP6.

Problema de teste MOP7:

MOP7 é um problema de teste de minimização de três (03) funções objetivo (min-min-min), considerado aqui de complexidade baixa na exploração do espaço de decisão (dois (02) variáveis) (Tabela 4.7). Apresenta uma frente de Pareto contínua variada (COELLO *et al.*, 2007), assim verifica-se na Figura 4.17. O foco do teste deste problema foi avaliar o desempenho do algoritmo MO-CORS em espaços objetivos tridimensionais com frente de Pareto em formato de superfície e linhas contínuas.

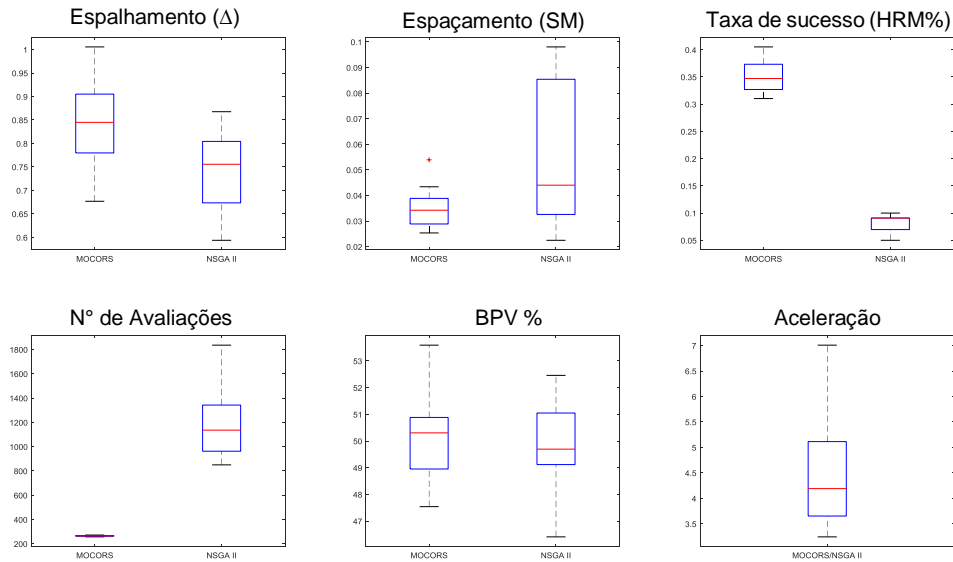


Figura 4.16: Métricas e aceleração para o problema MOP7

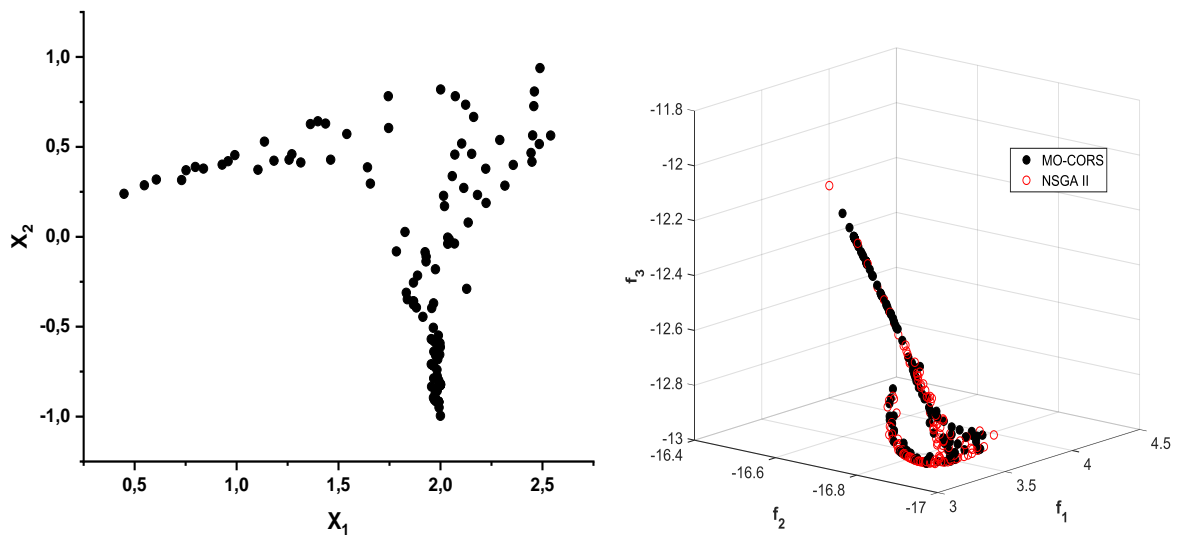


Figura 4.17: Espaço de decisão e frentes de Pareto MOP7.

Problema de teste ZDT1

ZDT1 é um problema de teste de minimização de ambas funções objetivo (min-min), considerado aqui de alta complexidade quanto ao número de variáveis de decisão (Tabela 4.8). Apresenta uma frente de Pareto contínua e convexa (COELLO *et al.*, 2007), assim verifica-se na Figura 4.19. O foco do teste deste problema foi, da mesma forma que na MOP2, verificar a capacidade do algoritmo MO-CORS na exploração de espaços de decisão com diferentes números de variáveis de decisão, mas em frentes convexas. Embora este problema tenha sido testado com três, seis e doze variáveis, os resultados apresentados correspondem a $n_{var} = 12$.

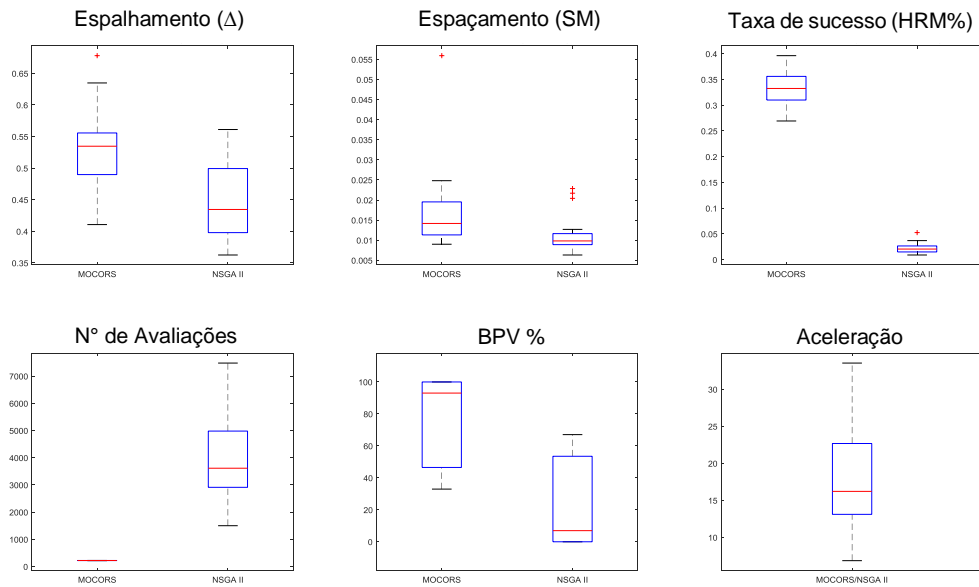


Figura 4.18: Métricas e aceleração para o problema ZDT1 (12 variáveis).

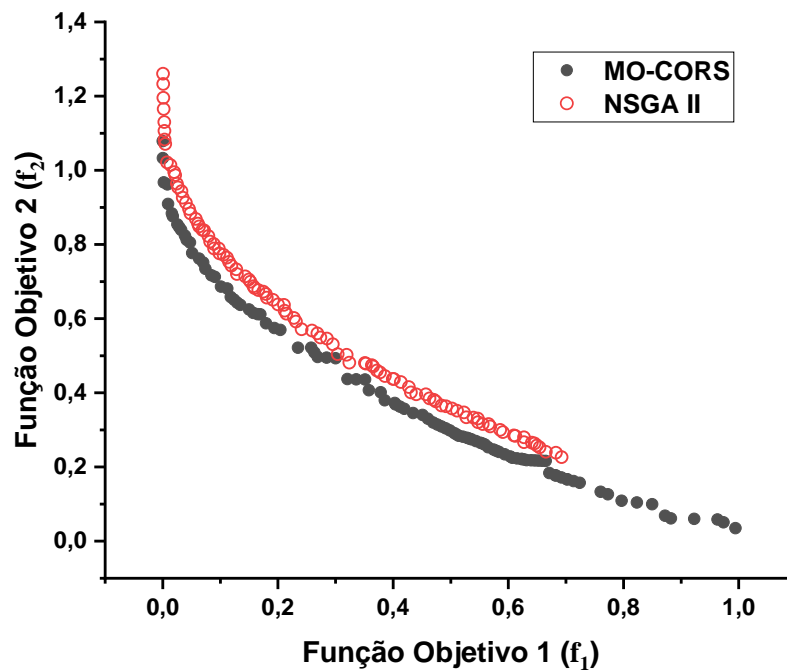


Figura 4.19: Frentes de Pareto do problema ZDT1 (12 variáveis).

Problema de teste ZDT2 (6 Variáveis)

ZDT2 é um problema de teste de minimização de dois (02) funções objetivo (min-min), considerado aqui de complexidade variável quanto ao número de variáveis de decisão, já que é possível aplicá-lo até trinta (30) variáveis (Tabela 4.8). Apresenta uma frente de Pareto contínua e não convexa (COELLO *et al.*, 2007), assim verifica-se na Figura 4.21. O foco do teste deste problema foi verificar a capacidade do algoritmo MO-CORS na exploração de espaços objetivos não convexos ($n_{var} = 6$).

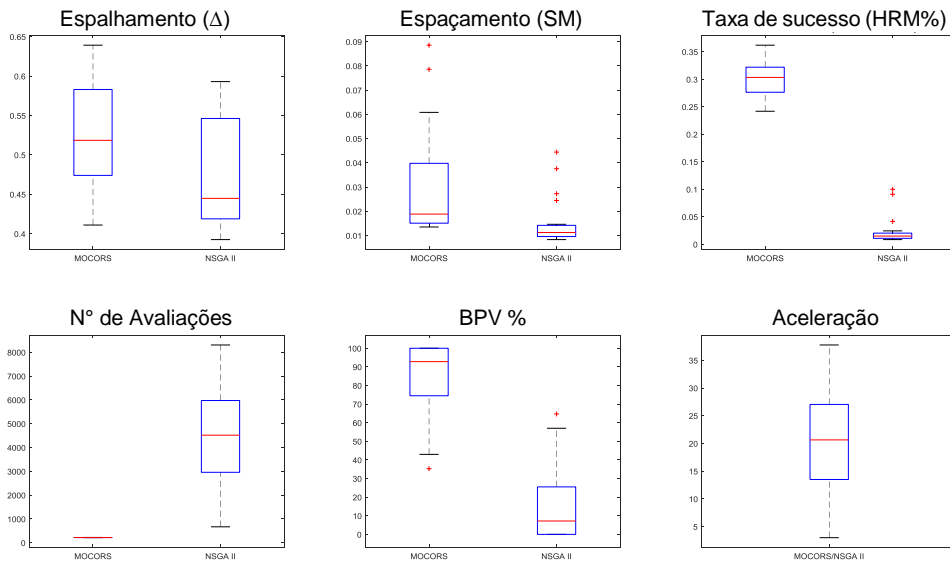


Figura 4.20: Métricas e aceleração para o problema ZDT2 (6 variáveis).

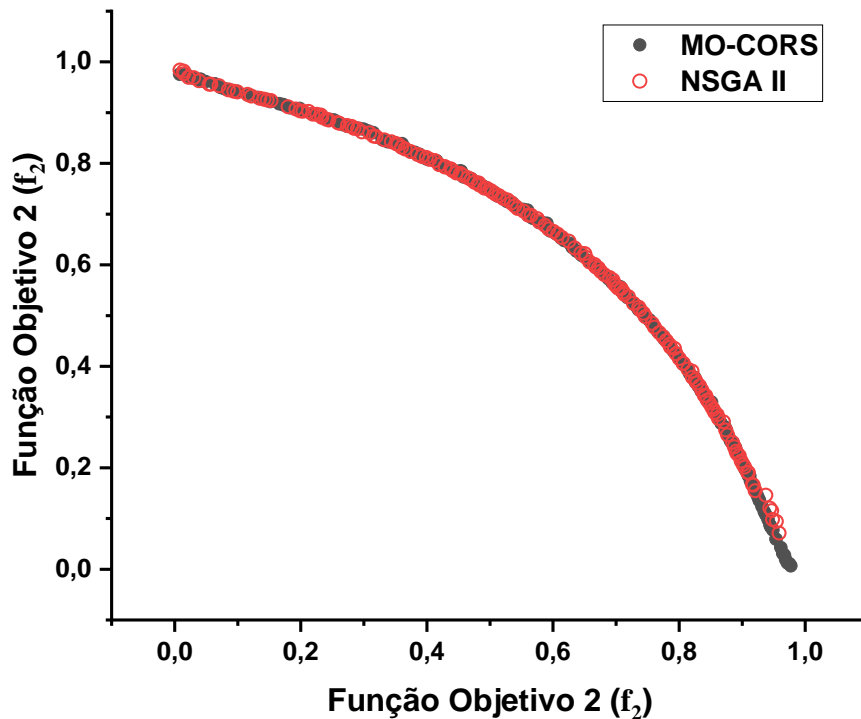


Figura 4.21: Frentes de Pareto do problema ZDT2 (6 variáveis).

4.4.3. Análise de resultados dos problemas de teste multi-objetivo.

Na Tabela 4.9, apresenta-se os valores meios das métricas de desempenho, do número de avaliações de cada algoritmo de otimização e da aceleração proporcionada pelo algoritmo MO-CORS em cada experimento computacional.

Tabela 4.9: Valores das métricas de desempenho e aceleração.

MOOP Teste	NSGA II (direto)					MO-CORS					
	Δ	SM	HRM	N_{eval}	BPV%	Δ	SM	HRM	N_{eval}	BPV%	Acel.
MOP1	0,531	0,001	0,081	2600	50	0,625	0,017	0,875	245,3	50	10,6
MOP2	0,362	0,005	0,062	1480	53	0,402	0,006	0,451	269,1	47	5,5
MOP3	0,913	0,011	0,052	780	52	0,915	0,231	0,325	156,0	48	5,0
MOP4	0,588	0,012	0,064	1000	53	0,652	0,121	0,167	416,7	47	2,5
MOP5	0,575	0,061	0,022	2000	40	0,583	0,312	0,312	219,8	60	9,1
MOP6	1,090	0,018	0,075	890	20	1,325	0,014	0,275	269,7	80	3,3
MOP7	0,759	0,043	0,043	1180	52	0,848	0,034	0,344	281,0	48	4,2
ZDT1	0,441	0,010	0,024	3500	8	0,538	0,014	0,325	278,8	92	16
ZDT2	0,442	0,012	0,021	4500	7	0,521	0,018	0,311	219,5	93	20,5

O primeiro aspecto a se ressaltar nos resultados em geral, é que os valores das métricas de espalhamento da frente do algoritmo NSGA II para todos os problemas, resultaram menores que às do algoritmo MO-CORS. Isso se explica pela *metodologia de comparação equilibrada* aplicada nos experimentos: uma rodada do algoritmo NSGA II só é interrompida quando a quantidade de pontos na frente ficar ao menos igual à obtida pelo algoritmo MO-CORS e com um valor de espalhamento igual ou menor. Contudo, os valores de espalhamento obtidos pelo MO-CORS (entre 0,402 e 1,325) podem ser considerados satisfatórios em relação aos do NSGA II (entre 0,362 e 1,09). Eles indicam que o algoritmo MO-CORS é capaz de obter frentes de Pareto com diversidade aceitável, embora demandando uma quantidade relativamente pequena de avaliações das funções “custosas”. Em relação à cobertura de pontos na frente, medida pela métrica de espaçamento (SM), embora os resultados da Tabela 4.9 mostrem que em geral, as frentes geradas por NSGA II produzam maior cobertura (menor SM) que o MO-CORS, os diagramas de caixa e bigode indicam também que as distribuições tendem a apresentar muitos valores atípicos pela aplicação de NSGA II. De mais relevante, o algoritmo MO-CORS alcança acelerações superiores as 2 vezes em todos os problemas, especificamente as acelerações estiveram entre 2,5 e 20,5, dependendo da complexidade do problema. Leve-se em conta também que, para a maioria dos problemas, o número de pontos aportados pelo MO-CORS na frente das frentes corresponde a mais do 50% do total. Mesmo para os problemas MOP2, MOP3, MOP4 e MOP7, a métrica BPV% do algoritmo MO-CORS resultou em 47%, 48%, 47% e 48%, respectivamente valores muito próximos de 50% – com as respectivas acelerações de 5,5, 5, 2,5 e 4,2. Em geral, pode-se dizer que o algoritmo MO-CORS converge com acelerações significativas para frentes próximas às obtidas pelo NSGA II.

É interessante fazer uma análise particular de cada grupo de problemas classificados de acordo com certas características comuns, tais como: o número de variáveis, número objetivos,

descontinuidade ou não descontinuidade da frente de Pareto e convexidade e não convexidade na frente de Pareto.

O problema MOP1, considerado o de menor complexidade entre todos, por ter única variável de decisão e uma frente convexa, busca testar o desempenho dos algoritmos em espaços de decisão extensos, verificando a importância da sua normalização nos espaços. Neste caso ambos algoritmos convergem na mesma frente com um aporte equitativo na frente das frentes de 50%. O MO-CORS, porém, apresenta uma aceleração de 10,6 vezes, com uma taxa de sucesso de 87,5% ($HRM \times 100\%$), contra apenas 0,1% para o NSGA II. A métrica de espaçamento indica que o NSGA II cobre melhor a frente de Pareto, mas pode-se considerar que a cobertura do MO-CORS é também aceitável (valor de SM abaixo de 0,02) .

Os problemas MOP2 e ZDT2 foram utilizados para testar o desempenho do algoritmo MO-CORS em problemas com um número considerável de variáveis, 12 e 6 respectivamente, e que apresentam frentes de Pareto não convexas. Os resultados das métricas de espaçamento indicam que o MO-CORS, atinge uma boa cobertura para os dois problemas, com valores muito próximos aos do NSGA II na ordem de 0,006 e 0,018 respectivamente. No caso da MOP2 os diagramas da Figura 4.6 indicam que as distribuições de todas as métricas de desempenho mantêm uniformidade e apresentam poucas amostras atípicas. O mesmo já não acontece no problema ZDT2: na Figura 4.20 pode-se observar que a métrica de espaçamento exibe uma boa quantidade de amostras atípicas, sobretudo para o algoritmo NSGA II. Para o problema MOP2, MO-CORS consegue acelerações em torno de 5 vezes em relação a NSGA II, com uma taxa de sucesso de 45,1%; para a ZDT2, as acelerações são em torno de 20 vezes, com uma taxa de sucesso de 31,1%.

Nos problemas MOP3, MOP4 e MOP6, de frentes descontínuas, o MO-CORS mostrou capacidade de convergir às frentes respectivas, como é observado nas Figuras 4.9, 4.11 e 4.15, com acelerações de 5, 2,5 e 3,3. Note-se que o problema MOP3 é do tipo *max-max* – um aspecto que o diferencia de todos outros. Verifica-se, segundo os resultados obtidos, um bom desempenho do MO-CORS no problema MOP3. No problema MOP4, com duas funções objetivo e três variáveis de decisão, foi atingida a mínima aceleração (2,5). Contudo, o diagrama da Figura 4.10 mostra resultados atípicos altos para o número de avaliações da função “custosa” no caso de NSGA II. Mesmo assim, a taxa de sucesso de MO-CORS foi de 16,7% contra 6,4% de NSGA II. Ressalta-se na Figura 4.11 que tanto a frente de MO-CORS como a de NSGA convergem à chamada frente real (COELLO *et al.* 2007). No caso do problema MOP6, observa-se na Tabela 4.7 que o algoritmo NSGA II não alcança a convergência, pois sua métrica BPV% fica em torno de 20%. Isto pode ser constatado nas frentes resultantes mostradas na Figura 4.15, onde os pontos NSGA II não alcançam a cobertura dos pontos pretos MO-CORS.

No problema ZDT1 de alta complexidade (com 12 variáveis e frente não convexa), observa-se na Tabela 4.9 que o algoritmo NSGA II não alcança a convergência assim como no problema MOP6, com sua métrica BPV% atingindo valores de 8%. Isto pode ser constatado nas frentes resultantes mostradas na Figura 4.19, onde os pontos NSGA II não alcançam a cobertura dos pontos de MO-CORS.

O desempenho do algoritmo MO-CORS nos problemas com espaço de objetivo tridimensional, MOP5 e MOP7, mostrou-se bem aceitável, obtendo-se boas representações das frentes de Pareto em cada caso, como se pode observar nas Figuras 4.13 e 4.17. As acelerações foram em torno de 9,1 e 4,2 com uma taxa de sucesso na ordem de 30% para MO-CORS (em ambos os casos) contra 2% e 4% para NSGA II.

4.4.4. Contribuição dos problemas auxiliares

Neste item, apresentam-se os resultados da percentagem na frente final de pontos oriundos da solução de cada problema auxiliar 1, 2 ou 3. Dos bancos de dados das frentes ilustradas nas Figuras 4.5, 4.7, 4.9, 4.11, 4.13, 4.15, 4.17, 4.19 e 4.21 foram construídos gráficos com a representação das frentes finais em separado, obtidas com cada algoritmo, MO-CORS e NSGA II. No caso de MO-CORS, foram também construídos gráficos com a representação dos pontos da frente, oriundos de cada problema auxiliar, e da evolução da quantidade de pontos pertencentes ao conjunto não dominado em cada iteração. A seguir nas Figuras 4.22, 4.23 e 4.24 ilustram-se três casos representativos de problemas com espaço objetivo bidimensional.

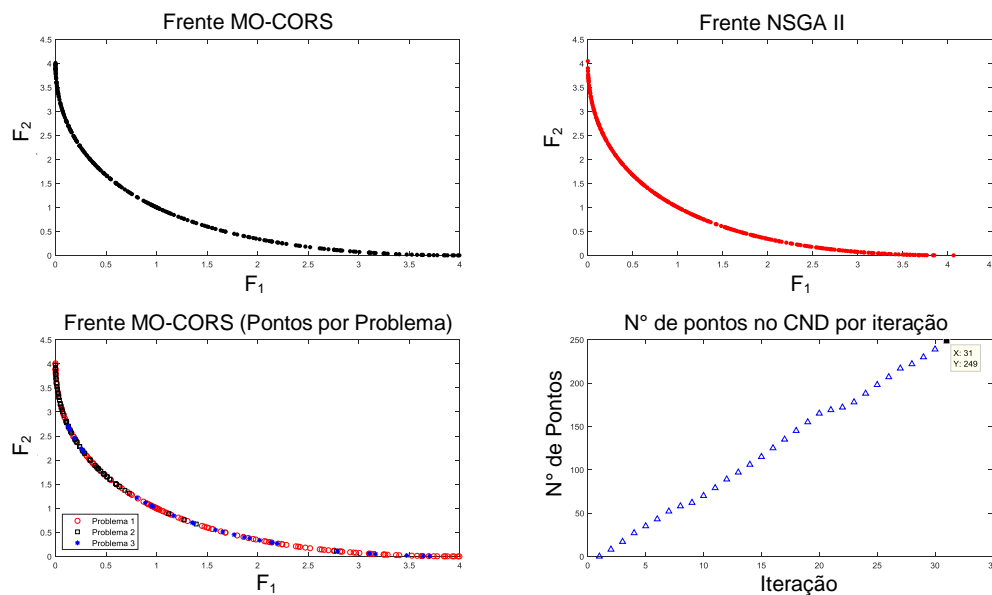


Figura 4.22: Frentes de Pareto e pontos por problema e por iteração (MOP1).

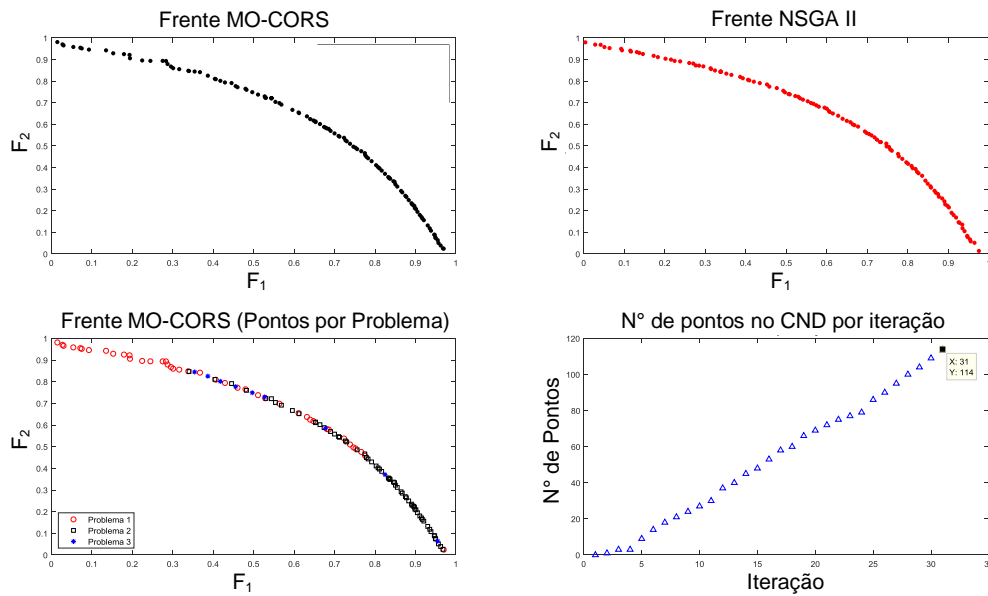


Figura 4.23: Frentes de Pareto e pontos por problema e por iteração (MOP2).

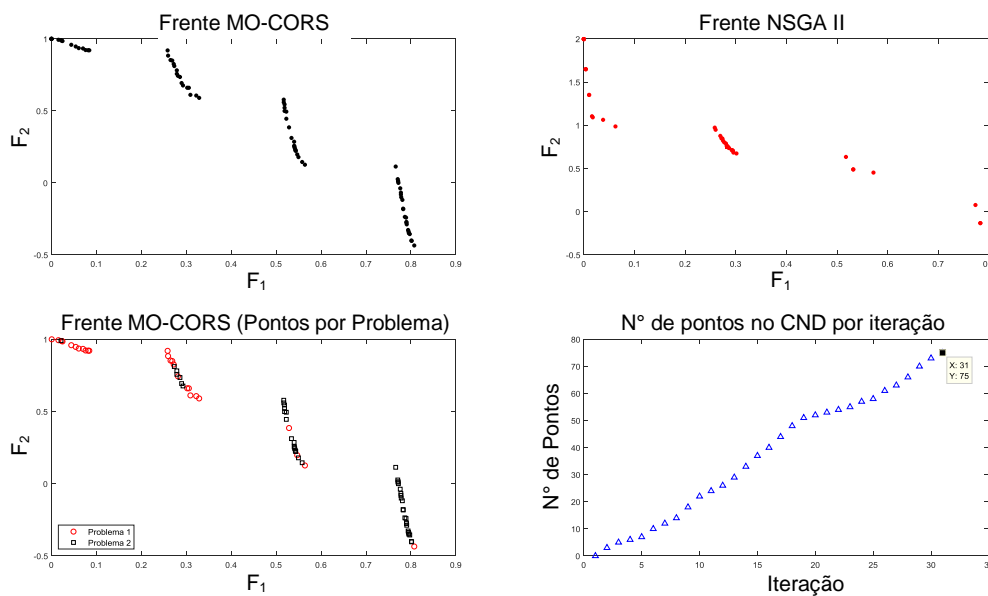


Figura 4.24: Frentes de Pareto e pontos por problema e por iteração (MOP6).

Mesmo apresentados de maneira ilustrativa, os gráficos para os resultados representativos levam em conta as situações de convexidade (não convexidade) e continuidade (descontinuidade) da frente de Pareto. Para todos os problemas de teste em geral, foi observado que o problema de otimização auxiliar 1 (efeito da busca CORS) tende a colocar pontos nos extremos da frente, ou dos ramos da frente no caso das frentes descontínuas. Isso ocorre tanto para frentes convexas como não convexas. Da mesma forma foi observado que a inclinação da reta que aproxima o gráfico do número de pontos da frente em cada iteração é sempre crescente, como é ilustrado nas Figuras 4.22, 4.23 e 4.24 para problemas de espaço objetivo

bidimensional. Na figura 4.25 ilustra-se para o caso de espaço objetivo tridimensional. Note-se a queda de pontos não dominados depois da iteração 5, a qual é devida a aparição, na sexta iteração, de pontos que dominam diversos pontos determinados em iterações anteriores.

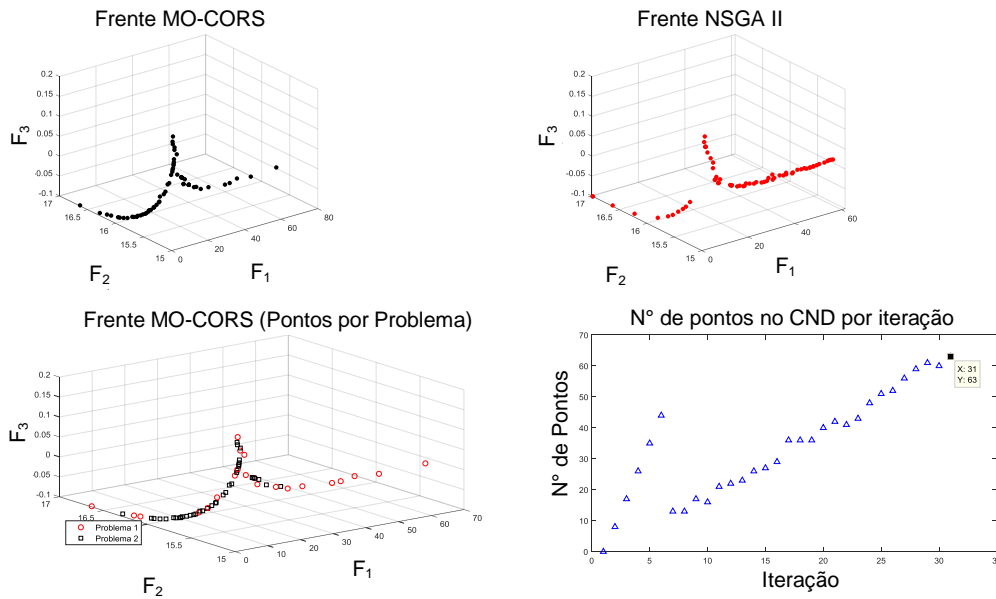


Figura 4.25: Frentes de Pareto e pontos por problema e por iteração (MOP5).

Tabela 4.10: Contribuição de pontos de cada problema de otimização auxiliar.

Problemas de Teste	Problema de Otimização Auxiliar			Número de Pontos na Frente
	Prob. 1 (%)	Prob. 2 (%)	Prob. 3 (%)	
MOP1	47,32	35,71	16,96	214,64
MOP2	33,33	60,36	6,31	121,36
MOP3	34,48	45,98	19,54	50,70
MOP4	16,59	77,97	5,08	69,59
MOP5	52,31	47,69	0,00	68,58
MOP6	33,33	66,66	0,00	74,17
MOP7	35,15	59,90	4,95	96,66
ZDT1	19,35	54,84	25,81	90,61
ZDT2	35,63	48,13	16,25	68,26

Na Tabela 4.10 são apresentados os valores médios das percentagens de pontos oriundos de cada problema auxiliar na frente final de cada problema de teste, assim como a média do número total de pontos obtidos em cada frente de Pareto. Dentre os resultados ressalta-se o pequeno aporte feito pelo problema auxiliar 3, mas tendo em conta que ele gera unicamente um ponto candidato em cada iteração por 80 gerados pelos problemas restantes, seu aporte acaba sendo significativo no processo total.

Capítulo 5

Aplicações em Problemas de Aerodinâmica e Máquinas De Fluxo

No intuito de avaliar o desempenho de MO-CORS em problemas reais, tanto para problemas mono-objetivo como multi-objetivo, neste capítulo apresentam-se testes do algoritmo em três aplicações aerodinâmicas de engenharia. A primeira aplicação foi num problema de otimização mono-objetivo em que foi maximizado o coeficiente de potência C_p , devido à variação da torção geométrica da pá de uma turbina eólica de eixo horizontal. A segunda aplicação foi em dois problemas multi-objetivo de projeto preliminar de asas de aeronaves subsônicas onde foram minimizados o módulo do coeficiente de momento $|C_m|$ e o inverso do coeficiente de sustentação $1/C_L$: um primeiro problema com a variação dos parâmetros geométricos da asa e um segundo com a variação da forma geométrica do aerofólio. A última aplicação neste trabalho foi num problema de otimização geométrica de uma grade linear de turbomáquina axial, no intuito de minimizar as perdas de energia. Em princípio, trata-se de um problema tipicamente mono-objetivo, porém transformado em multi-objetivo, acrescentando a minimização do desvio percentual da deflexão angular do escoamento em relação à geometria de base.

5.1. Otimização das Pás de uma Turbina Eólica.

Uma das metodologias muito utilizadas na literatura para projeto de turbinas eólicas nas últimas décadas baseia-se na teoria de quantidade de movimento do elemento de pá (em inglês, *Blade Element Momentum method* – BEM). Dentre os trabalhos que aplicaram BEM para projeto de pás de turbinas eólicas citam-se alguns: (DUAN e ZHAO,2010), (SONG *et al.*, 2011) e (MAHMUDDIN, 2017). A teoria BEM tem sido testada e comparada em muitos trabalhos com resultados experimentais (VELÁSQUEZ *et al.*, 2014) e com metodologias baseadas em dinâmica

de fluidos computacional CFD (BANGGA, 2018). Isso no intuito de verificar o quanto os resultados obtidos com o BEM (baseado em uma teoria bidimensional) se aproximam dos resultados de métodos mais confiáveis baseados em modelagem 3D que são computacionalmente bem mais custosos. Neste sentido foi desenvolvida uma função objetivo para o cálculo do coeficiente de potência baseada na teoria BEM, visando uma metodologia para otimização aerodinâmica de pás de turbinas eólicas de eixo horizontal, com base na construção de metamodelos, além de testar o desempenho do MO-CORS num problema de otimização real mono-objetivo.

5.1.1. Teoria de quantidade de movimento do elemento da pá (BEM)

O método BEM para projeto de turbinas eólicas é baseado no desenvolvimento das equações de conservação para um volume de controle que envolve o disco rotor. Assim, são determinadas as expressões para cálculo da potência P e o empuxo T , em função da velocidade axial média do ar no disco rotor, u , da velocidade do escoamento não perturbado, V_0 , da velocidade na esteira desenvolvida, u_1 , da massa específica do ar, ρ , da área do disco rotor, A , e da resultante das forças devidas à pressão atmosférica F_{pres} (HANSEN, 2015).

$$P = \frac{1}{2} \rho u A (V_0^2 - u_1^2); \quad u = \frac{V_0 + u_1}{2} \quad (5.1)$$

$$T = \frac{1}{2} \rho u A (V_0 - u_1) + F_{pres} \quad (5.2)$$

Considerando que u diminui através do rotor, essa componente é calculada, numa seção de pá r_i por meio de $V_0(1 - a)$; e, ainda, que, a componente tangencial relativa aumenta devido à velocidade angular ω induzida no rotor, essa componente é calculada por meio de $\omega r_i(1 + a')$. A Figura 5.1, ilustra essa situação, sendo a e a' os fatores de indução axial e tangencial numa seção de pá.

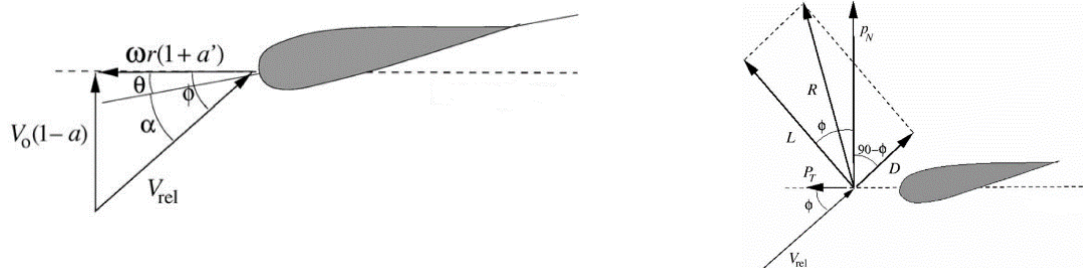


Figura 5.1: Ângulos, velocidades e forças em cada seção da pá.
Fonte: Hansen (2015).

5.1.2. Problema de otimização da pá

O problema de otimização consiste em determinar o ângulo de torção β e o ângulo de montagem θ_M ótimos da pá, que maximize o coeficiente de potência $C_p(\mathbf{X})$ de uma turbina eólica de eixo horizontal. O espaço de busca do vetor de decisão $\mathbf{X} = (\theta_M, \beta)$ está limitado por restrições laterais definidas em torno da configuração geométrica da turbina base de Velásquez *et al.* (2014). Na Figura 5.2 mostra-se um desenho para identificação das variáveis de projeto.

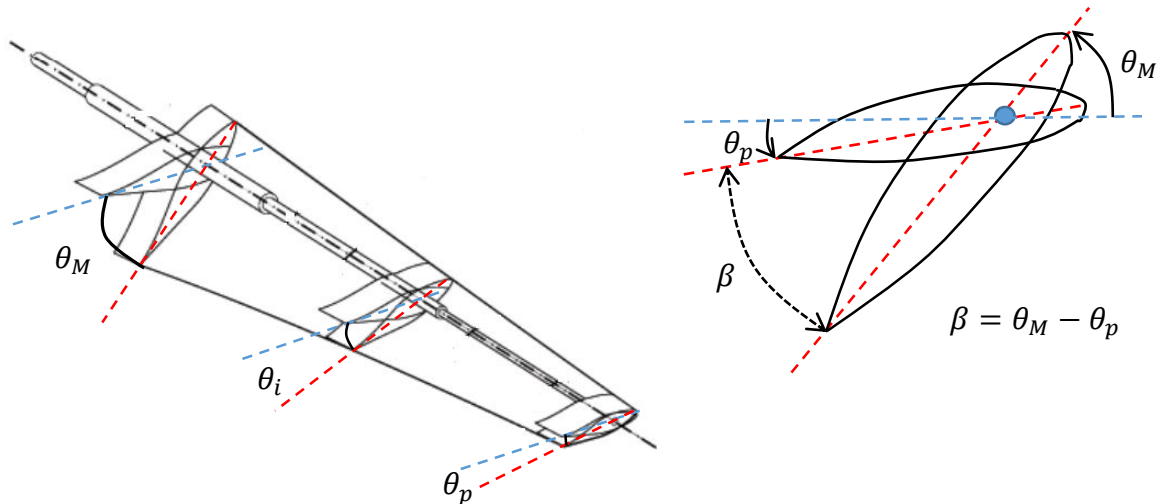


Figura 5.2: Torção da pá

Os limites laterais de cada variável de projeto e os parâmetros geométricos fixos são obtidos da turbina base de Velásquez *et al.* (2014) cujos valores são mostrados nas Tabelas 5.1 e 5.2. A potência nominal dessa turbina é de 10 kW.

Tabela 5.1: Dados do projeto base. (VELÁSQUEZ *et al.*, 2014)

Parâmetro	Valor
N_p (Número de pás)	3
R_c (Raio no cubo)	2,8 m
R_t (Raio na ponta)	28 m
N_{sec} (Número de seções)	20
θ_M	30°
θ_p	0°
β	30°
V_0 (Velocidade do vento)	8 m/s
Aerofólio Base	NREL S830

Considerando que os valores das variáveis de projeto da turbina base devem estar contidos dentro da região viável, na Tabela 5.1 mostram-se os limites laterais de cada variável de projeto seguindo a nomenclatura definida no capítulo 2 seção 2.1.

Tabela 5.2: Limites superiores e inferiores das variáveis de projeto. (Problema da pá)

Variável	x_{LI}	Base	x_{LS}
θ_M	0°	30°	40°
β	0°	30°	40°

5.1.3. A função objetivo para cálculo de C_p

Baseado nas variáveis ilustradas na Figura 5.1, Hansen (2015) define o seguinte algoritmo para o cálculo do coeficiente de potência.

Passo 1: Seleciona-se aleatoriamente os valores iniciais dos fatores de indução axial a e tangencial a' .

Passo 2: Cálculo do ângulo de incidência ϕ da velocidade relativa na seção através da Equação 5.3.

$$\tan \phi = \frac{(1-a)V_o}{(1+a')\omega r} \quad \text{ou} \quad \tan \phi = \frac{(1-a)}{(1+a')\lambda} \quad (5.3)$$

em que V_o , é a velocidade de incidência do vento (axial) e λ , é a relação entre as velocidades tangencial e axial.

Passo 3: Cálculo do ângulo de ataque da seção α , através da Equação 5.4.

$$\alpha = \phi - \theta \quad (5.4)$$

Passo 4: Cálculo dos valores dos coeficientes de sustentação C_l e de arrasto C_d do aerofólio da seção para o ângulo de ataque calculado no passo 3.

Passo 5: Cálculo do fator de perdas de Prandtl F_p , através da Equação 5.5

$$F_p = \frac{2}{\pi} \cos^{-1} \left(\exp \left(-\frac{N_p R_p - r}{2 r \sin \phi} \right) \right) \quad (5.5)$$

em que N_p , é o número de pás da turbina e R_p o raio do rotor na ponta da pá.

Passo 6: Cálculo dos novos valores dos fatores de indução de velocidade axial e tangencial, através das Equações 5.6 e 5.7.

$$a_{it} = \frac{1}{\frac{4F_P \sin^2 \phi}{\sigma C_n} + 1} \quad (5.6)$$

$$a'_{it} = \frac{1}{\frac{4F_P \sin \phi \cos \phi}{\sigma C_t} - 1} \quad (5.7)$$

em que C_n e C_t , são as componentes axial e tangencial da força aerodinâmica R . Segundo a Figura 5.1 estas componentes calculam-se através das expressões 5.8 e 5.9.

$$C_n = C_l \cos \phi + C_d \sin \phi \quad (5.8)$$

$$C_t = C_l \sin \phi - C_d \cos \phi \quad (5.9)$$

Passo 7: Verifica-se a convergência dos valores dos fatores de indução de velocidade através da Equação 5.10. Se não é verificada a convergência ($\varepsilon = 0,00001$) volta ao passo 2.

$$|a_{it} - a_{it-1}| < \varepsilon, |a'_{it} - a'_{it-1}| < \varepsilon; \quad t \leftarrow t + 1 \quad (5.10)$$

Passo 8: Cálculo do coeficiente de empuxo $(C_T)_i$, da seção i através da correlação de Glauert.

$$(C_T)_i = \begin{cases} 4a_{it}(1 - a_{it})F_P & a \leq \frac{1}{3} \\ 4a_{it} \left(1 - \frac{1}{4}(5 - 3a_{it})a_{it}\right) F_P & a > \frac{1}{3} \end{cases} \quad (5.11)$$

Passo 9: Cálculo do coeficiente de potência C_p , da seção i

$$(C_p)_i = \frac{8}{\lambda^2} F_{Pt} a'_{it} (1 - a_{it}) \lambda_i^3 (\Delta r)_i \quad (5.12)$$

em que λ_i , λ e λ_c , são as relações de velocidades tangencial e axial ($\omega r/V_0$) na seção genérica r_i , na ponta e no cubo da pá, respectivamente. $(\Delta r)_i$ É a diferença de raios entre seções consecutivas.

Passo 10: Fim. A saída é a distribuição de coeficientes de empuxo e potência ao longo da pá, já que o algoritmo se repete em cada seção $i = 1, \dots, N$.

É importante mencionar que um dos passos mais importantes do algoritmo é o cálculo dos coeficientes aerodinâmicos do aerofólio de cada seção em forma iterativa, segundo o ângulo de ataque α da velocidade relativa do ar em relação à corda. Nesta pesquisa foi utilizado um código computacional baseado no método dos painéis 3D (Ver seção 5.2.1) com camada limite bidimensional, que permite ângulos de ataque de até 15 graus, adaptado por Diaz *et al.* (2017) dos procedimentos de Katz e Plotkin (2001) e Moran (2003), no intuito de calcular os coeficientes de forma automática. Esta inclusão dentro do algoritmo incrementa o custo computacional o qual é diminuído através do uso de MO-CORS.

5.1.4. Metodologia de projeto otimizado das pás

O ponto de projeto foi definido mantendo fixas a velocidade angular e a velocidade do vento. Sendo assim, o parâmetro de relação de velocidades na ponta da pá é fixado em 8 como na turbina base. Igualmente tanto os raios como as cordas no cubo e na ponta da pá foram mantidos fixos durante todo o processo de otimização. Para a distribuição de cordas, foi considerada, a fórmula desenvolvida por Letcher (2017), (Equação 5.13).

$$C(r) = \frac{8\pi r \sin \phi}{3N_p \lambda} \quad (5.13)$$

A distribuição de raios de cada elemento ao longo da pá foi feita seguindo a lei cosenoidal da Equação 5.14.

$$\left(\frac{r}{R_T}\right)_i = R_T \cos \psi_i ; i = 1, \dots, N_{sec} \quad (5.14)$$

em que ψ_i , é determinado através da Equação 5.15.

$$\psi_i = \left(\frac{\cos^{-1}\left(\frac{R_c}{R_T}\right)}{N_{sec}}\right) \times j ; i = 1, \dots, N_{sec} ; j = N_{sec}, \dots, 1 \quad (5.15)$$

Na Figura 5.3 ilustra-se de forma representativa como foi obtida a distribuição de raios adimensional $(r/R_T)_i$, segundo a medida do ângulo ψ_i determinado para cada seção da pá da turbina eólica. A Equação (5.15) faz com que se distribua maior número de seções na ponta da pá.

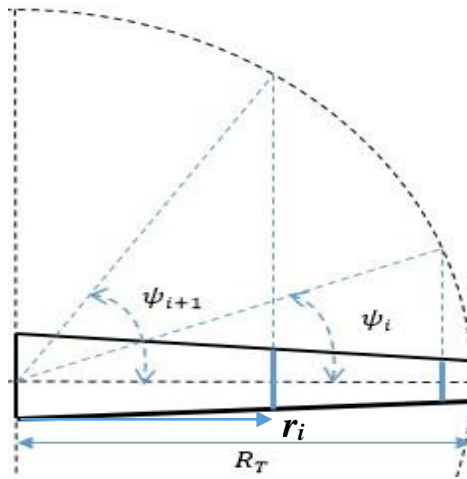


Figura 5.3: Ângulo ψ para a distribuição de raios.

O aerofólio de cada elemento de pá foi gerado através de um algoritmo baseado em curvas de Bézier (DÍAZ *et al*, 2017). As variáveis de projeto (θ_M, β), são relacionadas mantendo uma distribuição quadrática do ângulo de montagem de cada secção θ_i , ver Figura 5.1. Desta forma $\theta(r) = ar^2 + br + c$. Utilizando as condições que: $\theta(R_T) = \theta_p$, $\theta(R_c) = \theta_M$ e que o ângulo de torção mínimo encontra-se na ponta da pá, $\frac{d\theta}{dr}(R_T) = 0$, os parâmetros a, b e c podem ser determinados através da resolução do sistema de equações linear da expressão (5.16).

$$\begin{bmatrix} R_c^2 & R_c & 1 \\ R_T^2 & R_T & 1 \\ 2R_T & 1 & 0 \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} = \begin{Bmatrix} \theta_M \\ \theta_M - \beta \\ 0 \end{Bmatrix} \quad (5.16)$$

Note-se que o θ_p foi substituído por $\theta_M - \beta$, como se pode deduzir na Figura 5.2.

5.1.5. Resultados dos processos de otimização das pás

A Tabela 5.3 apresenta os resultados dos ângulos de montagem no cubo e a torção na ponta da pá que maximizam o coeficiente de potência do processo de otimização através dos algoritmos NSGA II e MO-CORS, além dos valores referentes à pá base.

Tabela 5.3: Resultados da otimização da pá da turbina eólica.

Algoritmo	θ_M (°)	β (°)	C_p
NSGA II	7,00	8,55	0,5111
MO-CORS	9,44	11,97	0,5196
Base	30,00	30,00	0,4150

As curvas da Figura 5.4 apresentam as distribuições quadráticas de torção para cada projeto otimizado pelos algoritmos NSGA II e MO-CORS.

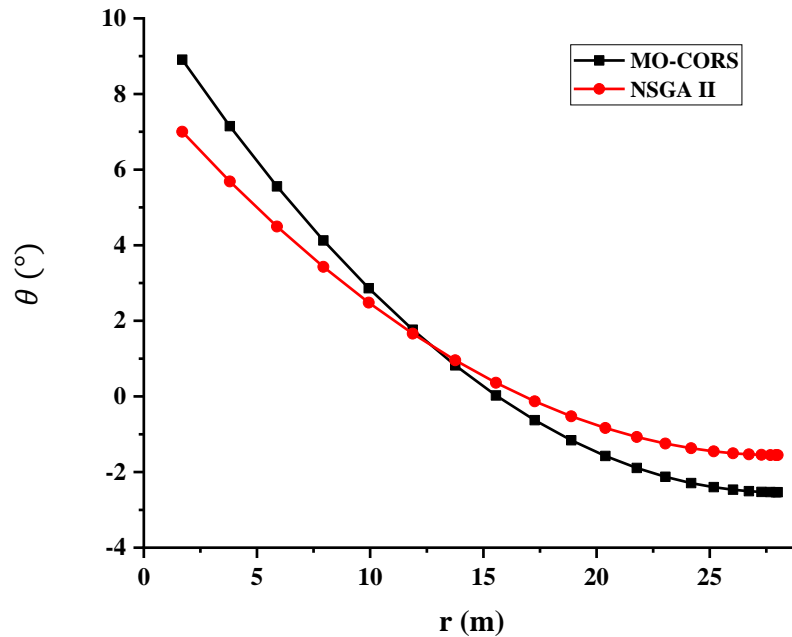


Figura 5.4: Distribuições de torção ao longo da pá (MO-CORS e NSGA II)

Quanto aos valores máximos do coeficiente de potência C_p obtidos, observa-se que tanto NSGA II como MO-CORS atingiram valores maiores que o da turbina base com uma diferença aproximada de 0,1; no entanto, o valor atingido pelo MO-CORS foi ainda 0,009 superior ao atingido pelo NSGA II. Na Figura 5.5 mostram-se as evoluções dos valores de C_p , obtidos pelos algoritmos ao longo de 150 avaliações da função objetivo.

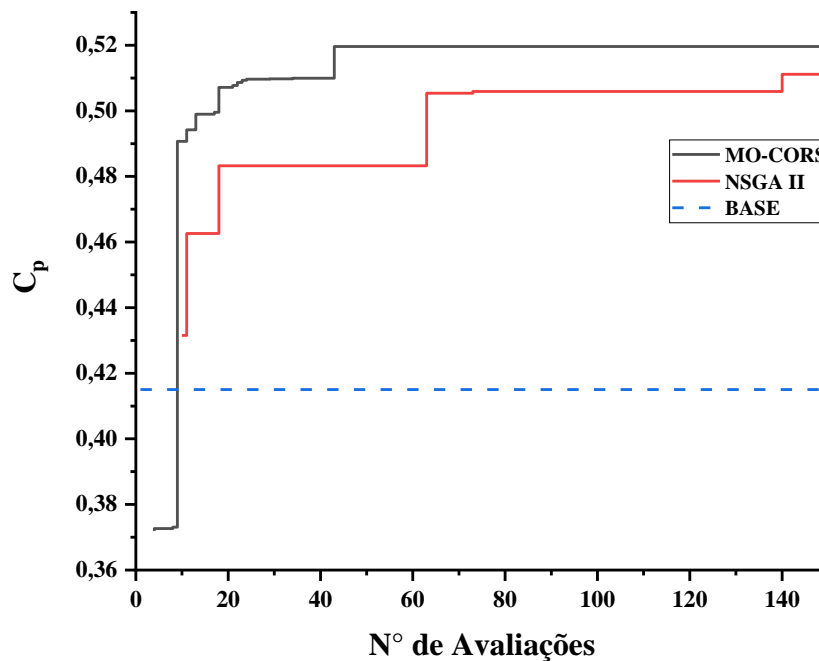


Figura 5.5: Evolução de MO-CORS e NSGA II .

Pelas evoluções mostradas na Figura 5.5 pode-se analisar outro aspecto importante, que é a aceleração aportada pelo algoritmo MO-CORS durante o processo de otimização. A aceleração neste problema foi calculada através do quociente entre o número de avaliações necessárias para os algoritmos atingirem 99% do melhor atingido pelo NSGA II. Para isso, MO-CORS e NSGA II precisaram de 18 e 73 avaliações, respectivamente. Nesse contexto, pode-se dizer que o MO-CORS obteve uma aceleração de 4,06 vezes em relação ao NSGA II.

5.1.6. Curvas operacionais dos projetos resultantes

Cada projeto otimizado e o base foram avaliados para valores de λ compreendidos entre 4 e 12, conforme a Tabela 5.4, mantendo fixa a velocidade do vento $V_0 = 8 \text{ m/s}$

Na Figura 5.6, observa-se que o máximo C_p para ambos algoritmos é muito alto em relação ao do projeto base. Isso provavelmente se deva a instabilidades do modelo utilizado para o cálculo do coeficiente de arrasto.

Tabela 5.4: Valores de C_p para cada pá

λ	C_p		
	Base	NSGA II	MO-CORS
4	0,1800	0,2614	0,2456
5	0,2550	0,3603	0,3870
6	0,3420	0,4501	0,4632
7	0,3960	0,4986	0,5104
8	0,4150	0,5111	0,5196
9	0,4001	0,4989	0,5100
10	0,3516	0,4499	0,4590
11	0,3010	0,4088	0,3988
12	0,2512	0,3301	0,3401

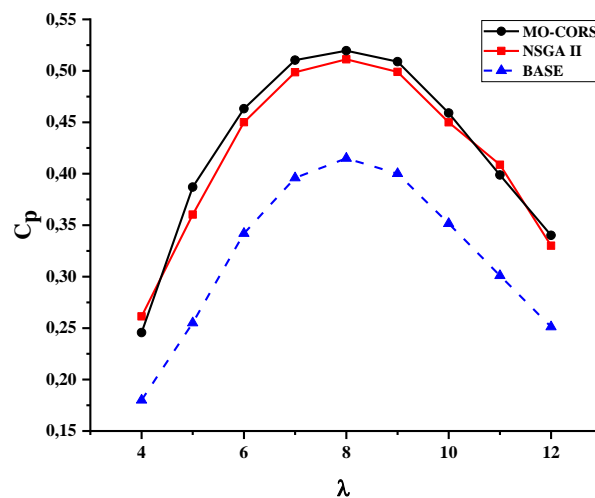


Figura 5.6: Curvas operacionais dos projetos otimizados

5.2. Otimização Aerodinâmica de Asas

Nesta seção apresentam-se os resultados da aplicação de MO-CORS em dois problemas clássicos de otimização multi-objetivo de aerodinâmica de asas de aeronaves subsônicas. Ambos os problemas tipo *min-min*, tratam da minimização do inverso do coeficiente de sustentação $f_1 = 1/C_L$ e o módulo do coeficiente de momento em relação ao bordo de ataque da asa $f_2 = |C_m|$. O projeto base foi a asa de uma aeronave agrícola desenvolvida por Bravo *et al.* (2018), onde é muito importante ter um coeficiente de sustentação alto e um coeficiente de momento próximo de zero. O ângulo de incidência fixo e igual ao do projeto base.

Os problemas testados são unicamente diferenciados nas variáveis de projeto ou espaço de decisão, sendo então que: no problema 1 as variáveis de projeto a considerar são 5 parâmetros geométricos da asa (mantido o aerofólio básico); no problema 2 são consideradas 10 variáveis para a modificação do formato do aerofólio apenas.

5.2.1. Função objetivo para problema de asas subsônicas.

A função objetivo utilizada para os problemas 1 e 2, é de tipo caixa preta em que os valores das funções são calculados através de um código computacional elaborado por Diaz *et al.* (2017), baseado no método dos painéis 3D (KAZT e PLOTKIN, 1991) com interação de camada limite (MORÁN, 2003).

O método dos painéis 3D é aqui utilizado para o cálculo da distribuição de pressões e velocidades ao redor de asas imersas num escoamento uniforme. Ele se baseia na teoria potencial em que o efeito de perturbação da asa e da esteira é representado por distribuições de dipolos de intensidade μ e fontes de intensidade σ (essas só do corpo) sobre as superfícies. É feita a prévia discretização das superfícies em painéis quadrilaterais com distribuições uniformes das singularidades em cada painel, para os quais se conhece a expressão analítica do potencial de velocidades induzido. Exceto em pontos singulares, cada distribuição introduzida representa uma solução da equação de Laplace. A superposição também irá representar uma solução da mesma:

$$\nabla^2\Phi = 0 \quad (5.17)$$

sendo Φ a soma dos potenciais da velocidade não perturbada Φ_∞ e dos potenciais induzidos pelas distribuições de fontes e dipolos colocadas nas superfícies do corpo s_b e da esteira s_w . Assim, o potencial total pode ser expresso pela Equação 5.18.

$$\Phi(x, y, z) = -\frac{1}{4\pi} \int_{s_b \cup s_w} \left[\sigma \left(\frac{1}{r} \right) - \mu \mathbf{n} \cdot \nabla \left(\frac{1}{r} \right) \right] dS + \Phi_\infty \quad (5.18)$$

em que r é a distância de cada ponto da superfície ao ponto de cálculo e \mathbf{n} é o vetor unitário normal à superfície. Note-se que $\sigma = 0$, em s_w . Introduzindo as discretizações geométrica e funcional do método dos painéis na Equação 5.18, e impondo a condição de potencial nulo no interior do corpo resulta a seguinte equação:

$$\sum_{k=1}^N C_k \mu_k + \sum_{l=1}^{N_w} C_l \mu_l = - \sum_{k=1}^N B_k \sigma_k \quad (5.19)$$

em que C_k e C_l , são os coeficientes de influência devido à distribuição de dipolos no corpo e na esteira respectivamente e B_k , são os coeficientes de influência devido à distribuição de fontes. A Equação 5.19 representa a condição de Dirichlet que deve ser aplicada a um ponto de colocação (de controle) de cada um dos N painéis do corpo e N_w painéis da esteira. Resulta assim um conjunto de N equações algébricas lineares a ser resolvido para N intensidades de dipolos. Para isso, as intensidades de dipolos na esteira são expressas pelo salto do valor de μ entre os painéis que formam o bordo de fuga (condição de Kutta). As intensidades das fontes são igualadas às componentes normais da velocidade não perturbada \mathbf{Q}_∞ :

$$\sigma_k = \mathbf{n}_k \cdot \mathbf{Q}_\infty \quad (5.20)$$

Determinados os valores das intensidades de dipolos, são calculados os vetores de perturbação de velocidades em cada painel, nas direções da corda da asa l , da envergadura m e normal ao painel n .

$$q_l = \frac{\partial \mu}{\partial l}; \quad q_m = \frac{\partial \mu}{\partial m}; \quad q_n = -\sigma, \quad (5.21)$$

As distribuições de velocidades e de coeficientes de pressão são calculadas através das Equações 5.22 e 5.23.

$$\mathbf{Q}_k = (Q_{\infty l}, Q_{\infty m}, Q_{\infty n})_k + (q_l, q_m, q_n)_k \quad (5.22)$$

$$Cp_k = 1 - \frac{Q_k^2}{Q_\infty^2} \quad (5.23)$$

Conhecida a distribuição de Cp , são calculados os coeficientes de sustentação C_l , arrasto induzido C_{di} e momento C_m (KAZT e PLOTKIN, 2001).

Uma vez que o método dos painéis é puramente potencial, ao código desenvolvido foi agregado um procedimento para o cálculo do coeficiente de atrito C_f baseado no método integral de camada limite bidimensional (2DBLM), especificamente com as correlações de Thwaites e Head nos casos de regiões de escoamento laminar e turbulento respectivamente. (MORAN, 2003). Note-se que, nas aplicações desta seção, as funções objetivo não envolvem nesta pesquisa o cálculo do coeficiente de arrasto, mas o cálculo dele foi considerado no problema de otimização da pá de turbinas eólicas de eixo horizontal.

5.2.2. Problemas de otimização de asas subsônicas.

Problema de otimização 1 (Geometria da asa).

Este problema de otimização consiste em determinar a frente de Pareto bidimensional produzida ao se minimizar as funções objetivo f_1 e f_2 , devida à variação dos seguintes parâmetros geométricos da asa: corda na raiz C_0 , a relação de afilamento $\lambda_a = C_t/C_0$, (em que C_t é a corda na ponta da asa), o ângulo de enflechamento Γ , o ângulo de diedro ε_d e a envergadura b , ilustrados na Figura 5.7. Sendo assim, o vetor de decisão fica definido como: $\mathbf{X} = (C_0, \lambda_a, \Gamma, \varepsilon_d, b)$. Os limites laterais que delimitam a região viável são determinados dos valores médios de cada variável, de um conjunto de 10 aeronaves semelhantes (de uso agrícola) referenciadas por Jackson (2005). Cuidou-se para que os valores da asa base do trabalho de Bravo *et al.* (2018) estejam contidos entre as medias calculadas de cada variável. Estes valores são mostrados na Tabela 5.5.

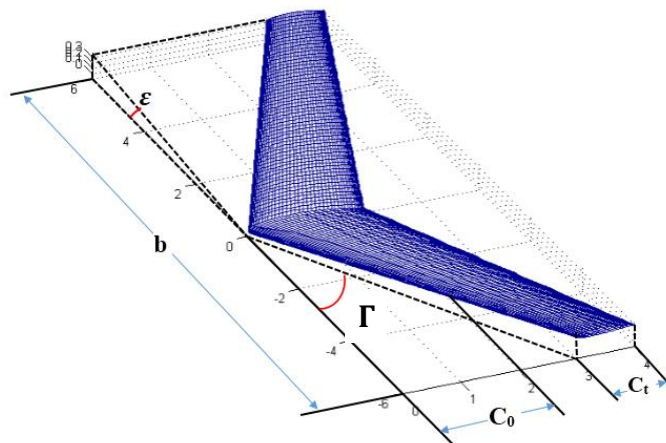


Figura 5.7: Parâmetros geométricos da asa.

Tabela 5.5: Valores da asa base e limites laterais (Problema 1 da asa).

X	x_{LI}	Asa Base	x_{LS}
C_0	0,9375 m	1,875 m	2,5500 m
λ_a	0,4	1	1
Γ	-2°	0°	10°
ε_d	0°	7°	10°
b	11,452 m	14,800 m	16,175 m
Aerofólio	N/A	NACA 23015	N/A

Problema de otimização 2 (Geometria do aerofólio).

Este problema de otimização consiste em determinar a frente de Pareto bidimensional produzida ao se minimizar as funções objetivo f_1 e f_2 , devido à variação das 10 ordenadas Z dos pontos de controle de duas curvas de Bézier de grau 6, que controlam a geometria do extradorso e intradorso do aerofólio.

Os pontos de controle para os aerofólios foram gerados por um código de refinamento da distribuição do parâmetro da curva Bézier (Equações 5.24 e 5.25), de modo a minimizar o erro quadrático entre as coordenadas da curva e as coordenadas tabeladas do aerofólio base (NACA 23015). A Figura 5.8 mostra os resultados do aerofólio NACA 23015 parametrizado para 120 pontos, assim como, a numeração de cada ponto de controle de extradorso e intradorso do aerofólio

$$x_p(t) = \sum_{i=1}^n \frac{n!}{i!(n-1)!} t^i (1-t)^{n-i} X_{p_i} \quad (5.24)$$

$$z_p(t) = \sum_{i=1}^n \frac{n!}{i!(n-1)!} t^i (1-t)^{n-i} Z_{p_i} \quad (5.25)$$

em que as coordenadas dos pontos parametrizados do aerofólio são (x_p, z_p) e as coordenadas dos pontos de controle do polinômio de Bézier de grau n são (X_p, Z_p) .

A distribuição do parâmetro t ao longo das curvas Bézier é feita de acordo com uma lei co-senoidal, concentrando mais pontos nas regiões dos bordos de ataque e de fuga (Sousa, 2008). As coordenadas X_{p_i} , são mantidas fixas em $\mathbf{X}_p = (0 \ 0,2 \ 0,4 \ 0,6 \ 0,8 \ 1)$.

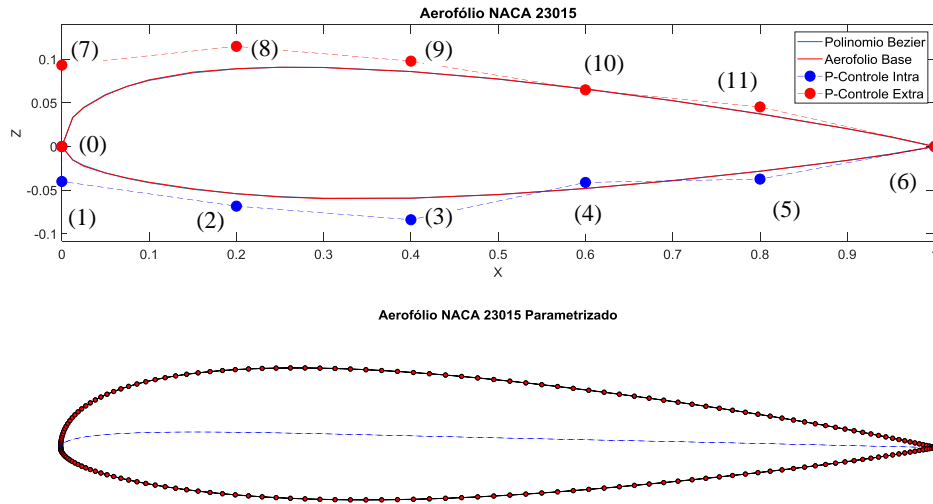


Figura 5.8: Aerofólio NACA 23015 parametrizado com curvas de Bézier

Neste problema é perturbado o vetor de decisão $\mathbf{X} = (Z_{p_1}, \dots, Z_{p_i}), i = 1, \dots, 11; i \neq 6$, dentro da região viável definida pelos limites laterais mostrados na Tabela 5.6.

Tabela 5.6: Valores da asa base e limites laterais (Problema 2 da asa).

\mathbf{X}	x_{LI}	Asa Base NACA 23015	x_{LS}
Z_{p_1}	-0,0441	-0,0401	-0,0361
Z_{p_2}	-0,1025	-0,0683	-0,0512
Z_{p_3}	-0,1258	-0,0839	-0,0629
Z_{p_4}	-0,0617	-0,0412	-0,0309
Z_{p_5}	-0,0411	-0,0373	-0,0336
Z_{p_7}	0,0839	0,0932	0,1026
Z_{p_8}	0,0573	0,1148	0,2295
Z_{p_9}	0,0489	0,0978	0,1956
$Z_{p_{10}}$	0,0325	0,0650	0,1299
$Z_{p_{11}}$	0,0585	0,0454	0,0714

Os limites superiores e inferiores foram calculados tendo em conta o critério de viabilidade geométrica empregado por Sousa (2008), onde estabelecem perturbações da ordenada Z_p dos pontos de controle em relação dos pontos de controle do aerofólio base, como é definido a seguir: (i) para os pontos 1,5,7 e 11, mostrados na Figura 5.8, a perturbação foi de $\pm 10\%$, (ii) para os pontos 8,9 e 10 as perturbações foram de -50% para o limite inferior e de $+100\%$ para o limite superior e (iii) para os pontos 2,3 e 4 as perturbações foram de -50% no

limite inferior e +25% no limite superior.

5.2.3. Resultados dos problemas de otimização de asas.

A seguir são apresentados para cada problema, quatro gráficos que ilustram o desempenho de MO-CORS em comparação com a aplicação direta do algoritmo NSGA II. Estes gráficos são, a representação: (i) da evolução dos conjuntos não dominados do algoritmo MO-CORS, (ii) da comparação entre as frentes de Pareto resultantes dos algoritmos MO-CORS e NSGA II, (iii) dos pontos da frente aportados por cada problema auxiliar de MO-CORS e (iv) a frente das frentes entre MO-CORS e NSGA II. Também foi apresentada uma tabela com os resultados das métricas de desempenho para cada problema aerodinâmico de asa. Para a aplicação de MO-CORS e NSGA II nestes problemas com fins comparativos forem, utilizados os mesmos critérios de comparação e parâmetros próprios de cada algoritmo usados nos problemas e funções de teste.

Resultados do problema de otimização 1.

A Figura 5.9 mostra a evolução do algoritmo MO-CORS num total de 427 avaliações da função objetivo. Embora não seja possível observar diretamente na Figura 5.9, em qual iteração o algoritmo MO-CORS já consegue encontrar pontos que dominam o projeto base, uma análise do banco de dados de cada iteração mostra que depois de 10 iterações foram atingidos dois projetos melhores que o de base no conjunto não dominado.

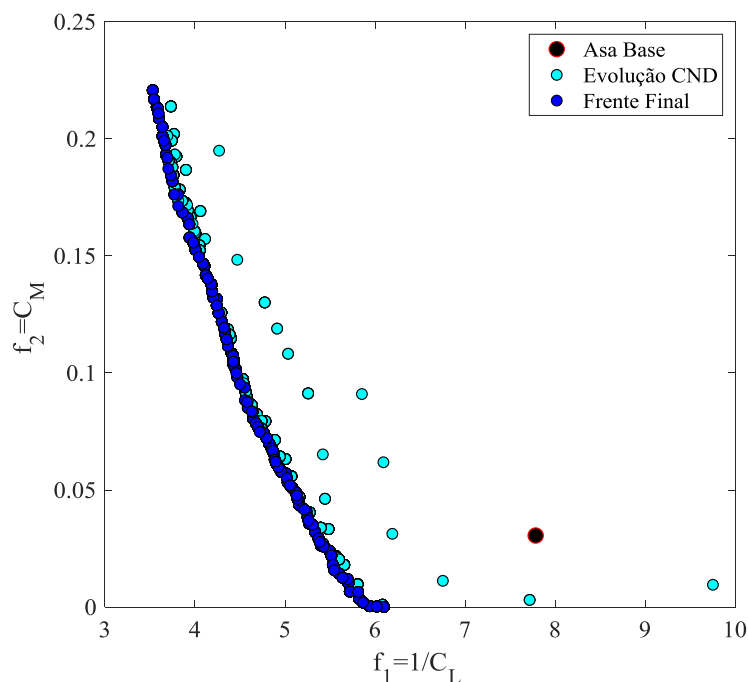
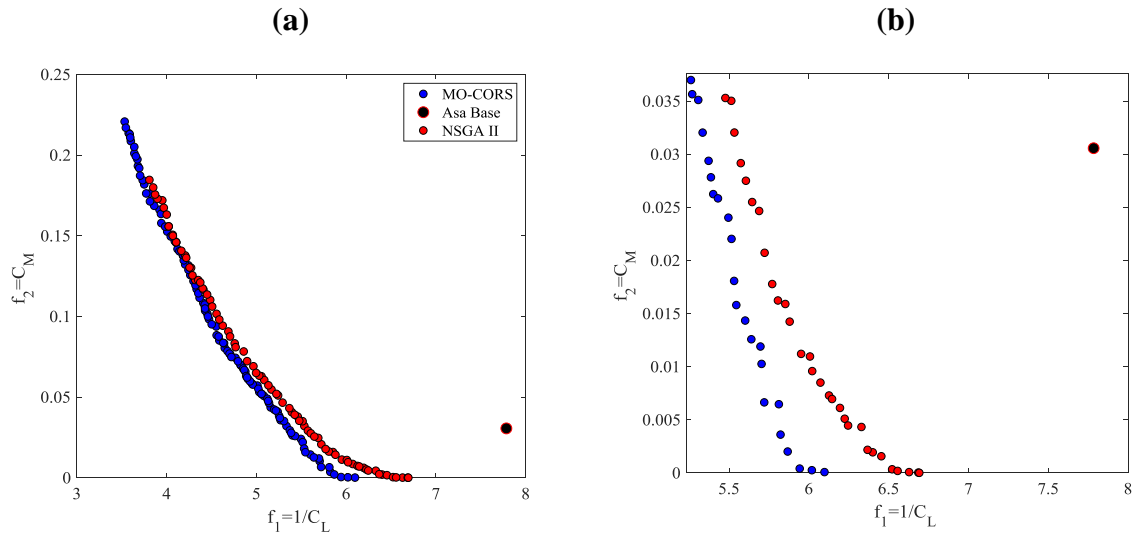


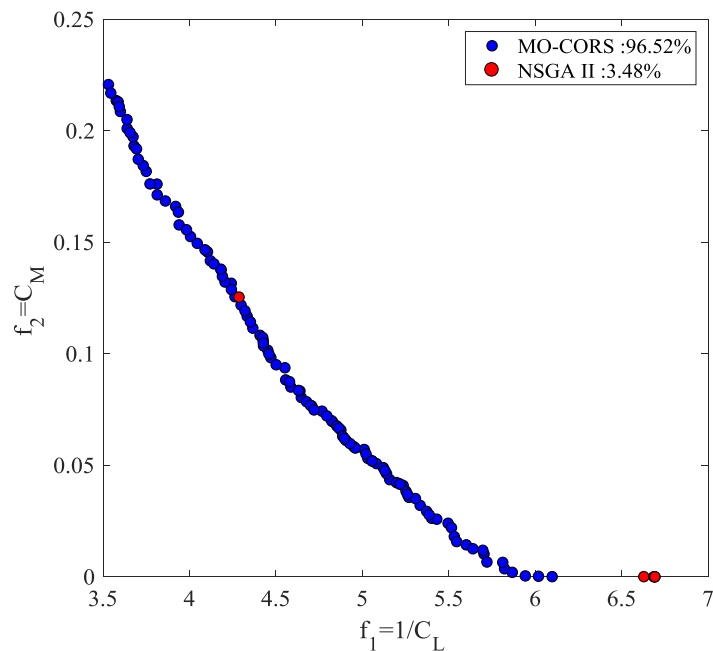
Figura 5.9: Evolução dos conjuntos não dominados MO-CORS (Variando Geometria da asa)

Na Figura 5.10 foi feita uma comparação direta entre as frentes de Pareto geradas pelos algoritmos MO-CORS e NSGA II, com ênfase nas proximidades do projeto base.



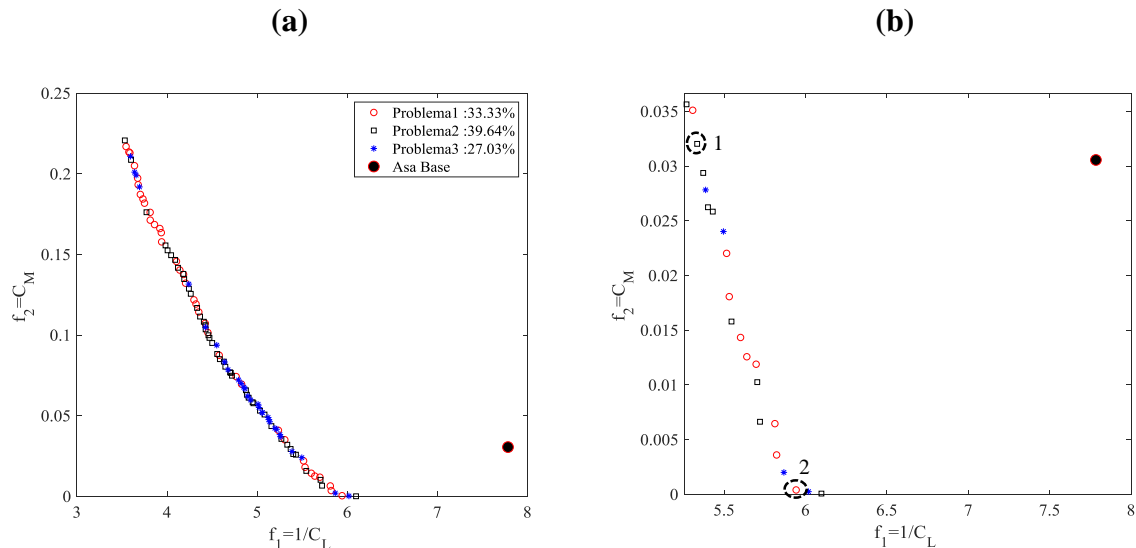
Figuras 5.10: Frentes de Pareto do Problema 1 de asas (a) Final (b) Ampliação

As frentes de Pareto da Figura 5.10a ilustram que o algoritmo MO-CORS consegue dispersar melhor as soluções neste problema, além de se observar um claro domínio das soluções de MO-CORS em relação às soluções de NSGA II. Na Figura 5.10b, onde mostra-se uma ampliação das frentes nas proximidades do projeto base, um total de 19 resultados de MO-CORS e 26 de NSGA II conseguem melhorar o projeto base, mas com um total domínio do conjunto de soluções de MO-CORS em relação ao de NSGA II. A seguir mostra-se a frente das frentes na Figura 5.11.



Figuras 5.11: Frente das frentes entre MO-CORS e NSGA II.

Na construção da Figura 5.11, conseguiu-se determinar a métrica percentual dos melhores valores na frente de Pareto *MVFP*, definida no capítulo 2, seção 2.5.5. Neste problema, o algoritmo MO-CORS contribui com o 96,52% dos pontos na frente das frentes contra 3,48% de NSGA II. É importante indicar que os valores da *MVFP* são calculados quando os algoritmos atingem a convergência definida na seção 5.2.3. O algoritmo MO-CORS precisou de 427 chamadas da função objetivo em 98 min, enquanto que NSGA II precisou de 2400 chamadas em 128 min, Logo, em termos de chamadas, o MO-CORS acelerou 5.62 vezes o processo.



Figuras 5.12: Contribuição dos problemas auxiliares de MO-CORS (Problema 1 da asa). (a) Final (b) Ampliação.

Observa-se na Figura 5.12a um razoável equilíbrio no aporte de pontos na frente de Pareto por parte de cada problema auxiliar. Embora a maioria se origine do problema auxiliar 2 (39,64%), os problemas auxiliares 1 e 3 contribuem com 33,33% 27,03% dos pontos, respectivamente. É interessante constatar que a maioria dos pontos aportados pelo problema auxiliar 1 tendem a se concentrar nos extremos da mesma (efeito da busca CORS). Na Figura 5.12b, observa-se que, nas proximidades do projeto base, dos 19 projetos da frente que dominam ele, 7 são colocados pelo problema auxiliar 2, 8 pelo problema auxiliar 1 e 4 pelo problema auxiliar 3.

Resultados das métricas de desempenho para o problema de otimização multi-objetivo do problema aerodinâmico de asa são mostrados na Tabela 5.7.

Tabela 5.7: Valores das métricas de desempenho para MO-CORS no problema 1 da asa.

Algoritmo	Métricas de desempenho (Capítulo 2)				N° de avaliações
	Espalhamento	Espaçamento	HRM	MVFP	
MO-CORS	0,4273	0,0140	0,2606	96,52%	427
NSGA II	0,9033	0,0030	0,0233	3,48%	2400

Uma boa otimização de asa para aeronaves agrícolas, consiste em aumentar o C_L , mantendo o C_m da aeronave bem próximo de zero. Ou seja, que o C_m da asa e do fuselagem sejam equilibrados pelo estabilizador horizontal em voo reto e nivelado. Sendo assim um bom projeto otimizado de asa de aeronaves agrícola é aquele que consegue aumentar o coeficiente de sustentação mantendo o coeficiente de momento longitudinal igual ao projeto base. Na Figura 5.12b, o projeto indicado com o números 1 representa uma das soluções da frente que cumprem com o critério pretendido. Mesmo assim, considerando também o critério de ter diversidade nos projetos solução foi escolhido para comparação aquele indicado com o número 2. A Tabela 5.8 mostra as variáveis de projeto e os vetores objetivo de cada solução e da base.

Tabela 5.8: Valores das variáveis de projeto das soluções 1 e 2 no problema 1 da asa.

Solução	Variáveis de decisão					Vetor de objetivos $f_i = (f_1 \ f_2)$
	C_0 (m)	λ_a	Γ (°)	ε_d (°)	b (m)	
1	2,1205	0,4012	-1,9841	5,7212	16,1647	(5,3340 0,0320)
2	1,5880	0,4265	-1,9968	0,8258	15,9978	(5,9424 0,0004)
Base	1,8750	1,0000	0,0000	7,0000	14,800	(7,7840 0,0301)

Na Figura 5.13 e na Tabela 5.8 observa-se que ambas as soluções buscam reduzir o ângulo de enflechamento para o limite inferior o qual minimiza o coeficiente de momento e também de aumentar levemente a envergadura para gerar um aumento do coeficiente de sustentação.

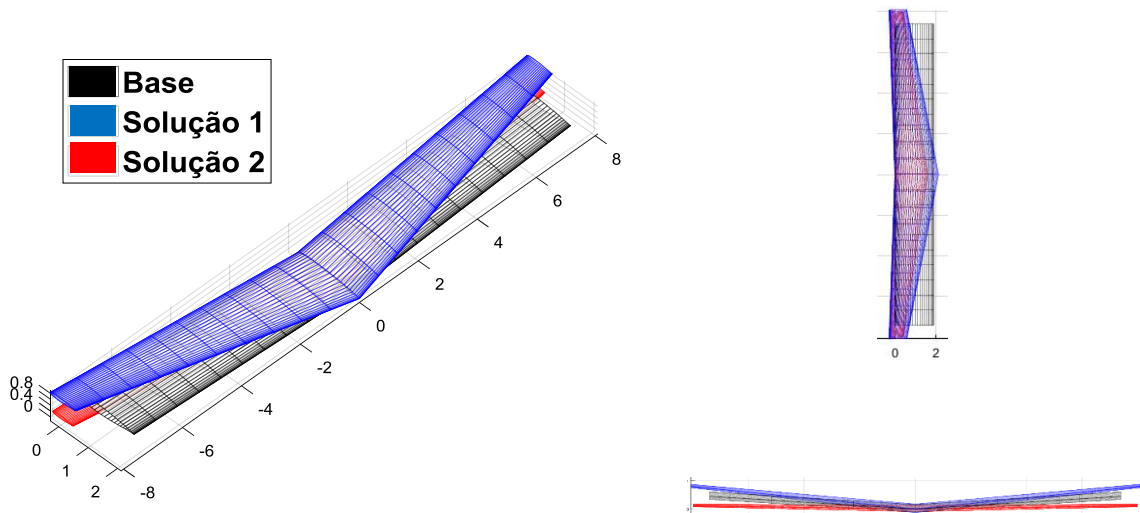


Figura 5.13: Isometria e vistas da asa base e das soluções 1 e 2.

Resultados do problema de otimização 2.

A Figura 5.14 ilustra a evolução do algoritmo MO-CORS num total de 402 avaliações da função objetivo. Uma análise do banco de dados de cada iteração, mostra que depois de 40 iterações foram atingidos dois projetos melhores que o de base no conjunto não dominado.

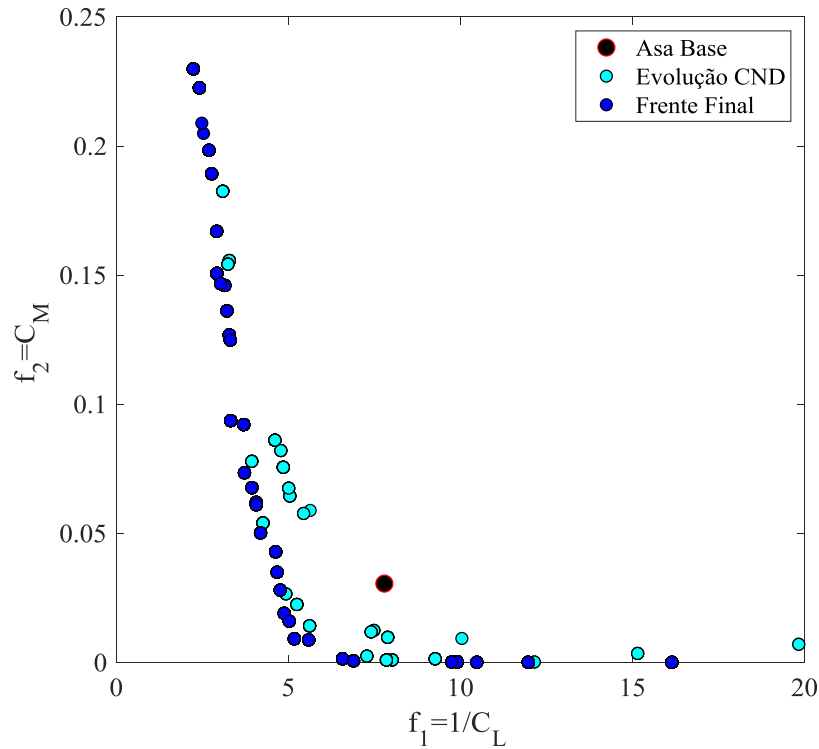
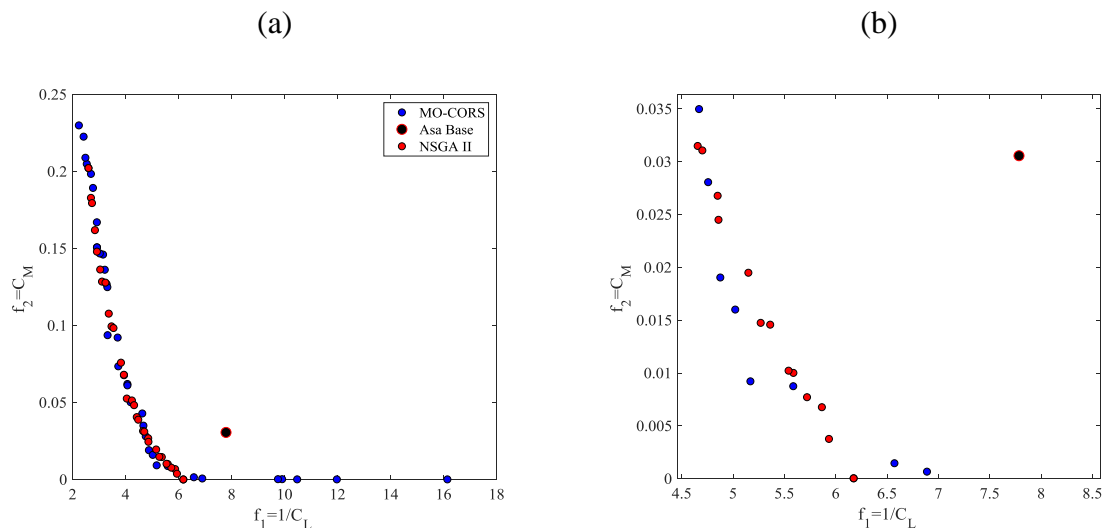


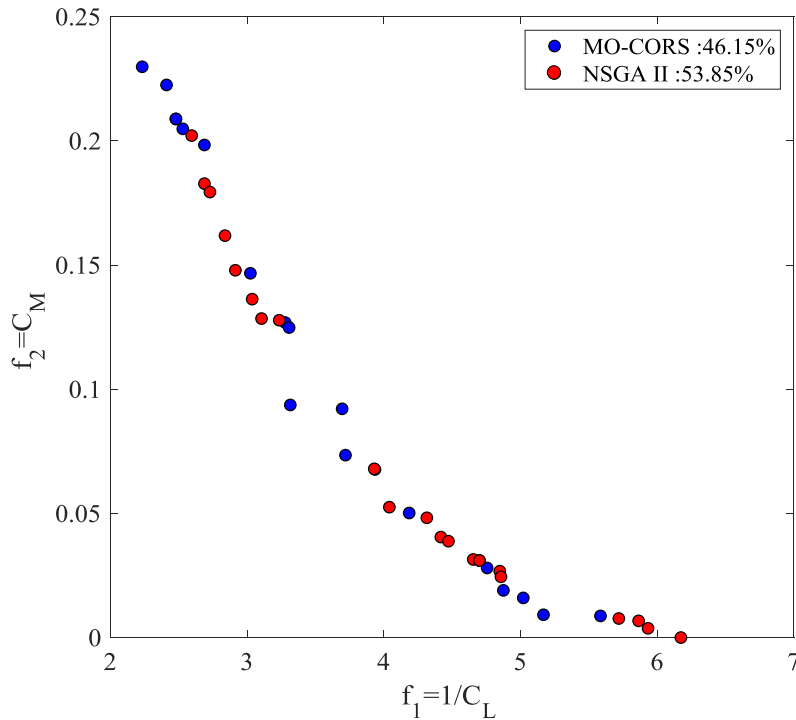
Figura 5.14: Evolução dos conjuntos não dominados MO-CORS (Variando Geometria do aerofólio)

Na Figura 5.15 foi feita uma comparação direta entre as frentes de Pareto geradas pelos algoritmos MO-CORS e NSGA II, com ênfase nas proximidades do projeto base.



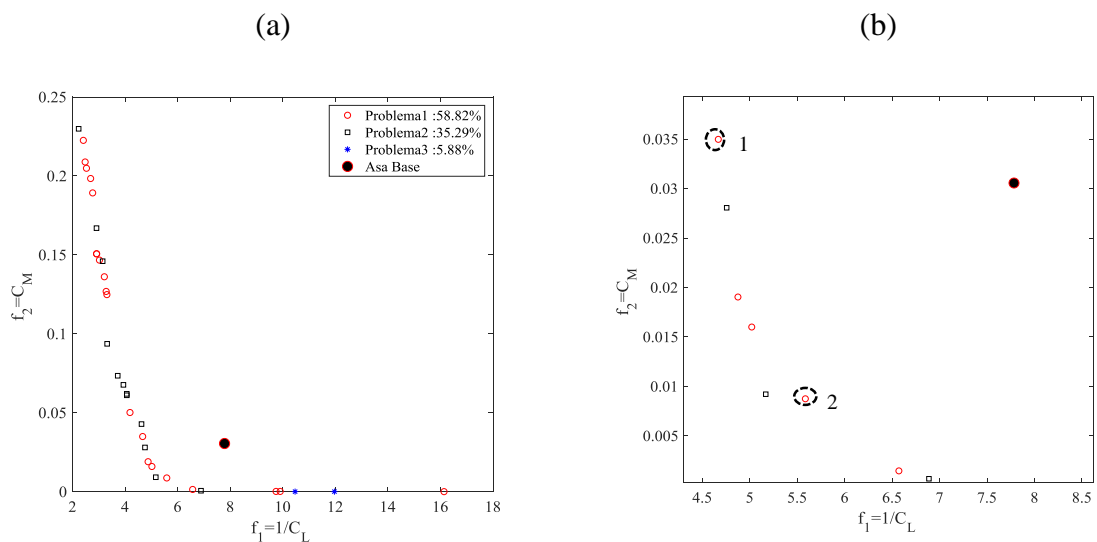
Figuras 5.15: Frentes de Pareto do Problema 2 de asas (a) Final (b) Ampliação

Assim como no problema 1, aqui as frentes de Pareto da Figura 5.15a ilustram que o algoritmo MO-CORS consegue dispersar melhor as soluções. Nas proximidades do ponto de projeto (Figura 5.10b), um total de 7 resultados de MO-CORS por 11 de NSGA II conseguem melhorar o projeto base.



Figuras 5.16: Frente das frentes entre MO-CORS e NSGA II.

Neste problema, o algoritmo MO-CORS contribuiu com 46,15% dos pontos na frente das frentes por 53,85% de NSGA II. O algoritmo MO-CORS precisou de 402 chamadas da função objetivo e 132 min, enquanto o NSGA II precisou de 2211 chamadas e 270 min. Logo, o MO-CORS acelerou ao menos 5,5 vezes, em chamadas, o processo de otimização.



Figuras 5.17: Contribuição dos problemas auxiliares de MO-CORS (problema 2 da asa). (a) Final (b) Ampliação.

A Figura 5.17a mostra neste caso um desequilíbrio bem acentuado no aporte de pontos na frente de Pareto por parte de cada problema auxiliar. Observa-se agora que a maioria dos pontos na frente de Pareto são colocados pelo problema auxiliar 1 (58,82%), enquanto os

problemas auxiliares 2 e 3 contribuem com 35,29% e 5,88%, respectivamente. Na Figura 5.17b, observa-se nas proximidades do projeto base que, dos 07 projetos da frente que o dominam, 04 são colocados pelo problema auxiliar 1 e 03 pelo problema auxiliar 2.

Resultados das métricas de desempenho para o problema de otimização multi-objetivo do problema aerodinâmico de asa, com variação da geometria do aerofólio, são mostrados na Tabela 5.9.

Tabela 5.9: Valores das métricas de desempenho para MO-CORS no problema 2 da asa.

Algoritmo	Métricas de desempenho (Capítulo 2)				N° de avaliações
	Espalhamento	Espaçamento	HRM	MVFP	
MO-CORS	0,8852	0,7356	0,0871	46,15%	402
NSGA II	0,1227	0,5023	0,0149	53,85%	2211

Baseado no mesmo critério do problema aerodinâmico 1 para aeronaves agrícolas, as soluções 1 e 2 da Figura 5.12b representam dois dos melhores projetos na frente de Pareto. Os aerofólios das soluções 1 e 2 são ilustrados na Figura 5.18.

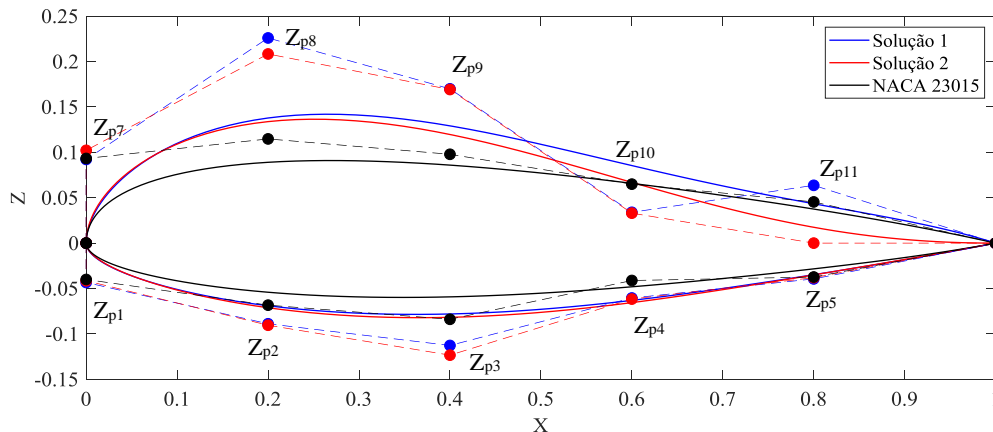


Figura 5.18: Aerofólios ótimos 1 e 2 do problema aerodinâmico de asas.

Na Figura 5.18, observa-se inicialmente que ambos os aerofólios solução apresentam uma curvatura maior, no intuito de gerar um aumento do C_L em relação à asa base. A análise do C_m , mostra que a solução 2 apresenta uma curvatura negativa em Z_{p11} , buscando minimizar o C_m .

5.3. Otimização de Grade Linear

Nesta seção, apresentam-se os resultados da aplicação de MO-CORS num problema de otimização multi-objetivo de uma grade linear de turbomáquina axial. Trata-se de um problema

de dois objetivos, tipo *min-min*, onde foram minimizados o coeficiente de perdas de energia ($f_1 = \bar{\omega}_v$) e outro objetivo relacionado ao ângulo de deflexão do escoamento (θ). Este problema seria originalmente mono-objetivo: minimização das perdas com restrição de ângulo de deflexão fixado. Essa restrição foi então transformada numa função objetivo adicional f_2 que determina o desvio percentual do ângulo de deflexão do escoamento em relação ao ângulo de deflexão do projeto base (θ_B), segundo a Equação 5.26.

$$f_2 = \frac{|\theta_B - \theta|}{\theta_B} 100\% \quad (5.26)$$

São consideradas como variáveis de projeto da grade: o ângulo de montagem da grade β , o inverso da razão de solidez $1/\sigma$ e o arqueamento do aerofólio γ . A geometria do aerofólio é definida na série NACA 65-(XX)10, sendo $\gamma = XX/10$.

5.3.1. Cálculo das funções para problema de grade linear.

O *solver* utilizado para o cálculo dos valores do coeficiente de perdas e do ângulo de deflexão do escoamento é tratado como *caixa preta*. Trata-se de um código computacional baseado em técnicas de interação viscosa/não viscosa. Emprega o método dos painéis (2D) de Hess e Smith com o desenvolvimento das camadas limites calculado por métodos integrais e as regiões de separação representadas por meio de um modelo semiempírico de injeção de vazão fictícia (RAMIREZ *et al*, 2001).

Análise Integral bidimensional da grade linear.

Grades lineares são retificações de cortes cilíndricos em máquinas de fluxo axiais. Análises em grades representam uma ferramenta fundamental no projeto de rotores. Muitos esforços têm sido empregados no sentido de alcançar projetos otimizados aplicados em aerodinâmica de rotores objetivando alcançar eficiências máximas, e assim controlar ou reduzir as perdas no escoamento. Um desenvolvimento teórico desta técnica foi elaborada por Reis (2010), para um problema de otimização robusta. Considera-se o escoamento a montante da pá com velocidade relativa W_1 e ângulo β_1 , e a jusante da grade com velocidade relativa W_2 e ângulo β_2 , conforme a Figura 5.19. Nessa figura, a velocidade W_2 , se apresenta menor que W_1 , o que caracteriza uma grade difusora ou geradora, típica de rotores de bombas, ventiladores e turbocompressores.

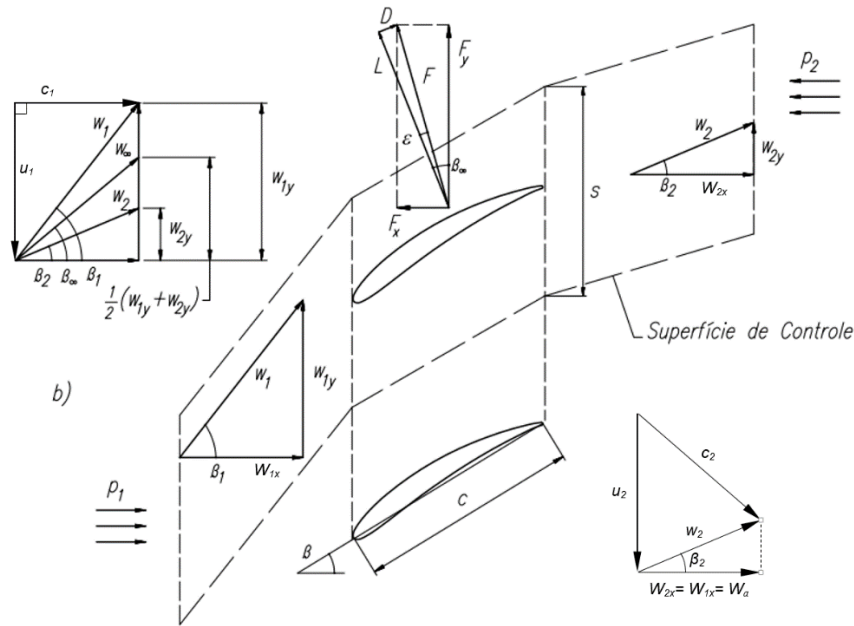


Figura 5.19: Forças e velocidades na grade linear.
Adaptado de: Schlichting e Scholz, (1951).

A força na direção x num canal de escoamento de comprimento vertical s (espaçamento entre pás), pode ser calculada pela diferença de pressões entre a entrada e a saída da grade $F_x = (p_2 - p_1)s$. Para escoamento incompressível, $(p_2 - p_1) = \frac{1}{2}\rho(W_1^2 - W_2^2)$ e $W_{1x} = W_{2x} = W_a$. Logo, a força F_x é calculada através da Equação 5.27, por unidade de largura da grade.

$$F_x = \rho s \frac{(W_{1y} + W_{2y})}{2} (W_{1y} - W_{2y}) \left[\frac{N}{m} \right] \quad (5.27)$$

Analogamente, a força na direção y pode ser calculada como: $F_y = \rho W_a s (W_{1y} - W_{2y})$. Das decomposições de velocidades da Figura 5.19, obtém-se: $(W_{1y} - W_{2y}) = (\tan \beta_1 - \tan \beta_2) W_a$. Sendo assim, a força na direção y , é calculada em função dos ângulos de entrada e saída do escoamento na grade através da Equação 5.28.

$$F_y = \rho s W_a^2 (\tan \beta_1 - \tan \beta_2) \quad (5.28)$$

Uma das grandezas relevantes em qualquer projeto de grades é o coeficiente de deflexão δ_y , razão entre a variação de velocidade tangencial e a velocidade axial:

$$\delta_y = \frac{(W_{1y} - W_{2y})}{W_a} = (\tan \beta_1 - \tan \beta_2) \quad (5.29)$$

Substituindo a Equação 5.29 na Equação 5.28, obtém-se:

$$F_y = \rho s \delta_y W_a^2 \quad (5.30)$$

Outro coeficiente importante no projeto de grades lineares móveis é o coeficiente de vazão Φ , razão entre a velocidade tangencial u e a velocidade axial:

$$\Phi = \frac{W_a}{u} \quad (5.31)$$

A Equação 5.31 se transforma na 5.32, considerando o caso sem pré-rotação do escoamento absoluto na entrada.

$$\Phi = \frac{1}{\tan \beta_1} \quad (5.32)$$

Conhecidas as componentes ideais de força nas direções x e y das Equações 5.27 e 5.30, é preciso introduzir agora as perdas de energia em termos de pressão, Δp_v .

O atrito do fluido com a superfície dos aerofólios da grade e os descolamentos de camada limite geram uma perda de energia, com parcela da energia total transformada em energia interna do fluido. Para fluidos incompressíveis, essa parcela é degradada na forma de uma diminuição da pressão total do fluido no escoamento:

$$\Delta p_v = (p_1 - p_2) + \frac{1}{2} \rho (W_1^2 - W_2^2) \quad (5.33)$$

Aplicando a diferença de quadrados e substituindo as Equações 5.27 e 5.30, obtém-se:

$$\Delta p_v = \frac{1}{s} (-F_x + F_y \tan \beta_\infty) \quad (5.34)$$

em que, $\tan \beta_\infty = \frac{1}{2} (\tan \beta_1 + \tan \beta_2)$.

O coeficiente de perdas $\bar{\omega}_v$ é a forma adimensional de representar as perdas de pressão, em relação à pressão dinâmica correspondente à velocidade relativa do escoamento na entrada da grade.

$$\bar{\omega}_v = \frac{\Delta p_v}{\frac{1}{2} \rho W_1^2} \quad (5.35)$$

A ligação entre os parâmetros geométricos da grade e o coeficiente de perdas da Equação 5.35 pode ser estabelecida através dos coeficientes de arrasto C_D e sustentação C_L . Da Figura 5.19 deduz-se que as forças de sustentação e arrasto são:

$$L = F_x \sin \beta_\infty + F_y \cos \beta_\infty \quad (5.36)$$

$$D = -F_x \cos \beta_\infty + F_y \sin \beta_\infty \quad (5.37)$$

Substituindo a Equação 5.34 na Equação 5.37, obtém-se:

$$D = \cos \beta_\infty (-F_x + F_y \tan \beta_\infty) = s \Delta p_v \cos \beta_\infty \quad (5.38)$$

Substituindo a Equação 5.35 na Equação 5.38

$$D = \frac{1}{2} \rho W_1^2 \bar{\omega}_v s \cos \beta_\infty \quad (5.39)$$

Sendo que por definição $C_D = D / \left(\frac{1}{2} \rho W_\infty^2 c \right)$.

$$C_D = \bar{\omega}_v \cos \beta_\infty \left(\frac{W_1}{W_\infty} \right)^2 \frac{s}{c} = \bar{\omega}_v \cos \beta_\infty \left(\frac{W_1}{W_a} \right)^2 \left(\frac{W_a}{W_\infty} \right)^2 \frac{s}{c} \quad (5.40)$$

$$C_D = \frac{\bar{\omega}_v \cos^3 \beta_\infty}{\sigma \cos^2 \beta_1} \quad (5.41)$$

Considerando as Equações 5.28, 5.34 e 5.35 na Equação 5.36 e aplicando a definição de coeficiente de sustentação $C_L = L / \left(\frac{1}{2} \rho W_\infty^2 c \right)$, obtém-se:

$$C_L = \frac{1}{\sigma} (2 \cos \beta_\infty \delta_y - \bar{\omega}_v \sin \beta_\infty \cos^2 \beta_\infty \cos^2 \beta_1) \quad (5.42)$$

Pelas correlações obtidas com base nos trabalhos de Speidel (1954) e Schlichting (1959), o coeficiente de perdas $\bar{\omega}'_v$ pode ser dado como uma função direta de grandezas da camada limite, através das Equações 5.43 e 5.44:

$$\bar{\omega}'_v = \frac{\sigma C_{d_1}^*}{\cos \beta_\infty} \quad (5.43)$$

em que $C_{d_1}^*$ é calculado através da expressão 5.42

$$C_{d_1}^* = 2\theta_{bf} \frac{\cos^2 \beta_1}{\cos^3 \beta_2} \cos \beta_\infty \quad (5.44)$$

onde θ_{bf} é a somatória das espessuras de quantidade de movimento nos lados de pressão e sucção no bordo de fuga. Note-se que as Equações 5.43 e 5.44 são válidas para camadas limites coladas. Entretanto, Speidel (1954), através de análises teóricas e experimentais, obteve uma correlação empírica para a determinação da espessura da quantidade movimento adicional devido a separação da camada-limite do lado de sucção do perfil aerodinâmico, θ_{sep} :

$$\theta_{sep} = \frac{1}{2} y_{tA} \left[\left(\frac{W_s}{W_2} \right) - 0,9 \right] \quad (5.45)$$

onde, W_s é a velocidade de separação, W_2 é a velocidade na saída da grade e y_{tA} é a somatória das distâncias da posição do ponto de separação na superfície do aerofólio até linha média de curvatura do lado de sucção, mais o de lado de pressão ($y_{tA} = y_{sucçãoA} + y_{pressãoA}$).

Finalmente o coeficiente de perdas de Speidel é calculado através da Equação 5.43, com um coeficiente de arrasto da Equação 5.44 acrescentado por efeitos de separação da camada limite.

$$C_{d_1} = 2(\theta_{bf} + \theta_{sep}) \frac{\cos^2 \beta_1}{\cos^3 \beta_2} \cos \beta_\infty \quad (5.46)$$

Assim, a Equação 5.47 representa a função objetivo f_1 para o problema de otimização, chamada coeficiente de perdas de Speidel adaptado para casos de separação de camada limite por Ramirez (2001).

$$\bar{\omega}_v^s = f_1 = \frac{\sigma C_{d1}}{\cos \beta_\infty} \quad (5.47)$$

5.3.2. Problema de otimização de grade linear.

Este problema de otimização consiste em se determinar a frente de Pareto bidimensional que se produz ao se minimizar o desvio percentual do ângulo de deflexão do escoamento (Equação 5.26) e o coeficiente de perdas de Speidel (Equação 5.47), devido à variação do ângulo de montagem β , do arqueamento do aerofólio γ e do inverso da solidez da grade $\lambda_s = 1/\sigma$, onde $\sigma = c/s$ – ou seja, com o vetor de decisão definido como: $\mathbf{X} = (\beta, \gamma, \lambda_s)$. Os limites laterais foram determinados com referência no projeto base, que corresponde à grade com um ângulo de escoamento na entrada $\beta_1 = 30^\circ$, um ângulo de ataque para máxima relação C_l/C_d , $\alpha_1 = 15,7^\circ$, solidez $\sigma = 1$ e arqueamento $\gamma = 0,8$ (aerofólio NACA 65-810). O valor do ângulo de montagem é obtido da diferença entre o ângulo de escoamento e o ângulo de ataque na entrada, $\beta = \beta_1 - \alpha_1$. O ponto de projeto base do problema de otimização fica então definido como: $\mathbf{X} = (14,3^\circ, 0,8, 1,0)$. Os valores dos limites laterais da Tabela 5.10 foram determinados buscando evitar regiões de descolamento total da camada limite no aerofólio (estol). As curvas características que incluem o projeto base foram obtidas do relatório de Emery *et al.* (1958) e são apresentadas na Figura 5.20.

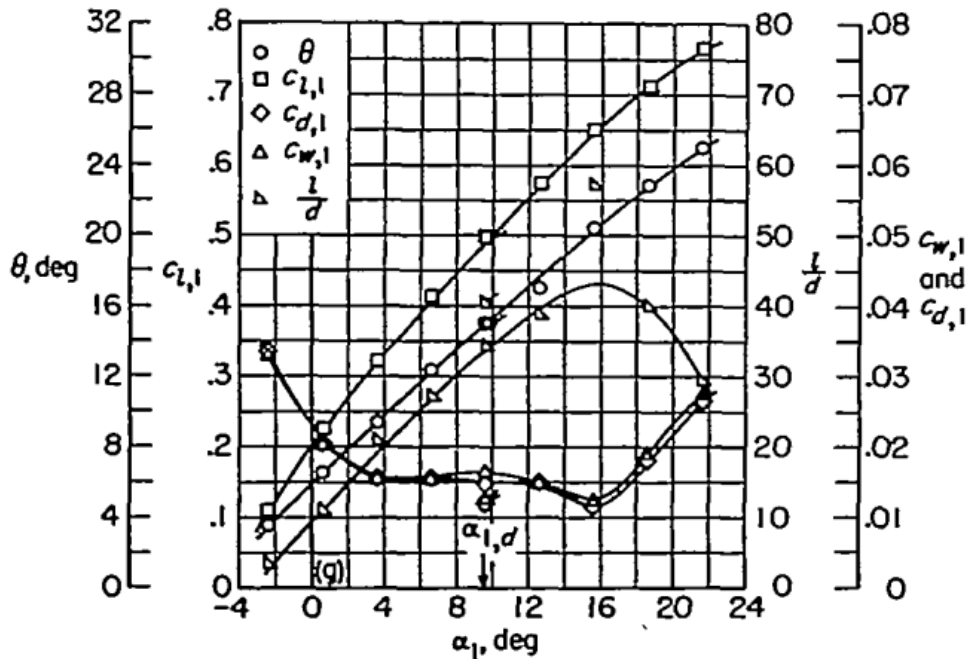


Figura 5.20: Curvas características que incluem o projeto base da grade.

Fonte: Emery *et al.* (1958) - NACA Report 1368.

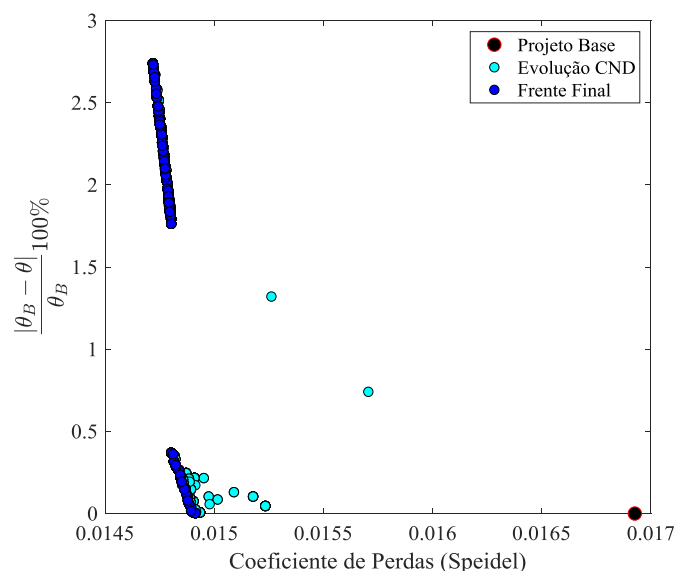
Tabela 5.10: Valores da grade base e limites laterais.

X	x_{LI}	Asa Base	x_{LS}
β	12,80°	14,30°	15,80°
γ	0,70	0,80	0,95
λ_s	0,90	1,00	1,12
Aerofólio	N/A	NACA 65810	N/A

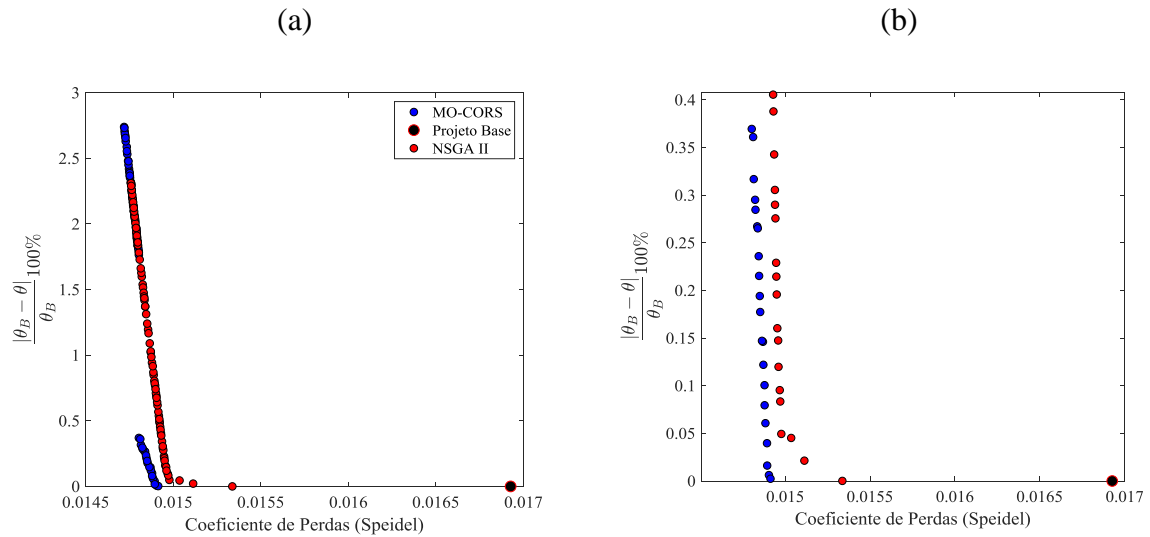
5.3.3. Resultados do problema de otimização da grade linear.

Todas as considerações feitas no preâmbulo da seção 5.2.3 a respeito da forma de apresentação de resultados (gráficos e tabela) e parâmetros ou critérios adotados para aplicação dos algoritmos MO-CORS e NSGA II aos problemas de asa, valem igualmente nesta seção no escopo da aplicação ao problema de grade linear.

Embora as soluções de Pareto ilustradas nas figuras seguintes não apresentem o ponto base como solução dos algoritmos MO-CORS e NSGA II, é sabido que ele está contido na frente de Pareto do problema, devido à natureza da função f_2 . A Figura 5.21 ilustra a evolução do algoritmo MO-CORS num total de 236 avaliações da função objetivo. Depois de 5 iterações, o algoritmo MO-CORS consegue atingir soluções com coeficientes de perdas menores que o do projeto base com um desvio percentual da deflexão do escoamento abaixo de 0,5% em relação ao projeto base.

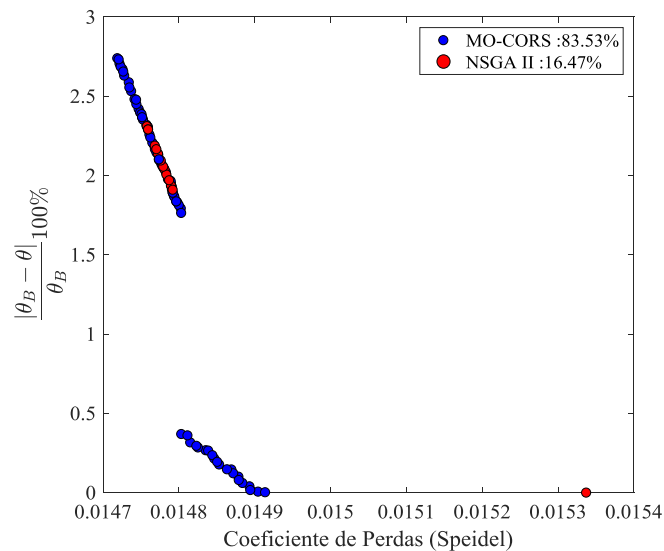
**Figura 5.21:** Evolução dos conjuntos não dominados MO-CORS (Problema da grade).

Pelos resultados obtidos pelo algoritmo MO-CORS, a Figura 5.21 também sugere que a frente de Pareto do problema seja descontínua.



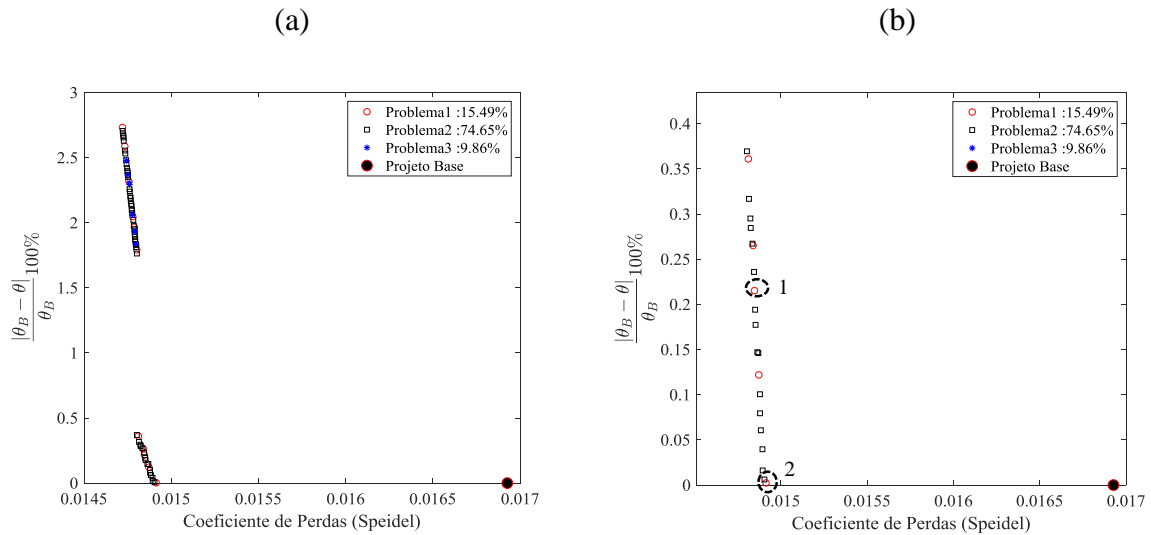
Figuras 5.22: Frontes de Pareto do Problema da Grade (a) Final (b) Ampliação.

Na Figura 5.22(a) mostra-se que o algoritmo NSGA II não consegue atingir a possível descontinuidade da frente de Pareto, além de não conseguir dominância de conjunto sobre a solução de MO-CORS para valores de f_2 menores que 0.40, como se observa na Figura 5.22(b). Seria interessante realizar estudos posteriores para avaliar a viabilidade e robustez de certas soluções da frente de Pareto atingidas apenas pelo MO-CORS.



Figuras 5.23: Frente das frentes entre MO-CORS e NSGA II (Problema da grade).

Neste problema, o algoritmo MO-CORS contribui com 83,53% dos pontos na frente das frentes contra 16,47% de NSGA II. Para atingir a convergência, o algoritmo MO-CORS precisou de 236 chamadas da função objetivo em um tempo total de 31,47 min, enquanto o NSGA II precisou de 1920 chamadas num tempo total de 70,21 min. Logo, o MO-CORS acelerou ao menos 8,13 vezes o processo de otimização, em relação ao número de avaliações e 2,23 vezes em tempo computacional.



Figuras 5.24: Contribuição dos problemas auxiliares de MO-CORS (Problema da grade). (a) Final (b) Ampliação.

A Figura 5.24a mostra neste caso um desequilíbrio muito acentuado no aporte de pontos na frente de Pareto por parte de cada problema auxiliar. Observa-se agora que a imensa maioria dos pontos na frente de Pareto são colocados pelo problema auxiliar 2 (74,65%), enquanto os problemas auxiliares 1 e 3 contribuem com 15,49% e 9,86%, respectivamente. Embora o problema auxiliar 1 coloque apenas 15,49% dos 79 pontos na frente, estes se concentram nos extremos das regiões descontínuas. Na Figura 5.24(b), observa-se que, para valores de $f_2 \leq 0,4\%$, dos 21 projetos da frente, 05 pontos uniformemente dispersos são colocados pelo problema auxiliar 1 e 16 pelo problema auxiliar 2. O primeiro ponto a aparecer nessa região do espaço objetivo, indicado como (1) na Figura 5.24(b), foi colocado pelo problema auxiliar 1. Por outro lado, o projeto solução (2) (também oriundo do problema 1) talvez seja o mais interessante de todos, já que apresenta 12% a menos de perdas e apenas 0,0024% de desvio do ângulo de deflexão em relação ao projeto base. O que parece ocorrer aqui é o seguinte: quando o problema auxiliar 1 consegue convergir para um ponto não dominado com imagem em uma região extrema ou inexplorada do espaço objetivo (efeito CORS), isso promove em seguida a ação do problema 2 em preencher os vazios (efeito *gap*).

Os valores das métricas de desempenho para o problema de otimização multi-objetivo da grade são mostrados na Tabela 5.11. Os valores das variáveis de projeto da solução (2) são mostrados na Tabela 5.12.

Tabela 5.11: Valores das métricas de desempenho para MO-CORS no problema da grade.

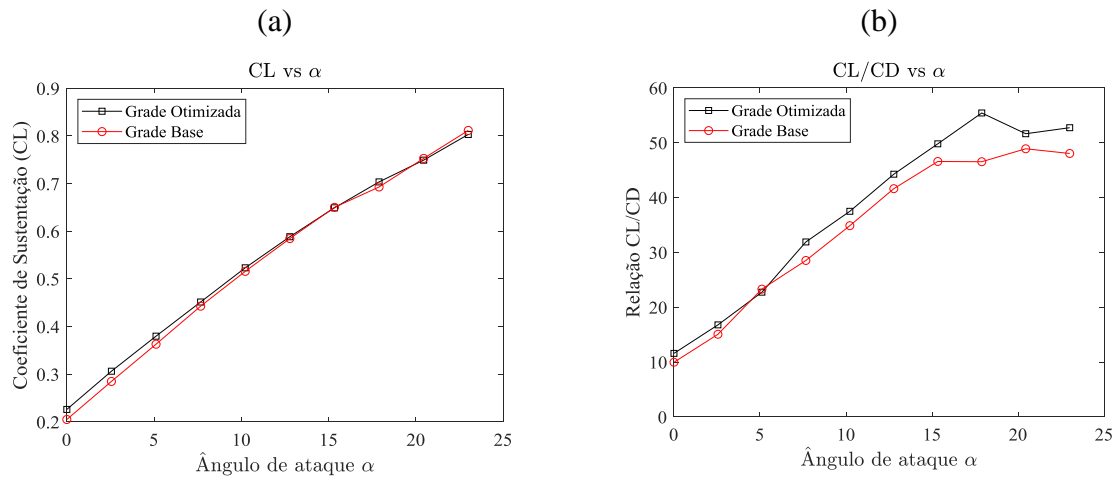
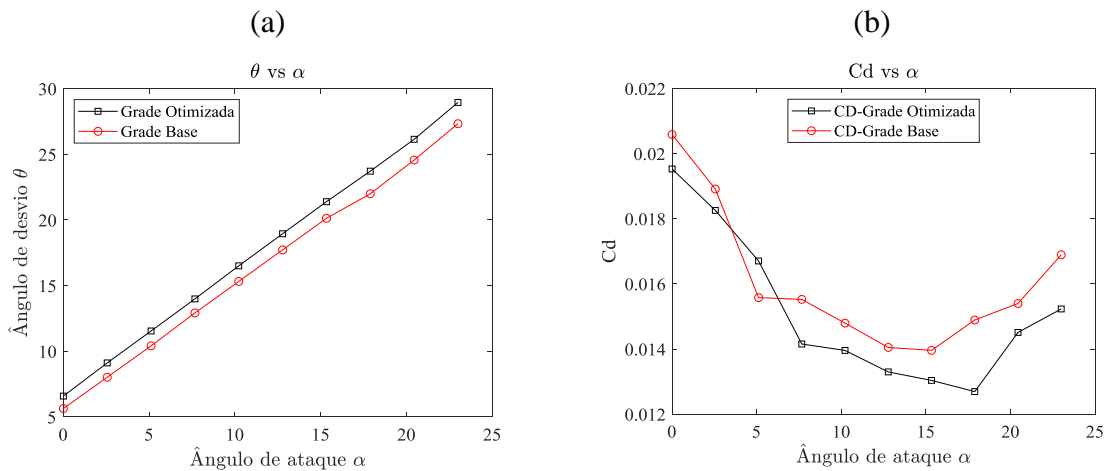
Algoritmo	Métricas de desempenho (Capítulo 2)				N° de avaliações
	Espalhamento	Espaçamento	HRM	MVFP	
MO-CORS	0,4500	0,0081	0,3051	83,53%	236
NSGA II	0,1474	0,0041	0,0417	16,47%	1920

Tabela 5.12: Valores das variáveis de projeto da solução 2 no problema da grade.

Solução	Variáveis de decisão			Vetor de objetivos $f_i = (f_1 f_2)$
	$\beta(^{\circ})$	$1/\sigma$	γ em % de c	
2	15,7503	0,9514	0,9495	(0,0149 0,0024)
Base	14,3000	1,0000	0,8000	(0,0169 0,0000)

5.3.4. Curvas aerodinâmicas das grades ótima e base.

É importante estabelecer uma comparação entre as curvas aerodinâmicas da grade base e a grade otimizada. Estas curvas representam a variação de: C_L (i), (ii) C_L/C_D , (iii) θ e (iv) $C_{di}(C_W)$ e C_D , versus o ângulo de ataque α ou ângulo do escoamento relativo na entrada em relação à linha de corda do aerofólio.

**Figuras 5.25:** Curvas aerodinâmicas da grade. (a) Curva C_L vs α . (b) Curva C_L/C_D vs α .**Figuras 5.26:** Curvas aerodinâmicas da grade. (a) Curva θ vs α . (b) Curva C_D vs α .

Observa-se na Figura 5.25(a) que os coeficientes de sustentação das grades base e otimizada são praticamente os mesmos na faixa entre seus ângulos de ataque de projeto na entrada, $\alpha_{1p} = 14,25^\circ$ para a otimizada e $\alpha_{1p} = 15,7^\circ$ para a base. Lembre-se que o ângulo do escoamento na entrada foi mantido em $\beta_1 = 30^\circ$ e que $\alpha_1 = \beta_1 - \beta$. A relação de C_L/C_D da grade otimizada resulta superior à da base em uma ampla faixa de α_1 (Figura 5.25(b)). Para toda faixa de α_1 analisada, o ângulo de deflexão θ da grade otimizada é superior ao da grade base (Figura 5.26(a)). Mas é importante notar que, em seus respectivos pontos de projeto, o valor de θ é praticamente o mesmo em função da intensa minimização do objetivo f_2 . Observa-se na Figura 5.25(b) que o coeficiente de arrasto da grade otimizada foi inferior ao da base em praticamente toda a faixa analisada.

5.3.5. Curvas características operativas das grade ótima e base.

A eficiência hidráulica de uma grade geradora apresentada no diagrama da Figura 5.27 pode ser expressa como a razão entre o trabalho específico total da grade e o trabalho específico da pá.

$$\eta_h = \frac{Y}{Y_{pá}} = \frac{Y_{pá} - Y_p}{Y_{pá}} = 1 - \frac{Y_p}{Y_{pá}} \quad (5.48)$$

sendo $Y_{pá} = u(C_{u2} - C_{u1})$, em que u é a velocidade tangencial; C_{u2} e C_{u1} são os componentes circunferenciais da velocidade absoluta na saída e na entrada, respectivamente; Y_p representa as perdas de energia do escoamento. Considera-se o caso de escoamento sem pré-rotação na entrada ($C_{u1} = 0$). Usando os triângulos de velocidades na definição do trabalho específico as formulações da seção 5.3.1, é possível mostrar que:

$$\eta_h = 1 - \frac{\bar{\omega}_v}{\sin^2 \beta_1 (1 - \Phi \tan \beta_2)} \quad (5.49)$$

em que Φ é o coeficiente de vazão, Equações 5.31 e 5.32.

Define-se o coeficiente de pressão como:

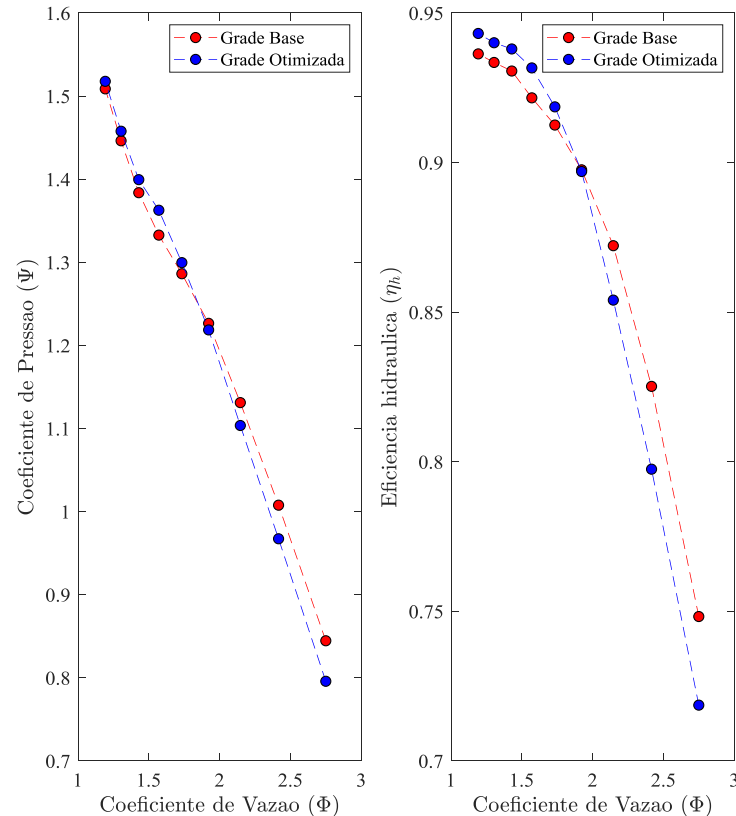
$$\Psi = \frac{2Y}{u^2} \quad (5.50)$$

Considerando a definição de rendimento hidráulico em (Equação 5.48) e os triângulos de velocidade, resulta que

$$\Psi = 2\eta_h(1 - \Phi \tan \beta_2) \quad (5.51)$$

As curvas de desempenho operacional das grades lineares representam a variação do coeficiente de pressão Ψ e da eficiência η_h em função do coeficiente de vazão Φ . Na Figura 5.27 ilustram-se as curvas de desempenho operacional da grade base e da grade otimizada.

Verifica-se uma pequena diferença de eficiência de 0,6% entre a grade otimizada (0,9186) e a grade base (0,9125) no ponto de projeto. Em geral, para coeficientes de vazão entre 1,19 e 1,73 (vazão de projeto), a eficiência da grade otimizada foi maior que a da grade base. Para valores maiores que 2, a grade base foi mais eficiente. Verifica-se também uma diferença muito pequena no valor do coeficiente de pressão no ponto de projeto: 1,301 para a grade otimizada e 1,286 para a grade base. Logo, a grade otimizada reproduz praticamente o mesmo ponto de operação nominal da grade base, mas com uma ligeira melhoria de eficiência nesse ponto e na faixa de menores vazões.



Figuras 5.27: Curvas de desempenho operacional das grades.

5.4. Tempo Computacional Próprio do MO-CORS

O teste de desempenho da maioria dos AOAM se baseia no número de avaliações das funções objetivo. A razão é que, assim como MO-CORS, eles são projetados com o intuito de acelerar processos de otimização que envolvem funções objetivo computacionalmente custosas de fato. Ou seja, funções cujo tempo computacional de cada avaliação, TF , seja significativamente alto em comparação com o tempo utilizado pelo próprio algoritmo. Esse *tempo próprio*, TP , exclui o tempo computacional empregado para fazer as avaliações das funções objetivo, mas envolve todos os processos inerentes ao algoritmo, tais como: construção do banco de dados, construção dos metamodelos, normalização, determinação dos conjuntos não dominados, resolução dos problemas de otimização auxiliar, dentre outros. Sendo assim, pode-se dizer que TP não é fixo para cada algoritmo e que ele vai depender de muitos fatores, dentre os quais destacam-se o número de variáveis de projeto e o número de objetivos.

Nesta pesquisa, a avaliação das funções objetivo de todas as aplicações escolhidas foi relativamente barata em tempo computacional. Desse modo, tornou-se exequível testar o algoritmo numa certa diversidade de problemas aerodinâmicos reais e obter resultados comparativos em relação à aplicação direta do algoritmo NSGA II. Mesmo assim, é interessante ter-se uma medida do tempo computacional próprio de MO-CORS, para cada aplicação feita, conforme expresso na Equação 5.52.

$$TP = TT - N_{eval} \times TF \quad (5.52)$$

Nessa equação, TT é o tempo total do processo de otimização e N_{eval} é o número de avaliações das funções objetivo (ou melhor, o número de chamadas do *solver*). Os resultados comparativos do tempo computacional próprio entre os algoritmos MO-CORS e NSGA II, para as três aplicações multi-objetivo, são mostrados na Tabela 5.13.

Tabela 5.13: Tempo próprio dos algoritmos MO-CORS e NSGA II.

Aplicação	$TF(s)$	NSGA II			MO-CORS		
		$TT(s)$	N_{eval}	$TP(s)$	$TT(s)$	N_{eval}	$TP(s)$
1	2,0366	7680	2400	2792,2	5880	427	5010,4
2	2,0537	16200	2211	11659,3	7920	402	7094,4
3	0,4005	4200	1920	3431,0	1860	236	1765,5

Aplicação 1: Problema da asa 1. Variação dos parâmetros da asa.

Aplicação 2: Problema da asa 2. Variação da forma do aerofólio.

Aplicação 3: Problema da grade linear

Capítulo 6

Conclusões e Trabalhos Futuros

6.1 Conclusões.

6.1.1. Conclusões gerais sobre o desempenho do algoritmo MO-CORS.

Neste trabalho, foi desenvolvido o algoritmo MO-CORS, para assistência de processos de otimização multi-objetivo envolvendo funções computacionalmente custosas. Trata-se de uma extensão do algoritmo de assistência mono-objetivo CORS, baseado em técnicas de construção de metamodelos através de funções de base radial (RBF). Analisando os resultados obtidos pode-se concluir que, o algoritmo MO-CORS demonstrou eficácia na resolução de problemas mono-objetivo e multi-objetivo, obtendo-se acelerações computacionais de 4 a 8 vezes em aplicações de problemas de aerodinâmica. Sendo assim, MO-CORS pode se tornar uma ferramenta de auxílio muito eficaz para processos de otimização que exijam recursos computacionais custosos.

Embora o algoritmo permite resolver, em teoria, problemas com qualquer quantidade de variáveis e qualquer número de objetivos, seu desempenho foi analisado nesta pesquisa unicamente com até 12 variáveis de projeto e 3 funções objetivo, prevendo que o cálculo contínuo de distancias no espaço de decisão discretizado precisado pelo CORS (Base desta pesquisa), poderia contribuir num incremento elevado do custo computacional próprio do algoritmo MO-CORS.

A versatilidade do MO-CORS em tratar tanto problemas de otimização mono-objetivo como multi-objetivo é um dos aspectos mais importantes do desenvolvimento de algoritmos de otimização assistidos por técnicas de metamodelagem embora pouco abordado na literatura da

área. A estratégia empregada no MO-CORS é baseada na resolução iterativa de três (03) problemas auxiliares de otimização de metamodelos. Esta estratégia se centrou primeiramente na intensificação da capacidade de exploração do algoritmo na procura dos extremos da frente de Pareto em problemas multi-objetivo, através da resolução do problema auxiliar 1 (“efeito CORS”) que garante também, no caso de problemas mono-objetivo, a obtenção de ótimos globais. Em segundo lugar, depois de se atingir pontos mais extremos (ou mais isolados) da frente de cada iteração, esses geram lacunas na mesma, as quais passam a ser abordadas por pontos resultantes da resolução dos problemas auxiliares 2 e 3 (“gap” e “vetor alvo”). Os resultados da contribuição de cada problema auxiliar na frente final mostram que MO-CORS é uma ferramenta versátil na assistência dos problemas de otimização multi-objetivo pretendidos. Além disso, garante a obtenção de ótimos globais nos problemas mono-objetivo (pela introdução do algoritmo CORS), conforme verificado em funções de teste e em uma aplicação em pás de turbinas eólicas. Por outro lado, além de diversidade nas soluções na frente de Pareto, MO-CORS proporcionou maior taxa de sucesso e convergência mais rápida em relação ao algoritmo NSGA II, tanto nos problemas de teste como em aplicações de aerodinâmica multi-objetivo.

6.1.2. Conclusões sobre o desempenho do algoritmo MO-CORS em funções de teste mono-objetivo.

O algoritmo MO-CORS respondeu eficientemente na procura do mínimo global das funções de teste Dixon-Szegö. Para todas as funções testadas, os mínimos foram encontrados com 100% de sucesso. A estrutura do algoritmo permitiu melhorar o processo de otimização, reduzir o número de chamadas da função objetivo e melhorar o desempenho do metamodelo.

Em relação ao algoritmo CRSI, o uso do algoritmo NSGA II na resolução do problema de otimização do metamodelo no caso mono-objetivo aporta um melhor desempenho ao algoritmo MO-CORS para as funções de teste analisadas quando a RBF foi a multiquádrica inversa, produzindo sempre acelerações maiores. Um aspecto importante verificado nos problemas mono-objetivo é a maior eficiência que as RBFs multiquádricas podem apresentar em relação às TPS-pol, desde que o parâmetro de forma seja adequado (DIAZ *et al*, 2018.b). Sendo assim, foi inserido dentro do algoritmo o uso da técnica eficiente de Rippa de aplicação de LOOCV para a otimização do parâmetro de forma das RBF multiquádricas. A aplicação dessa técnica com a RBF inversa multiquádrica, não afeta substancialmente o desempenho do MO-CORS. Constitui, porém, uma ferramenta de automatização da seleção de um parâmetro de forma adequado para a construção do metamodelo em cada iteração. Esta implementação aporta versatilidade ao algoritmo MO-CORS para futuros testes e aplicações com RBF Gaussiana, multiquádrica e inversa multiquádrica.

6.1.3. Conclusões do desempenho do algoritmo MO-CORS em problemas de teste multi-objetivo.

O algoritmo MO-CORS respondeu eficientemente na procura da frente de Pareto de problemas de teste (MOPs e ZDTs) envolvendo até 12 variáveis de decisão e até três (3) funções objetivo: atingiu na maioria dos casos uma boa quantidade de pontos na frente, com boa diversidade, taxa de sucesso superior a 0,3, dominância dos pontos na frente das frentes maior que 45% e acelerações entre 2,5 e 20 vezes, em relação ao NSGA II. Em geral, observou-se, nas frentes de Pareto finais, que os pontos oriundos da resolução do problema auxiliar 1 (“efeito” CORS) tendem a se aglomerar nas regiões mais extremas e isoladas da frente. Essa estratégia de busca global potencializa a eficácia das técnicas de preenchimento dos espaços com baixa aglomeração, referentes à resolução dos problemas de otimização auxiliar 2 e 3. Desse modo, o algoritmo MO-CORS mostrou-se eficiente na procura dos mais diversos tipos de frentes, em particular as frentes descontínuas.

6.1.4. Conclusões sobre o desempenho do algoritmo MO-CORS em problemas de aerodinâmica.

Em geral as aplicações foram feitas, no intuito de testar a qualidade da frente obtida pelo MO-CORS em problemas reais de engenharia e não de medir o tempo computacional empregado pelo mesmo, na resolução destes problemas. Sendo assim o tempo computacional empregado por cada código utilizado para resolver as metodologias de projeto (funções objetivo) em cada aplicação, é considerado baixo. Mesmo assim o MO-CORS consegue acelerar desde o ponto de vista de tempo computacional, todas as aplicações testadas

A) Otimização aerodinâmica de pás de turbinas eólicas de eixo horizontal baseado na teoria de elemento de pá.

A teoria de elemento de pá baseada na solução das equações de quantidade de movimento linear e angular do fluxo de ar (BEM) constitui uma ferramenta prática e bem estabelecida para previsão do desempenho de turbinas eólicas de eixo horizontal. Ela serviu de base à metodologia desenvolvida neste trabalho para projeto otimizado mono-objetivo da geometria da pá (maximização do coeficiente de potência). A aplicação do MO-CORS possibilitou obter geometrias preliminares de pá que melhoram o coeficiente de potência de uma turbina base em relação ao obtido pelo uso direto do NSGA II e com um custo computacional 4 vezes menor. As curvas de desempenho previstas com o modelo indicam um desempenho superior da turbina otimizada em relação à base em uma ampla faixa de razões de velocidade de ponta (Figura. 5.6). Além das aproximações da própria teoria BEM, o modelo utiliza coeficientes aerodinâmicos provenientes de um método potencial com acoplamento da camada limite.

Apesar dessas limitações, as curvas são similares às curvas experimentais indicadas na literatura. Desse modo, as melhorias previstas pela metodologia podem ser consideradas significativas.

B) Otimização aerodinâmica de asas.

Dois problemas de otimização multi-objetivo de asas, ambos de minimização do inverso do coeficiente de sustentação e do módulo do coeficiente de momento, foram tratados neste trabalho, visando comparar o algoritmo MO-CORS com o algoritmo NSGA II aplicado diretamente.

No problema 1, de otimização dos parâmetros geométricos da asa, verificou-se que a aplicação do algoritmo MO-CORS consegue atingir pontos que dominam o projeto base já nas primeiras 10 iterações. O MO-CORS consegue dispersar melhor os pontos de sua solução (frente) neste problema, com um claro domínio sobre a solução obtida pelo NSGA II. Na comparação direta entre as frentes de cada algoritmo, o MO-CORS contribuiu com quase a totalidade dos pontos na frente das frentes. Além disso, conseguiu uma aceleração do processo computacional de 5,62 vezes em relação ao NSGA II. A estratégia de seleção de pontos dos problemas auxiliares resultou equilibrada e eficiente, observando-se uma contribuição mais o menos significativa de cada problema auxiliar.

No problema 2, de otimização do formato do aerofólio da asa, verificou-se que a aplicação do algoritmo MO-CORS conseguiu atingir pontos que dominam o projeto base depois de 40 iterações. O MO-CORS consegue dispersar melhor as soluções finais neste problema com uma aceleração computacional de 5,5 vezes em relação ao NSGA II. Nesse caso, porém, as soluções diretas do NSGA II contribuíram com 54% dos pontos na frente das frentes. Por outro lado, dentro da estratégia de seleção de pontos, o problema auxiliar 1 exerceu uma grande influência na formação da frente final do algoritmo MO-CORS, colocando quase 60% dos pontos.

B) Otimização aerodinâmica de uma grade linear de turbomáquinas

No problema de otimização aerodinâmico de uma grade linear, de minimização do coeficiente de perdas de Speidel e do desvio percentual do ângulo de deflexão com respeito ao projeto base, o algoritmo MO-CORS obteve um alto desempenho em comparação com o algoritmo NSGA II. MO-CORS encontrou soluções que dominaram o projeto base já nas iterações iniciais. Ao final dos processos, verificou-se que o MO-CORS conseguiu uma aceleração computacional de 8,13 vezes em relação ao NSGA II, atingindo inclusive pontos não encontrados pelo algoritmo NSGA II. Embora nesse caso o problema auxiliar 1 tenha colocado relativamente poucos pontos na frente final, estes se posicionaram nas proximidades dos extremos das faixas descontínuas da frente capturadas pelo MO-CORS, mas não pelo

NSGA II. Esses pontos, no entanto, promoveram a atuação eficaz dos outros dois problemas (principalmente o problema 2) no preenchimento de espaços de baixa aglomeração.

Nas curvas aerodinâmicas da grade observa-se que a solução encontrada com o menor ângulo de desvio e menor coeficiente de perdas e a base, apresentam praticamente o mesmo coeficiente de sustentação para cada um de seus ângulos de ataque de projeto. Entanto a relação C_L/C_D da grade otimizada resulta superior à da base numa ampla faixa de ângulos de ataque, isto devido à diminuição do coeficiente de arrasto na mesma faixa, mantendo uma diferença máxima de $1,4^\circ$ no ângulo de deflexão do escoamento θ . A diminuição do coeficiente de arrasto e a mínima diferença no ângulo de deflexão foi devida ao próprio problema de otimização aplicado, onde foram minimizados o coeficiente de perdas e o ângulo de deflexão

Em relação à operabilidade da grade otimizada baseada na variação do coeficiente de vazão, a máxima eficiência hidráulica da grade otimizada foi maior que a da grade base em boa parte das curvas de operação, com uma diferença na ordem de 1%, na faixa de baixos coeficientes de vazão até a vazão do projeto base ($\Phi = 1,732$). Esta pequena variação é devida a que os coeficientes aerodinâmicos alteram muito pouco seus valores com a variação da curvatura do aerofólio.

6.2 Trabalhos Futuros.

Em problemas multi-objetivo, o algoritmo MO-CORS incorpora a técnica do CORS de exploração global do espaço de decisão em geral – porém com ênfase mais específica na seleção de pontos próximos das extremidades da frente – em conjunto com métodos de exploração e seleção adaptados de outros algoritmos AOAM propostos na literatura base. Isso no intuito de acelerar eficientemente os processos de otimização mono-objetivo e multi-objetivo de problemas que envolvam funções computacionalmente custosas, do tipo *caixa preta*. Em princípio, o MO-CORS tem potencial para ser aplicado em qualquer problema de otimização (não só de aerodinâmica ou máquinas de fluxo) em que o cálculo das funções seja bem mais caro do que aqueles empregados nas aplicações desta pesquisa sempre levando em conta as limitantes quanto ao número de variáveis de projeto e número de objetivos para os quais MO-CORS já foi testado. Busca-se assim maior nível de fidelidade dos projetos otimizados através, por exemplo, de simulações com CFD. Sendo assim, são sugeridos os seguintes trabalhos de aplicação do algoritmo:

- 1) Aplicação do MO-CORS em problemas de otimização de aeronaves e turbomáquinas baseados em CFD.

- 2) Aplicação do MO-CORS em projetos de otimização robusta de projetos de engenharia mecânica e aeroespacial.

Devido a versatilidade do algoritmo, este pode ser utilizado para a realização de pesquisas orientadas ao aprimoramento do mesmo em alguns aspectos, tais como: a própria aceleração computacional, o nível de fidelidade dos resultados e a capacidade de trabalho com um grande número de variáveis. Nesse sentido, seguem as seguintes sugestões:

- 1) Estudo comparativo do uso de técnicas de Kriging e Funções de Base Radial para a construção de metamodelos dentro do contexto do desempenho de MO-CORS mono-objetivo e multi-objetivo.
- 2) Desenvolvimento de uma técnica computacionalmente eficiente para a exploração global do espaço de decisão dentro do contexto de MO-CORS.
- 3) Estudo comparativo da influência de diferentes algoritmos de otimização mono e multi-objetivo no desempenho de MO-CORS, além do NSGA II.
- 4) Introdução de técnicas de otimização paralela dentro do contexto de MO-CORS.
- 5) Desenvolvimento de um algoritmo de otimização multi-fidelidade baseado na estrutura do MO-CORS.
- 6) Criação de versões do MO-CORS em linguagens de programação compiladas de alto desempenho (p. ex., Fortran)

Outras possíveis aplicações no contexto de projetos avançados de engenharia são:

- 1) Otimização multi-disciplinar de projetos de estruturas aeronáuticas e superfícies aerodinâmicas tipo *Morphing*.

Referências Bibliográficas

- Akhtar e Shoemaker(2016)** Akhtar Taimoor e Christine Shomaker. Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection. *Journal of Global Optimization*, 64(1):17–32.
- Arispe et al(2018)** Tania M Arispe, Waldir de Oliveira e Ramiro G Ramirez. Francis turbine draft tube parameterization and analysis of performance characteristics using CFD techniques, *Renewable Energy*, 127(1):114-124.
- Bangga et al(2018)** Galih Bangga, Giorgia Guma, Thorsten Lutz e Ewald Krämer. Numerical simulations of a large offshore wind turbine exposed to turbulent inflow conditions, *Wind Engineering*, 42(2):88-96.
- Bartoli et al(2016)** N Bartoli, I Kurek, R Lafage, T Lefebvre, R Priem, MA Bouhlel, J Morlier, V Stiliz, R Regis. Improvement of efficient global optimization with mixture of experts: Methodology developments and preliminary results in aircraft wing design. Em *17th Multidisciplinary Analysis and Optimization Conference*, AIAA/ISSMO.
- Botan(2019)** Antônio Carlos Botan Barkett. *Otimização de um modelo de turbina hidráulica tipo bulbo aplicada em condições de queda ultrabaixa*, Tese de Doutorado, Universidade Federal de Itajubá.
- Bravo et al(2017)** Pedro David Bravo Mosquera, Laura Botero Bolivar, Daniel Acevedo Giraldo e Hernán Darío Cerón Muñoz. Aerodynamic design analysis of a UAV for superficial research of volcanic environments, *Aerospace Science and Technology*, 70(1):600-614.
- Bravo et al(2018)** Pedro David Bravo Mosquera, Hernán Darío Cerón Muñoz, Guillermo Díaz Vásquez e Fernando Martini Catalano. Conceptual design and CFD analysis of a new prototype of agricultural aircraft, *Aerospace Science and Technology*, 80(1):156-176.
- Caixeta(2017)** Caixeta Júnior Paulo Roberto. *Otimização Multidisciplinar em Projeto de Asas Flexíveis utilizando Metamodelos*. Tese de Doutorado, Universidade de São Paulo.

- Castilho(2013)** Lorena Castilho. *Projeto otimizado de grades lineares para aplicação em turbomáquinas*. Dissertação de Maestrado, Universidade Federal de Itajubá.
- Chen(2014)** Xiaomin Chen. *Optimization of Wind Turbine Airfoils / Blades and Wind Farm Layouts*, PhD Dissertation, Washington University.
- Coello(2011)** Carlos A Coello Coello. Evolutionary multiobjective optimization. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge*, 1(5):444-447.
- Coello et al(2007)** Carlos A Coello Coello, G Lamont e D Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*, volume 5, Springer.
- Conn et al(2009)** Andrew Conn, Katya Scheinberg e Luis N Vicente. *Introduction to derivative-free optimization*, MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics and Mathematical Programming, Philadelphia, USA.
- Cosentino e Holst(1986)** Gary B Cosentino e Terry L Holst. Numerical optimization design of advanced transonic wing configurations, *Journal of aircraft*, 23(3):192–199.
- Datta e Regis(2016)** Rituparna Datta e Rommel G Regis. A surrogate-assisted evolution strategy for constrained multi-objective optimization. *Expert Systems with Applications*, 57: 270–284.
- Deb(2001)** Kalyamoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*, volume 5, John Wiley & Sons.
- Deb e Jain(2002)** Kalyamoy Deb e Sachin Jain. Running Performance Metrics for Evolutionary Multi-Objective Optimization. Em *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL 02)*.
- Deb et al(2002)** Kalyamoy Deb, Amrit Pratap, Sameer Agarwal e Tamt Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6(2):182 – 197.
- Diaz(2015)** Nelson Jose Diaz Gautier. *Análisis aerodinámico de alas subsónicas basado en el método del vortex lattice (VLM) mediante un código computacional*. Tesis de Maestría, Universidad de Carabobo.
- Diaz et al(2017)** Nelson Diaz, Nelson Manzanares Filho e Bruno de Sousa. Metodologia para diseño aerodinamico de alas basado en tecnicas de optimizacion multiobjetivo. Em *XIII Congreso Iberoamericano de Ingeniería Mecânica*, (CIBIM-2017).

- Diaz et al(2017)** Nelson Diaz, Nelson Manzanares Filho e Edna da Silva Ramirez. **Global Optimization Based on Metamodels Using Radial Basis Function With Application to Aircraft Wing.** Em *24th ABCM International Congress of Mechanical Engineering, (COBEM-2017)*.
- Diaz et al(2018.a)** Nelson Diaz, Nelson Manzanares Filho e Edna da Silva Ramirez. Comparison and Application of Different Techniques of Multi-objective Optimization for Aerodynamics Wing Design. Em *31st congress of international council of the aeronautical Sciences, (ICAS-2018)*.
- Diaz et al(2018.b)** Nelson Diaz, Nelson Manzanares Filho e Edna da Silva Ramirez. Radial Basis Functions Influence in CORS Methodology Applied on Aerodynamic Wing Optimization Problems, *International Conference on Engineering Optimization*. Springer
- Dixon e Szëgo(1978)** LCW Dixon e GP Szëgo. The optimization problem: An introduction, *Towards Global Optimization II*.
- Drela(1989)** Mark Drela. XFOIL: An analysis and design system for low Reynolds number airfoils, *Springer*, p.1-12.
- Duan e Zhao(2010)** Wei Duan e Feng Zhao. Loading analysis and strength calculation of wind turbine blade based on blade element momentum theory and finite element method. Em *2010 Asia-Pacific Power and Energy Engineering Conference, IEEE*.
- Eberhart e Kennedy(1995)** Russell Eberhart e James Kennedy. A new optimizer using particle swarm theory. Em *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE*.
- Ehrgott(2006)** Matthias Ehrgott. A discussion of scalarization techniques for multiple objective integer programming, *Annals of Operations Research*, 147(1): 343-360.
- Falkenauer(1999)** E Falkenauer. Genetic algorithms and grouping problems. 1998, volume 1, John Wiley & Sons.
- Emery et al. (1958)** JC Emery, JR Erwin, LJ Herrig e AE Felix. Systematic two-dimensional cascade tests of NACA 65 series compressor blades at low speeds. *NACA report 1368*.
- Fonseca e Fleming(1993)** Carlos M Fonseca e Peter J Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. Em *Proceedings of the Fifth International Conference, Citeseer*.
- Friedman(1991)** Jerome H Friedman. Multivariate adaptive regression splines. *The annals of*

statistics, 1(1): 1 – 67.

- Glaz et al(2008)** Bryan Glaz, Peretz P Friedmann e Li B Liu. Surrogate Based Optimization of Helicopter Rotor Blades for Vibration Reduction in Forward Flight, *Structural and Multidisciplinary Optimization*, 35(4):341-363.
- Goldberg e Holland(1988)** David E Goldberg e John Henry Holland. *Genetic Algorithms and Machine Learning*, Volume 1, Kluwer Academic Publishers.
- Gutmann(2001)** H-M Gutmann. A Radial Basis Function Method for Global Optimization. *Journal of Global Optimization*, 19(3):201–227.
- Hansen(2015)** Martin OL Hansen. *Aerodynamics of wind turbines*, volume 1, Routledge.
- Hess e Smith(1967)** Jhon L Hess e AMO Smith. Calculation of potential flow about arbitrary bodies, *Progress in Aerospace Sciences*. 8(1): 1-138.
- Hicks e Henne(1978)** Raymond Morton Hicks e Preston A Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412.
- Hicks et al(1974)** Raymond Morton Hicks, Earl Murman e Garret N Vanderplaats. An assessment of airfoil design by numerical optimization. *National Aeronautics and Space Administration*, NASA TM-X3092, p.33.
- Hinterding(2000)** Robert Hinterding. Representation, mutation and crossover issues in evolutionary computation. Em *Proceedings of the 2000 Congress on Evolutionary Computation, 2000*, IEEE.
- Horn et al(1994)** Jeffrey Horn, Nicholas Nafpliotis e David E Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. Em *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE.
- Huband et al(2006)** Simon Huband, Philip Hingston, Luigi Barone e Lyndon While. A review of multiobjective test problems and a scalable test problem toolkit, *IEEE Transactions on Evolutionary Computation*, 10(5):477-506.
- Hunsaker et al(2017)** Douglas F Hunsaker, Warren F Phillips e James J Joo. Aerodynamic Shape Optimization of Morphing Wings at Multiple Flight Conditions. Em *55th AIAA Aerospace Sciences Meeting*.
- Jackson(2005)** Paul Jackson. *All The World's Aircraft*, volume 2004-2005. Jane's.

- Jones et al(1998)** Donald R Jones, Matthias Schonlau e William J Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492.
- Katz e Plotkin(2001)** Joseph Katz e Allen Plotkin. *Low-speed aerodynamics*, volume 13, Cambridge University Press.
- Knowles e Nakayama(2008)** Joshua Knowles e Hirotaka Nakayama. *Multiobjective optimization: Meta-modeling in multiobjective optimization*. Springer Collection.
- Kumar e Minz(2014)** Vipin Kumar e Sonajharia V Minz. Multi-Objective Particle Swarm Optimization: An Introduction, *SmartCR*, 4(5):335-353.
- Kunakote e Bureerat(2013)** Tawatchai Kunakote e Sujin Bureerat. Surrogate-assisted multiobjective evolutionary algorithms for structural shape and sizing optimisation. *Mathematical Problems in Engineering*, 2013.
- Kung et al(1975)** Hsiang-Tsung Kung, Fabrizio Luccio e Franco P Preparata. On Finding the Maxima of a Set of Vectors. *Journal of the ACM (JACM)*, 22(4):469–476.
- Letcher(2017)**, Trevor M Letcher. *Wind energy engineering: a handbook for onshore and offshore wind turbines*, volume 1, Academic Press.
- Lyu et al(2015)** Zhoujie Lyu, Gaetan Kenway e Joaquim Martins. Benchmarking Optimization Algorithms for Wing Aerodynamic Design Optimization. *AIAA Journal*, 53(4):968-985.
- Mahmuddin(2017)** Faisal Mahmuddin. Rotor blade performance analysis with blade element momentum theory, *Energy Procedia*, 105(1):1123-1129.
- Manzanares et al. (2018)** Nelson Manzanares-Filho, Rodrigo Albuquerque, Bruno de Sousa e Luiz Guilherme Santos Cunha. A comparative study of controlled random search algorithms with application to inverse aerofoil design. *Engineering Optimization*, 50(6):996–1015.
- Matheron(1967)** G Matheron. Kriging, or polynomial interpolation procedures, *CIMM Transactions*. 70(1):240-244.
- Michalewicz(2013)** Zbigniew Michalewicz. *Genetic Algorithms + Data Structures: Evolution Programs*, volume 1, Springer Science & Business Media.
- Miettinen(2008)** Kaisa Miettinen. *Introduction to multiobjective optimization: Noninteractive approaches*, volume 1, Springer Collection.

- Moran(2003)** Jack Moran. *An introduction to theoretical and computational aerodynamics*, volume 1, Courier Corporation.
- Müller(2015)** Juliane Müller. MISO: Mixed-Integer Surrogate Optimization Framework, *Optimization and Engineering*, 17(1):177-203.
- Müller(2017)** Juliane Müller. SOCEMO: Surrogate optimization of computationally expensive multiobjective problems. *INFORMS Journal on Computing*, 29(4):581–596.
- Müller e Shoemaker(2014)** Juliane Müller e Christine Shoemaker. Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *Journal of Global Optimization*, 60(2):123-144.
- Nain e Deb(2005)** Pawan KS Nain e Kalyanmoy Deb. A multi-objective optimization procedure with successive approximate models. *KanGAL report*, 2(1).
- Obayashi et al(1997)** Shigeru Obayashi, Yoshihiro Yamaguchi e Takashi Nakamura. Multiobjective genetic algorithm for multidisciplinary design of transonic wing planform, *Journal of Aircraft*, 34(5):690-693.
- Ong et al(2005)** Yew Ong, Nair Soon, Keane PB e Wong AJ. Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. *Knowledge Incorporation in Evolutionary Computation*, p. 307–331.
- Osyczka(1985)** Andrzej Osyczka. *Design optimization: Multicriteria optimization for engineering design*. Elsevier Collection, p. 193-227.
- Pappa(2002)** Gisele Lobo Pappa. *Seleção de Atributos Utilizando Algoritmos Genéticos Multiobjetivo*, Dissertação de Mestrado, Pontifícia Universidade Católica de Paraná.
- Powell(1987)** Michael JD Powell. Radial basis functions for multivariable interpolation: a review. *Algorithms for approximation*. p. 143–167.
- Ramirez(2001)** Ramiro Gustavo Ramirez Camacho. *Análise de Escoamento em Grades de Turbomáquinas Axiais Incluindo o Efeito de Camada-Limite*, Tese de Doutorado, Universidade Federal de Itajubá.
- Regis e Shoemaker(2005)** Rommel G Regis e Christine A Shoemaker. Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global Optimization*, 31(1):153–171.

- Regis e Shoemaker(2007)** Rommel G Regis e Christine A Shoemaker. Parallel radial basis function methods for the global optimization of expensive functions. *European journal of operational research*, 182(2):514–535.
- Reis(2010)** Cleber Juliano dos Reis Braga. *Metodologia para Projeto Inverso Robusto de Perfis Aerodinâmicos e Grades de Turbomáquinas*. Dissertação de Mestrado. Universidade Federal de Itajubá.
- Reis et al(2019)** Cleber J Reis, Nelson Manzanares Filho e Antonio Marcos de Lima Gonçalves. Robust optimization of aerodynamic loadings for airfoil inverse designs. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 38(1):297-305.
- Renó(2007)**, Maria Luiza Grillo Renó. *Uso de Técnicas de Otimização Robusta Multi-Objetivos na Produção de Cimento*, Dissertação de Mestrado, Universidade Federal de Itajubá.
- Reyes e Coello(2006)** M Reyes Sierra e Carlos A Coello Coello. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2(3): 287–308.
- Rezek(2019)** Thiago Junqueira Rezek. *Metodologia de Projeto Hidrodinâmico de Turbinas Hidrocinéticas Carenadas Baseada na Otimização e Simulação em Dinâmica dos Fluidos Computacional*, Dissertação de Mestrado, Universidade Federal de Itajubá.
- Rippa(1999)** Shmuel Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation, *Advances in Computational Mathematics*. 11(1):193-210.
- Santos et al(2018)** Luis Guilherme Cunha Santos, Nelson Manzanares-Filho, GJ Menon e E Abreu. Comparing RBF-FD approximations based on stabilized Gaussians and on polyharmonic splines with polynomials. *International Journal for Numerical Methods in Engineering*, 115(4):462–500.
- Schaback(1995)** Robert Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3(3): 251-264.
- Schaffer(1985)** James David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. Em *Proceedings of the first international conference on genetic algorithms and their applications*. Lawrence Erlbaum Associates.
- Shields e Zhang(2016)** Michael D Shields e Jiaxin Zhang. The generalization of Latin hypercube sampling, *Reliability Engineering and System Safety*, 148(1):96–108.

- Schlichting e Scholz(1951)** H Schlichting e N Scholz. Über die theoretische Berechnung der Strömungsverluste eines ebenen Schaufelgitters, *Ingenieur-Archiv*. 19(1):42–65.
- Silva(2011)** Edna Raimunda da Silva Ramirez. *Técnicas de Metamodelagem Aplicadas à Otimização de Turbomáquinas*, Tese de Doutorado, Universidade Federal de Itajubá.
- Silva et al(2011)** Edna Raimunda da Silva Ramirez, Nelson Manzaneres Filho e Ramiro Gustavo Ramirez Camacho. Metamodelling approach using radial basis functions, stochastic search algorithm and CFD – application to blade cascade design. *International Journal of Mathematical Modelling and Numerical Optimisation*, 3(1):p. 82.
- Sousa(2008)** Bruno da Silva Sousa. *Projeto de Perfis Aerodinâmicos Utilizando Técnicas de Otimização Mono e Multiobjetivos*. Dissertação de Mestrado, Universidade Federal de Itajuba.
- Song et al(2011)** Fangfang Song, Yihua Ni e Zhiqiang Tan. Optimization design, modeling and dynamic analysis for composite wind turbine blade, *Procedia Engineering*, 16(1):369-375.
- Souza e Diniz-Ehrhardt(2011)** Matheus Souza e Maria A Diniz-Ehrhardt. *Otimização e Análise Convexa: Aspectos Teóricos e Aplicações*, volume 1. Universidade Estadual de Campinas, Instituto de Matemáticas e Estatística.
- Srinivas e Deb(1994)** Nidamarthi Srinivas e Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, (2)3: 221-248.
- Sun e Yuan(2006)**. Wenyu Sun e Ya-Xiang Yuan. Optimization Theory and Methods: nonlinear programming, volume 1. Springer Science & Business Media.
- Suppapitnarm et al.(2000)** A Suppapitnarm, Keith A Seffen, Geoff T Parks e PJ Clarkson. A simulated annealing algorithm for multiobjective optimization. *Engineering Optimization*, 33(1):59–85.
- Syberfeldt et al(2008)** Anna Syberfeldt, Henrik Grimm, Amos Ng e Robert I A John. Em 2008 *IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, IEEE p.3177-3184.
- Tang et al(2018)** Kit Sang Tang, Tak Ming Chan, Richard Jacob Yin e Kim Fung Man,. *Multiobjective optimization methodology: a jumping gene approach*, volume 1, CRC Press.
- Van Veldhuizen e Lamont(2000)** David A Van Veldhuizen e Gary B Lamont. On Measuring Multiobjective Evolutionary Algorithm Performance. Em *Proceedings of the IEEE*

Congress on Evolutionary Computation, IEEE.

- Velázquez et al(2014)** Miguel Velázquez Toledo, Marcelino del Carmen Vega, Juan Francisco Abugaber, and Luis A Pacheco Moreno e Guilibaldo Tolentino Eslava. Design and experimentation of a 1 MW horizontal axis wind turbine, *Journal of Power and Energy Engineering*.
- Wild et al(2008)** Stefan M Wild, Rommel G Regis e Christine A Shoemaker. ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*. 30(6):3197-3219.
- Zhu et al(2014)** Yongsheng Zhu, Jie Wang e Boyang Qu. Multi-objective economic emission dispatch considering wind power using evolutionary algorithm based on decomposition. *International Journal of Electrical Power & Energy Systems*, 63(1): 434–445.
- Zitzler et al(2013)** Eckart Zitzler, Kalyamoy Deb e Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195.
- Zitzler et al(2001)**. Eckart Zitzler, Marco Laumanns e Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *TIK-Report, ETH Zurich Research Collection*, 103(1):1–20.
- Zitzler e Thiele(1999)** Eckart Zitzler e Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

Apêndice A

Algoritmo para construção de metamodelos baseado em funções de Base Radial (RBF)

A seguir descreve-se um algoritmo para a construção de metamodelos baseado em funções de base radial, seguido de um exemplo literal baseado num banco de dados inicial com 4 pontos ($N_{exp} = 4$), uma função objetivo ($N_{obj} = 1$) e um espaço de decisão de duas variáveis ($n_{var} = 2$).

Início do Algoritmo:

Passo 1: *Construção do banco de dados do espaço de decisão.* Usando as técnicas de construção de planos de experiência (DOE), constrói-se a matriz do conjunto de pontos $S = \{\mathbf{X}_1, \dots, \mathbf{X}_{N_{exp}}\}$ do banco de dados, em que, $\mathbf{X}_i = [x_{i,j}]$; $i = 1, \dots, N_{exp}$ e $j = 1, \dots, n_{var}$; sendo então S de ordem $N_{exp} \times n_{var}$.

Tabela A.1: Exemplo: \mathbf{X}_i pontos da matriz de banco de dados

i	$\mathbf{X}_i = [x_j]; j = 1, 2$
1	$\mathbf{X}_1 = [x_{1,1} \ x_{2,1}]$
2	$\mathbf{X}_2 = [x_{1,2} \ x_{2,2}]$
3	$\mathbf{X}_3 = [x_{1,3} \ x_{2,3}]$
4	$\mathbf{X}_4 = [x_{1,4} \ x_{2,4}]$

Logo, a matriz de banco de dados no espaço de decisão fica:

$$S = \begin{bmatrix} x_{1,1} & x_{2,1} \\ x_{1,2} & x_{2,2} \\ x_{1,3} & x_{2,3} \\ x_{1,4} & x_{2,4} \end{bmatrix}$$

Passo 2: *Avaliação do banco de dados do espaço de decisão, na função real.* Cada ponto \mathbf{X}_i do banco de dados S , é avaliado nas funções reais f_k ; $k = 1, \dots, N_{obj}$, obtendo-se o vetor de funções objetivo para cada ponto do banco de dados, $\mathbf{F}_i = f_k(\mathbf{X}_i)$ em que, $i = 1, \dots, N_{exp}$ e $k = 1, \dots, N_{obj}$. Neste sentido, o espaço objetivo é representado através da matriz $\Omega = \{\mathbf{F}_1, \dots, \mathbf{F}_{N_{exp}}\}$; de ordem $N_{exp} \times N_{obj}$.

Tabela A.2: Exemplo: \mathbf{F}_i vetores objetivo para cada ponto do banco de dados

\mathbf{X}_i	$\mathbf{F}_i = f_k(\mathbf{X}_i); k = 1$
\mathbf{X}_1	$\mathbf{F}_1 = f_1(\mathbf{X}_1)$
\mathbf{X}_2	$\mathbf{F}_2 = f_1(\mathbf{X}_2)$
\mathbf{X}_3	$\mathbf{F}_3 = f_1(\mathbf{X}_3)$
\mathbf{X}_4	$\mathbf{F}_4 = f_1(\mathbf{X}_4)$

sendo a matriz do espaço objetivo:

$$\Omega = \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \\ \mathbf{F}_4 \end{pmatrix}$$

Passo 3: *Geração da matriz de dados para construção do metamodelo.* A matriz do banco de dados no espaço de decisão S e a matriz do espaço de objetivos Ω (Tabelas A.1 e A.2), são unidas para gerar a matriz $M_p = S \cup \Omega$, chamada: matriz de dados para construção do metamodelo, de ordem $N_{exp} \times (n_{var} + N_{obj})$.

$$M_p = \begin{bmatrix} x_{1,1} & x_{2,1} & \mathbf{F}_1 \\ x_{1,2} & x_{2,2} & \mathbf{F}_2 \\ x_{1,3} & x_{2,3} & \mathbf{F}_3 \\ x_{1,4} & x_{2,4} & \mathbf{F}_4 \end{bmatrix}$$

Passo 4: *Construção e cálculo da matriz de avaliação da RBF ($\boldsymbol{\varphi}$).* Constrói-se a matriz de avaliação da Função de Base Radial (ϕ), $\boldsymbol{\varphi} = [\varphi_{i,j}]$, em que cada elemento $\varphi_{i,j} = \phi(r_{ij})$ é função da distância radial $r_{ij} = \|\mathbf{X}_i - \mathbf{X}_j\|$ calculada entre os pontos i e j do banco de dados no espaço de decisão S . Caso por exemplo a RBF utilizada for a multiquádrica, $\phi(r) = \sqrt{r^2 + c^2}$ (Tabela A.3), o elemento φ_{13} é calculado com a distância (norma Euclidiana) entre os pontos (1) e (3) ($r_{13} = \|\mathbf{X}_3 - \mathbf{X}_1\|$), ou seja, $\varphi_{13} = \sqrt{(\|\mathbf{X}_3 - \mathbf{X}_1\|)^2 + c^2}$, em que c é o fator de forma.

$$\boldsymbol{\varphi} = \begin{pmatrix} \varphi_{11} & \varphi_{12} & \varphi_{13} & \varphi_{14} \\ \varphi_{21} & \varphi_{22} & \varphi_{23} & \varphi_{24} \\ \varphi_{31} & \varphi_{32} & \varphi_{33} & \varphi_{34} \\ \varphi_{41} & \varphi_{42} & \varphi_{43} & \varphi_{44} \end{pmatrix}$$

Mais especificamente, a distância no exemplo é:

$$\|\mathbf{X}_3 - \mathbf{X}_1\| = \sqrt{(x_{2,3} - x_{2,1})^2 + (x_{1,3} - x_{1,1})^2}$$

O caso particular $r_{ii} = 0$ define os valores φ_{ii} da diagonal principal da matriz $\boldsymbol{\varphi}$. Esses valores são constantes para cada tipo de RBF e são listados na Tabela A.3.

Tabela A.3: Valores da diagonal principal segundo a RBF utilizada.

RBF	ϕ	φ_{ii}
Gaussiana	$\exp(-r^2/c^2)$	1
Multiquádrica	$\sqrt{r^2 + c^2}$	c
Multiquádrica inversa	$1/\sqrt{r^2 + c^2}$	$1/c$
Poliharmônicas	$r^{2\beta-1} \quad \beta \in \mathbb{N}$	0
Splines Poliharmônicas	$r^{2\beta} \log r \quad \beta \in \mathbb{N}$	1

Observa-se que a matriz $\boldsymbol{\varphi}$ é simétrica $\varphi_{ij} = \varphi_{ji}$, e que todos seus valores não diagonais são positivos $\varphi_{ij} \geq 0$ para $i \neq j$. Seguindo o exemplo com a RBF multiquádrica, a matriz $\boldsymbol{\varphi}$ pode ser escrita como:

$$\boldsymbol{\varphi} = \begin{pmatrix} c & \varphi_{12} & \varphi_{13} & \varphi_{14} \\ \varphi_{12} & c & \varphi_{23} & \varphi_{24} \\ \varphi_{13} & \varphi_{23} & c & \varphi_{34} \\ \varphi_{14} & \varphi_{24} & \varphi_{34} & c \end{pmatrix}$$

Passo 5: *Construção da matriz de extensão de pontos do espaço de decisão.* No caso em que o metamodelo seja construído usando alguma RBF com o polinômio de extensão linear (usado neste trabalho), é preciso construir uma matriz *de extensão* \mathbf{P} pela união de um vetor coluna unitário $\hat{\mathbf{u}}^T$ com a matriz de pontos do espaço de decisão S , ou seja, $\mathbf{P} = \hat{\mathbf{u}}^T \cup S$, de ordem $N_{exp} \times (n_{var} + 1)$:

$$\mathbf{P} = \begin{pmatrix} 1 & x_{1,1} & x_{2,1} \\ 1 & x_{1,2} & x_{2,2} \\ 1 & x_{1,3} & x_{2,3} \\ 1 & x_{1,4} & x_{2,4} \end{pmatrix}$$

Passo 6: *Cálculo dos coeficientes de interpolação α_i e μ_i .* Dependendo da RBF ter ou não polinômio de extensão, para o cálculo dos coeficientes de interpolação deve-se resolver um dos seguintes sistemas de equações algébricas lineares:

Sem polinômio de extensão:

$$[\boldsymbol{\varphi}]\{\boldsymbol{\alpha}\} = \{\mathbf{F}\}$$

Seguindo o exemplo:

$$\begin{bmatrix} c & \varphi_{12} & \varphi_{13} & \varphi_{14} \\ \varphi_{21} & c & \varphi_{23} & \varphi_{24} \\ \varphi_{31} & \varphi_{32} & c & \varphi_{34} \\ \varphi_{41} & \varphi_{42} & \varphi_{43} & c \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix}$$

Com polinômio de extensão

$$\begin{bmatrix} \boldsymbol{\varphi} & \mathbf{P} \\ \mathbf{P}^T & \hat{\mathbf{0}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \hat{\mathbf{0}} \end{bmatrix}$$

Seguindo o exemplo:

$$\begin{bmatrix} c & \varphi_{12} & \varphi_{13} & \varphi_{14} & 1 & x_{1,1} & x_{2,1} \\ \varphi_{21} & c & \varphi_{23} & \varphi_{24} & 1 & x_{1,2} & x_{2,2} \\ \varphi_{31} & \varphi_{32} & c & \varphi_{34} & 1 & x_{1,3} & x_{2,3} \\ \varphi_{41} & \varphi_{42} & \varphi_{43} & c & 1 & x_{1,4} & x_{2,4} \\ \hline 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & 0 & 0 & 0 \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \mu_0 \\ \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Passo 7: Construção dos metamodelos $\tilde{f}_k(\mathbf{X})$. Calculados os coeficientes de interpolação, α_i e μ_i segundo seja o caso, constroem-se os metamodelos através de uma das expressões seguintes:

Sem polinômio de extensão:

$$\tilde{f}_k(\mathbf{X}) = \sum_{i=1}^{N_{exp}} \alpha_i \phi(\|\mathbf{X} - \mathbf{X}_i\|)$$

Com polinômio de extensão:

$$\tilde{f}_k(\mathbf{X}) = \sum_{i=1}^{N_{exp}} \alpha_i \phi(\|\mathbf{X} - \mathbf{X}_i\|) + \mu_0 + \sum_{i=1}^{n_{var}} \mu_i x_i$$

Seguindo o exemplo:

$$\begin{aligned} \tilde{f}(\mathbf{X}) &= \alpha_1 \phi(\|\mathbf{X} - \mathbf{X}_1\|) + \alpha_2 \phi(\|\mathbf{X} - \mathbf{X}_2\|) + \alpha_3 \phi(\|\mathbf{X} - \mathbf{X}_3\|) + \dots \\ &+ \alpha_4 \phi(\|\mathbf{X} - \mathbf{X}_4\|) + \mu_0 + \mu_1 x_1 + \mu_2 x_2 \quad \forall \mathbf{X} = (x_1, x_2) \end{aligned}$$

Apêndice B

Algoritmo para automatização da escolha do parâmetro de forma c

Na implementação desta primeira versão do algoritmo MO-CORS, unicamente para problemas mono-objetivo, foi implementada e testada uma estratégia de automatização da escolha do parâmetro de forma das funções de base radial, Gaussiana, multiquádrica e inversa multiquádrica. No algoritmo MO-CORS o banco de dados de pontos para avaliação da função custosa é atualizado a cada iteração e um novo metamodelo é construído com esses pontos. Diaz *et al.* (2018) demonstraram que o valor escolhido afeta fortemente o efeito de aceleração do CORS (Estratégia seguida pelo algoritmo MO-CORS no caso mono-objetivo) ao usar funções multiquádrica, (dependentes de c) em comparação ao uso das poliharmônicas na construção de metamodelos. No contexto de MO-CORS, é importante então uma escolha adequada do valor de c no uso de RBF que dele dependam. Portanto, para evitar escolhas arbitrárias em cada novo problema, é desejável um esquema confiável para o ajuste automático de valor entre as iterações do MO-CORS.

B.1. Experimentos computacionais LOOCV.

Num trabalho submetido para publicação foram apresentados resultados de experimentos computacionais com funções de teste para avaliar o desempenho da função de medida de erro de Rippla (1999) em comparação com o erro real (RMSE) produzido por metamodelos construídos com RBF dependentes de c . O intuito foi extrair informações úteis para testes subsequentes com LOOCV no contexto do algoritmo CORS, cujo propósito ideal é reduzir ao mínimo possível a quantidade de avaliações da função objetivo custosa. Nesse contexto, o metamodelo não representa um produto em si, mas um artefato auxiliar a ser reconstruído a

cada iteração e descartado no final – com a menor quantidade possível de pontos no banco de dados.

Para o trabalho submetido, foram realizados 21 conjuntos de teste computacionais envolvendo 7 funções de teste de Dixon-Szegö (1978) e as RBF multiquádrica, inversa multiquádrica e Gaussiana. Resultados para três conjuntos de teste típicos estão ilustrados nas Figuras B.1, B.2 e B.3. À esquerda, são mostradas curvas de variação com c do erro médio quadrático real (RMSE) e da função de erro de Rippa (F_{erro}) usando as normas unitária (l_1) e a Euclidiana (l_2). À direita acima, essas mesmas curvas estão destacadas na faixa de valores c que produzem pequenos erros (faixa de interesse). À direita abaixo, é mostrada a variação do número de condição N_{cond} da matriz de interpolação, que pode atingir valores extremamente altos na região de interesse (caso da Fig, B.1, por exemplo).

Verifica-se que a tendência de variação da função de erro de Rippa (F_{erro}) em ambas as normas reproduz bem a tendência de variação do erro real (RMSE), em especial nas faixas de interesse. Os valores de erro não são importantes aqui. O que importa é um valor de c que produza um erro próximo ao mínimo.

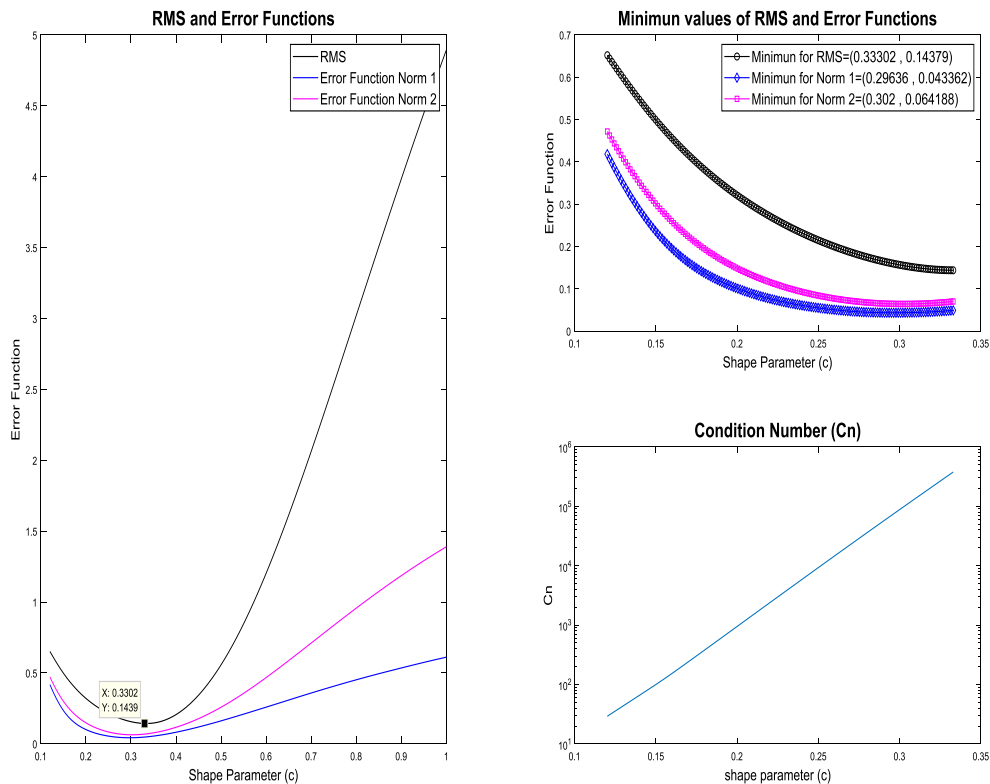


Figura B.1: Erro RMS e F_{erro} (l_1 e l_2) - função Shekel 5 - RBF multiquádrica.

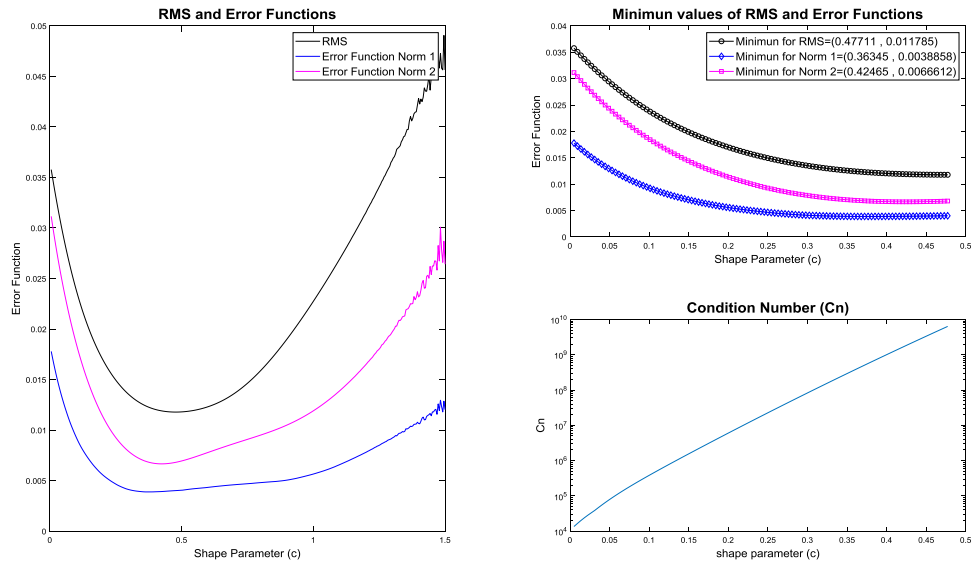


Figura B.2: Erro RMS e F_{erro} (l_1 e l_2) - função Hartmann 3 - RBF inversa multiquádrica.

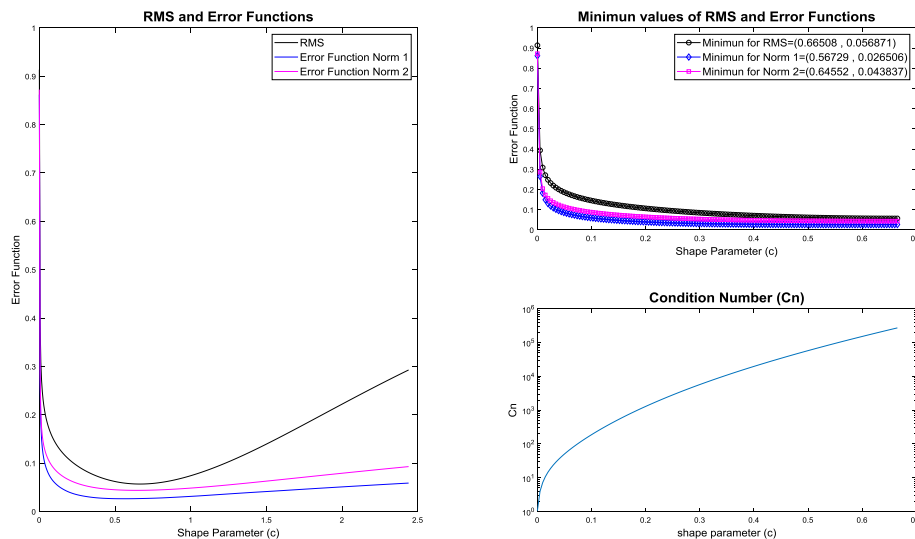


Figura B.3: Erro RMS e F_{erro} (l_1 e l_2) - função Hartmann 6 - RBF Gaussiana.

Tendo em vista os resultados gerais da pesquisa, o algoritmo eficiente LOOCV de Rippa aparece como opção promissora para realizar a almejada escolha automática do valor do parâmetro de forma, no contexto do algoritmo MO-CORS. Nesse sentido, seria suficiente minimizar a função de erro de Rippa (com a norma l_1 ou l_2), para obter um valor ótimo de c em um intervalo adequado. O limite superior desse intervalo deve ser controlado para evitar um mau condicionamento excessivo da matriz de interpolação (número de condição muito alto). Daí então, o limite inferior poderia ser calculado como uma pequena fração do limite superior.

B.2. Algoritmo para escolha automática de c .

Este algoritmo permite a construção dos metamodelos provisórios ao longo do processo iterativo com os parâmetros de forma ajustados pelo algoritmo eficiente Rippa (LOOCV). A base do algoritmo é minimizar a função de erro de Rippa (F_{erro}) num intervalo controlado pelo número de condição da matriz de interpolação $c_{LI} \leq c \leq c_{LS}$

Algoritmo para escolha automática de c (em cada iteração do MO-CORS)

Passo 1: *Dados preliminares necessários.* Devido que este procedimento é utilizado dentro do processo iterativo de construção do metamodelo, ele requer previamente em cada iteração t da matriz de construção do metamodelo normalizada \hat{M}_p que contem o banco de dados do espaço de decisão e o vetor objetivo (problema mono-objetivo).

$$\hat{M}_p = \hat{S} \cup \hat{F}$$

Passo 2: *Inicialização do limite superior de valores de c .* Seleciona-se o limite superior inicial $(c_{LS})_0$. Como o parâmetro de forma nas RBF representam uma distância dentro do espaço de decisão normalizado, esse limite inicial é assumido como a maior distância possível nesse espaço $(\sqrt{n_{var}})$.

$$c_{LS} \leftarrow (c_{LS})_0 \leftarrow \sqrt{n_{var}}$$

Passo 3: *Construção da matriz de interpolação.* Usando a matriz de construção do metamodelo normalizada \hat{M}_p , se constrói a matriz de interpolação $\boldsymbol{\varphi}$, segundo procedimento estabelecido no Apêndice A, usando o parâmetro de forma $c = c_{LS}$ e uma Função de Base Radial (RBF), Gaussiana, multiquádrica ou inversa multiquádrica.

Passo 4: *Cálculo do número de condição (N_{cond}) da matriz de interpolação.* Sendo que $\|\mathbf{A}\|$ representa a norma de uma matriz qualquer \mathbf{A} ., então

$$N_{cond} = \|\boldsymbol{\varphi}\| \|\boldsymbol{\varphi}^{-1}\|$$

Passo 5: *Teste de convergência.* Se $\log(N_{cond}) > 16$,

Então faça

Passo 5.1: *Cálculo do novo valor do limite superior do intervalo de c .* O novo limite superior c_{LS} é uma fração ($0 < \lambda < 1$) do limite superior anterior. Sendo então:

$$c_{LS} \leftarrow \lambda c_{LS}$$

Passo 5.2: Volte ao Passo 3.

Se não, continue.

Passo 6: *Cálculo do valor do limite inferior de valores de c .* O limite inferior c_{LI} é uma fração pequena ($0 < \varepsilon \ll 1$) do limite superior. Sendo então:

$$c_{LI} \leftarrow \varepsilon c_{LS}$$

Passo 7: *Minimização da função erro F_{erro} .* Resolve-se um problema de otimização mono-objetivo para determinar o valor do parâmetro de forma c que minimiza a função de erro de Rippa, calculado segundo seu algoritmo eficiente, baseado na técnica LOOCV.

Função objetivo: (com norma euclidiana)

$$E_i = \frac{\alpha_i}{[\varphi_{ii}]^{-1}}; \quad F_{erro}(c) = \left(\frac{1}{N_{exp}} \sum_{i=1}^{N_{exp}} (E_i)^2 \right)^{\frac{1}{2}}$$

em que N_{exp} é o número de pontos para a construção do metamodelo.

Problema de otimização:

$$\begin{array}{l} \text{Minimizar } F_{erro}(c) \\ \text{Sujeito a:} \\ c_{LI} \leq c \leq c_{LS} \end{array}$$

Passo 8: *Fim.* O resultado do problema de otimização c^* , é o valor usado para a construção iterativa do metamodelo dentro do contexto MO-CORS, unicamente nos problemas mono-objetivo.

Apêndice C

Desempenho de algoritmos de otimização em problema aerodinâmico de asas

O presente estudo foi utilizado para determinar o comportamento de diferentes algoritmos de otimização multi-objetivo (MOOA) aplicados no projeto aerodinâmico de asas (DIAZ et al., 2018). Geralmente, um método não pode ser declarado melhor que outros sem aplicá-los comparativamente a um problema específico que precisa ser enfrentado. Nos algoritmos de otimização multi-objetivo as características mais importantes dos resultados na frente de Pareto que precisam ser avaliadas incluem: 1) a cobertura 2) a diversidade e 3) a capacidade de geração de pontos, além da convergência do próprio algoritmo. Estas características foram avaliadas através de métricas de desempenho, nos algoritmos de otimização multi-objetivo: NSGA II, MOSA, MOPSO e MOGA II aplicados em dois problemas aerodinâmicos da asa.

C.1. Metodologia de otimização.

São resolvidos dois problemas de otimização de dois objetivos tipo min-min, em que $f_1 = 1/C_L$ e $f_2 = |C_m|$. Para o problema 1 as variáveis de decisão são os parâmetros da asa e para o problema 2, são as ordenadas dos pontos de controle de Bézier que geram a forma do aerofólio. Para a resolução dos problemas de otimização multi-objetivo foi usado o software ModeFrontier. Este permite utilizar vários algoritmos multi-objetivos entre os quais estão NSGA II, MOSA, MOPSO e MOGA II. Os problemas são estruturados através de uma interface gráfica em que é possível configurar um diagrama de fluxo que descreve o processo onde são envolvidas as variáveis de decisão, parâmetros fixos do problema, os códigos computacionais necessários para o cálculo das funções objetivo e os resultados destes valores ao final do processo de otimização para o processamento dos resultados. A seguir apresentam-se nas

Figuras C.1 e C.2 os digramas gerais na interface ModeFrontier dos problemas aerodinâmicos desenvolvidos.

C.1.1. Diagrama de fluxo do problema de otimização de asas.

Na Figura C.1 e C.2 ilustram-se os esquemas gerais dos diagramas de fluxo dos problemas aerodinâmicos de asas, tratados nesta pesquisa.

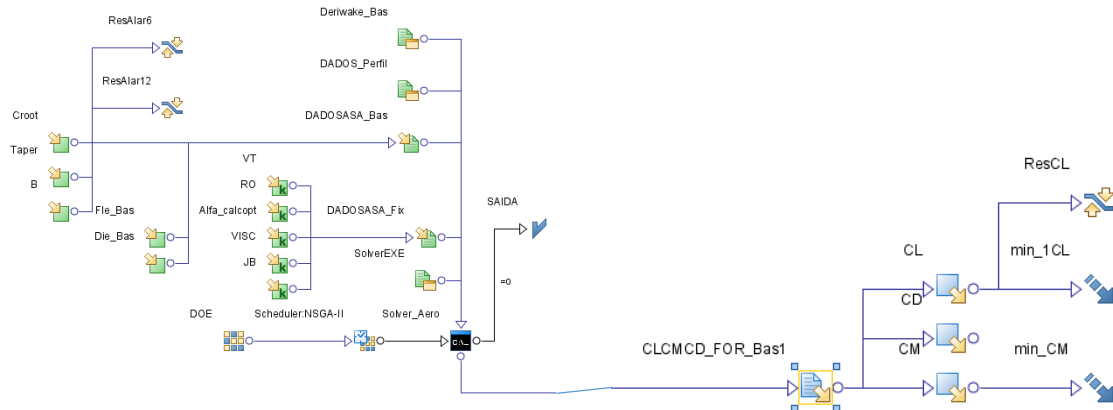


Figura C.1 Diagrama de fluxo do problema aerodinâmico de asas 1 (Variação de parâmetros de asa)

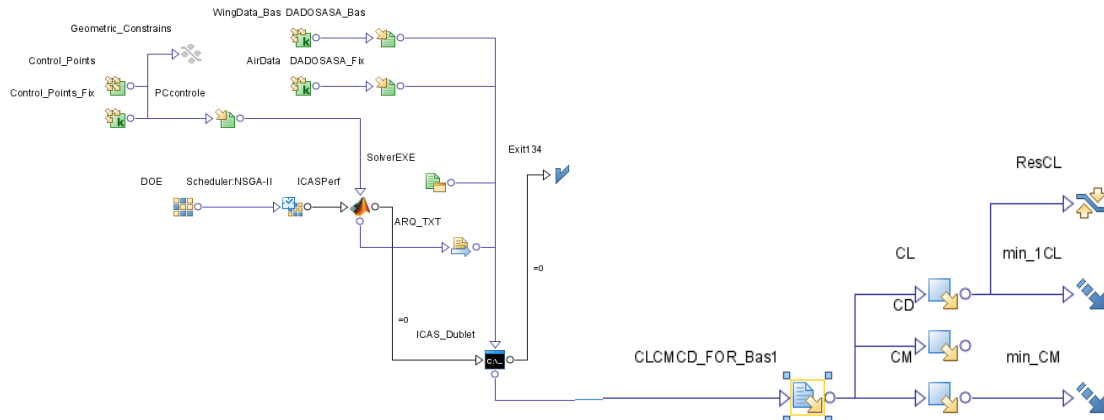


Figura C.2. Diagrama de fluxo do problema aerodinâmico de asas 2 (Variação da forma do aerofólio)

Podem-se observar as caixas das variáveis de decisão (Caixa verde com seta amarela) dos dois problemas, além dos parâmetros constantes (Caixa verde com seta amarela e letra "K") do lado esquerdo. Do lado direito as caixas azuis com seta amarela representam os valores para o cálculo das funções objetivo e as setas tracejadas representam as funções objetivo minimizadas. Outros aspectos importantes são as caixas de interface com programas executáveis e códigos de MATLAB para o cálculo das funções caixa preta. Por exemplo, no problema 2, um código em MATLAB é utilizado para o cálculo dos pontos de controle do aerofólio e esses resultados

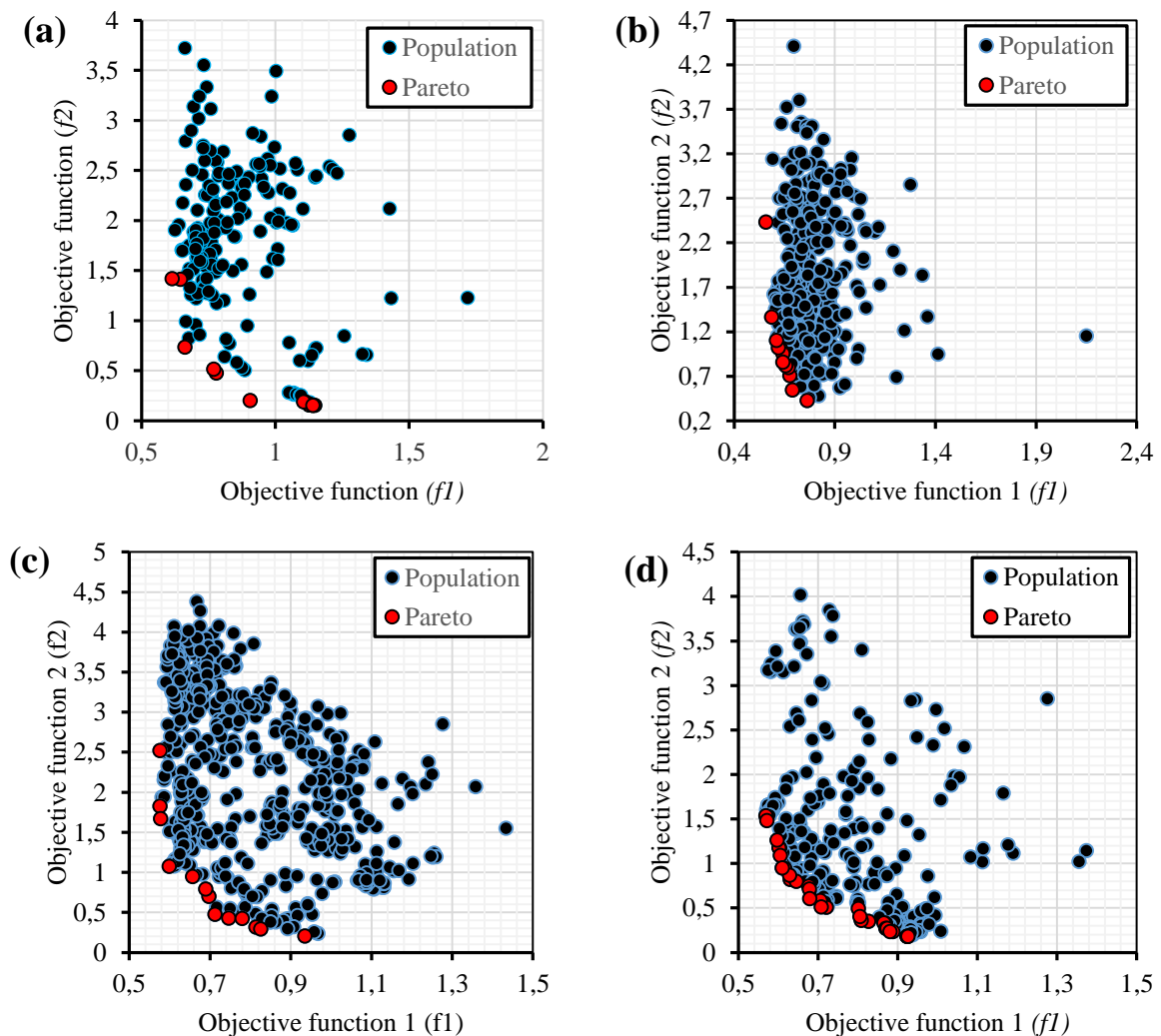
são enviados à caixa preta do executável em FORTRAN para o cálculo dos coeficientes aerodinâmicos através do método dos painéis 3D.

C.2. Experimentos computacionais.

Os parâmetros de operação de cada algoritmo foram mantidos na condição “*default*” ou valores padrão de ModeFrontier, e as populações iniciais (DOE) foram mantidas constantes. O número de indivíduos na população inicial (número de projetos) foi fixado em 20 e o número de iterações ou gerações, em 30. Foi aplicado um número baixo de gerações no intuito de reduzir o tempo computacional em cada execução, em que uma quantidade total de 600 indivíduos é avaliada.

C.3. Resultados para os problemas 1 e 2.

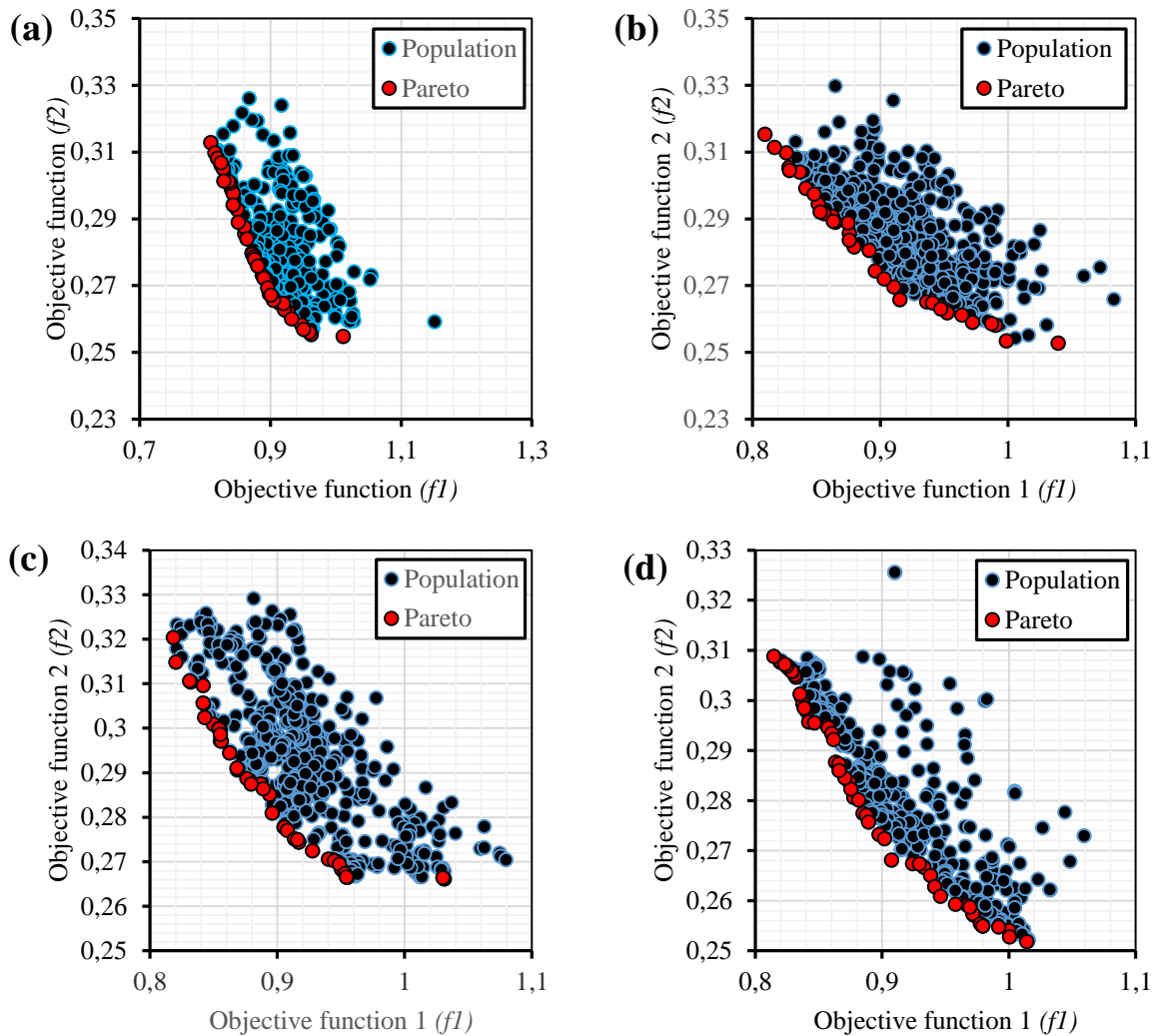
C.3.1. Evolução dos algoritmos de otimização (Problema 1).



Figuras C.3: Evolução de (a) MOGAI, (b) MOPSO, (c) MOSA e (d) NSGAI

(Problema 1 - Variação de parâmetros de asa)

C.3.2. Evolução dos algoritmos de otimização (Problema 2).



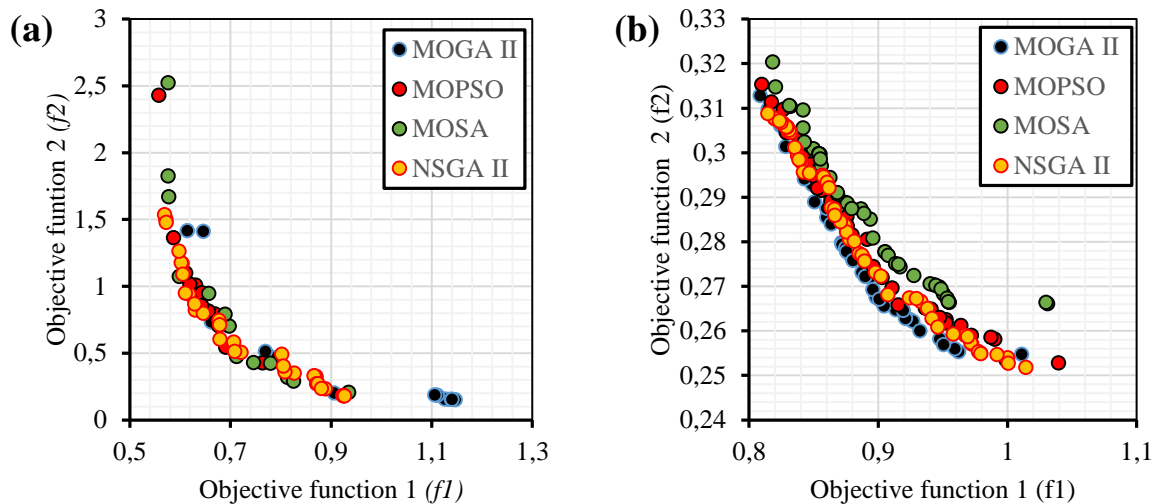
Figuras C.4: Evolução de (a) MOGAI, (b) MOPSO, (c) MOSA e (d) NSGAI (Problema 2 - Variação de aerofólio)

Para o problema 1, Figura C.3^a, observa-se para o algoritmo MOGA II que a população no início é dispersa e que, durante a evolução, foi difícil para o algoritmo encontrar a frente de Pareto, sendo ela bastante pobre. Os algoritmos MOPSO e MOSA (Figuras C.3b e C.3c) comprimem a população rapidamente, mas os pontos nas frentes de Pareto são poucos e dispersos. A Figura C.3d mostra que o NSGAI apresenta uma distribuição dispersa de pontos nas iterações iniciais e converge gradualmente numa frente de Pareto comprimida. Para o problema 2, observa-se nas Figuras C.4 que todos os algoritmos apresentam uma evolução rápida até a frente de Pareto. Os algoritmos MOPSO e MOGAI comprimem a população rapidamente, atingindo uma representação considerável de pontos nas frentes de Pareto. A Figura C.4d mostra que NSGAI apresenta uma evolução rápida até a frente final, com uma

quantidade considerável de pontos distribuídos com boa uniformidade. A seguir são ilustradas as figuras da frente das frentes de Pareto de cada algoritmo de otimização.

C.3.3. Frente das frentes de Pareto dos problemas 1 e 2.

Na Figura A.1 ilustra-se a distribuição de pontos das Frentes de Pareto de cada algoritmo de otimização no intuito de observar se algumas soluções de um algoritmo dominam as soluções do outro algoritmo.



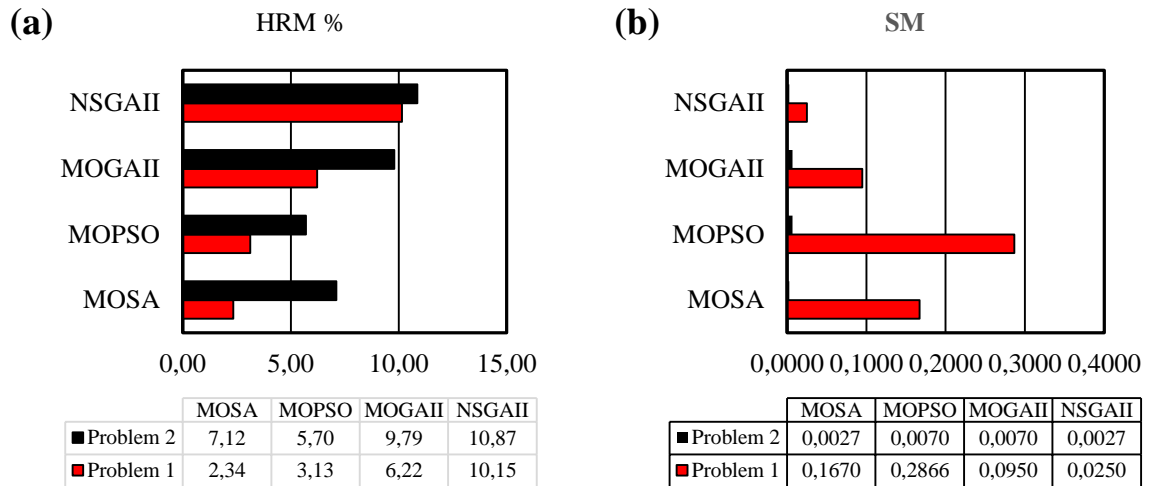
Figuras C.5: Comparação entre frentes (a) problemas 1 e (b) problema 2

A Figura C.5a e C.5b apresenta uma comparação entre as soluções de cada algoritmo aplicado nos problemas aerodinâmicos de asa 1 e 2. Aqui pode-se observar no caso do problema 1 (Figura C.5a), a dominância dos resultados do NSGA II sobre os outros algoritmos. Para o problema 2 (Figura C.5b) pode-se observar uma clara dominância dos pontos resultantes do algoritmo MOGA II. Estes resultados são quantificados mais na frente com a métrica BPV. A seguir apresentam-se os resultados das métricas de sucesso, cobertura e convergência para cada algoritmo de otimização aplicado em cada problema aerodinâmico de asas.

C.4. Métricas de desempenho.

Para o cálculo das métricas de desempenho dos problemas de otimização multi-objetivo, foi desenvolvido um script de MATLAB, elaborado com base nas definições descritas no capítulo 2.

A seguir nas Figuras C.6, C.7 e C.8, são apresentados os resultados obtidos de cada métrica de desempenho de cada algoritmo de otimização. Para as métricas de sucesso e cobertura, são apresentadas também tabelas com seus resultados numéricos.



Figuras C.6. Métricas de taxa de sucesso percentual (HRM%) e espaçamento (SM)

Os valores da métrica HRM% da Figura C.6a mostram que para os dois problemas aplicados, o algoritmo NSGA II tem a melhor capacidade de encontrar um grande número de pontos na frente de Pareto. O MOSA apresenta a pior capacidade para o problema 1 e o MOPSO para o problema 2. Além de NSGAII mostrar a maior capacidade, a Figura C.6b mostra que a melhor cobertura para ambos problemas aerodinâmicos de asa é obtida também pelo algoritmo NSGAII cujos valores de espaçamento foram as menores e os piores valores para esse parâmetro foram calculados para o MOPSO.

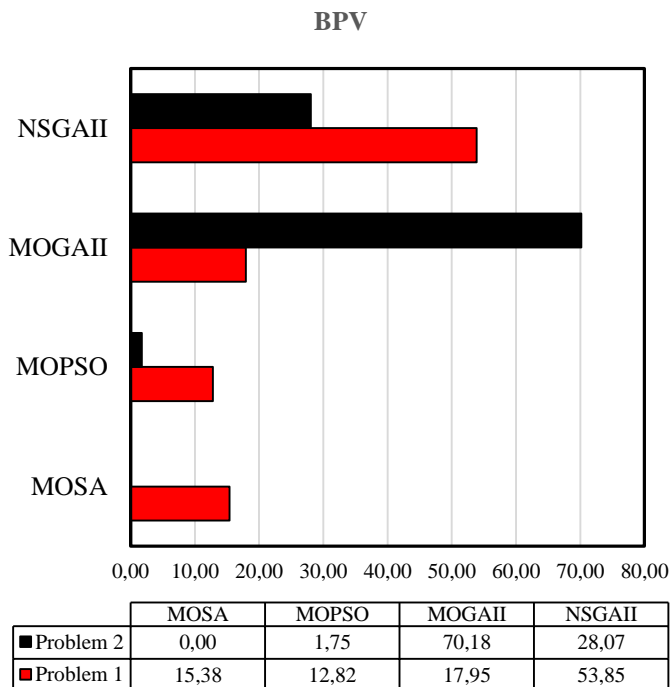


Figura C.7. Métrica percentual de melhores pontos na frente das frentes (BPV)

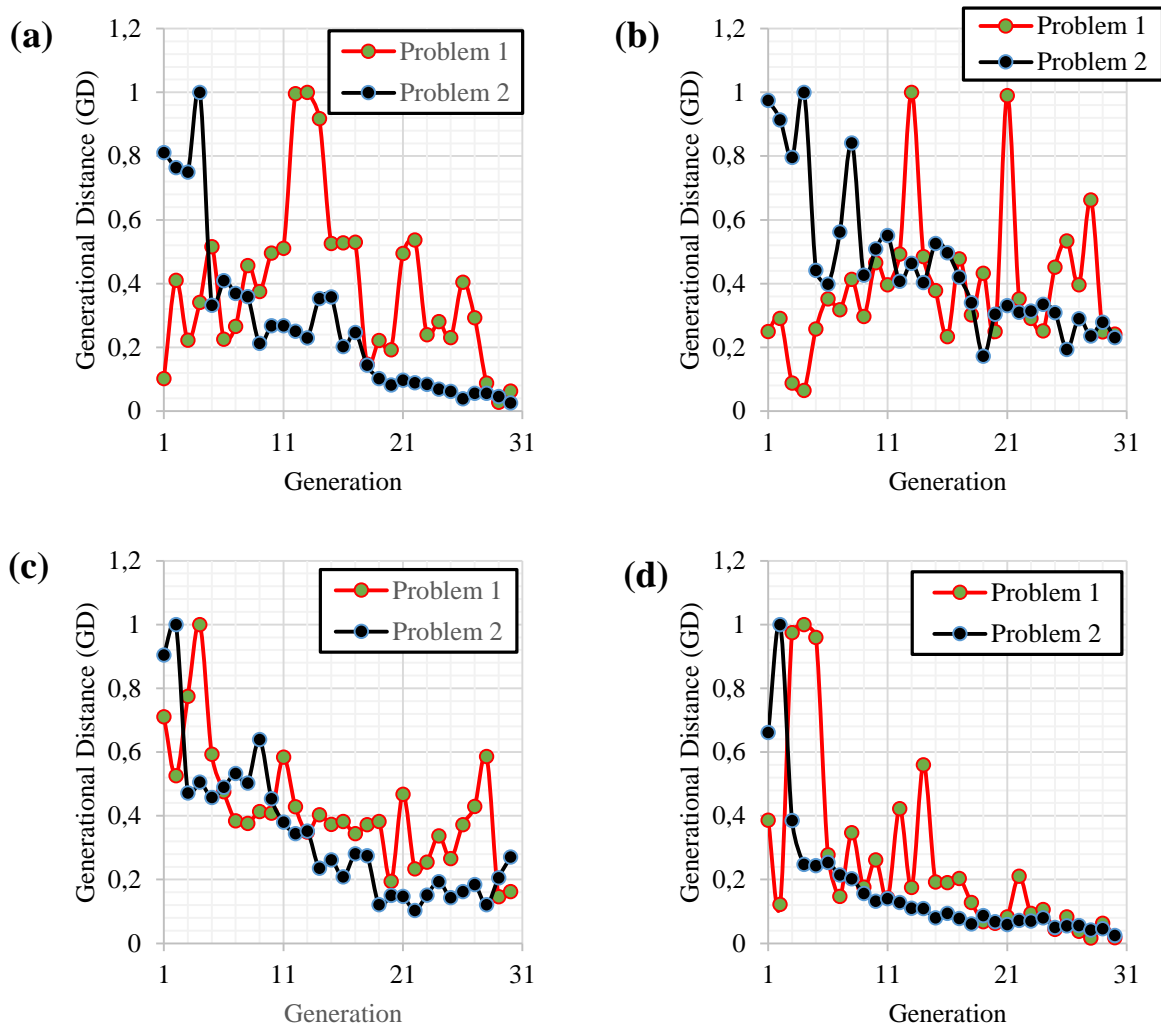


Figura C.8. Métrica de distância geracional (GD). (a) MOGAI, (b) MOSA, (c) MOPSO e (d) NSGAI

Para cada problema aerodinâmico, foi avaliada a métrica BVP da frente das frentes de Pareto (Figuras C.7). Nesse aspecto, o melhor algoritmo para o problema 1 que envolve apenas parâmetros de asa, foi o NSGA II, com uma contribuição de 53,85% do total de pontos da frente de Pareto geral. A mesma análise foi feita para o problema 2, e nesse caso o melhor algoritmo foi o MOGA II com 70,18%.

A Figura C.8 mostra que a melhor convergência para os problemas de asa aerodinâmica é obtida usando o método NSGA II. Mesmo levando em consideração que, neste trabalho, o número de gerações utilizadas foi relativamente baixo para estes tipos de algoritmo, ressalta-se a rápida convergência atingida pelo algoritmo NSGAI, principalmente no problema 2, em que o número de variáveis envolvidas foi de 10. Pela análise das métricas, pode-se considerar que o algoritmo NSGAI atingiu o melhor desempenho geral nos problemas de aerodinâmica de asas escolhidos. Desse modo, ele foi selecionado para a resolução dos problemas auxiliares 1 e 2 do algoritmo MO-CORS.