OTIMIZAÇÃO MULTI-OBJETIVO DE ROTAS DE DUTOS SUBMARINOS ATRAVÉS DE SISTEMAS IMUNOLÓGICOS ARTIFICIAIS

Ian Nascimento Vieira

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Civil.

Orientadores: Breno Pinheiro Jacob
Beatriz de Souza Leite Pires de Lima

Rio de Janeiro
Setembro de 2011

OTIMIZAÇÃO MULTI-OBJETIVO DE ROTAS DE DUTOS SUBMARINOS ATRAVÉS DE SISTEMAS IMUNOLÓGICOS ARTIFICIAIS

Ian Nascimento Vieira

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Examinada por:

Prof. Breno Pinheiro Jacob, Dsc.

Profª. Beatriz de Souza Leite Pires de Lima, Dsc.

Prof. Nelson Francisco Favilla Ebecken, Dsc.

Prof. Afonso Celso de Castro Lemonge, Dsc.

Eng. Carlos de Oliveira Cardoso, Dsc.

Eng. Isaias Quaresma Masetti, Dsc.

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2011

Vieira, Ian Nascimento

Otimização Multi-Objetivo de Rotas de Dutos Submarinos Através de Sistema Imunológicos Artificiais/ Ian Nascimento Vieira. – Rio de Janeiro: UFRJ/COPPE, 2011.

XIX, 153 p.: il.; 29,7 cm.

Orientadores: Breno Pinheiro Jacob

Beatriz de Souza Leite Pires de Lima

Tese (doutorado) – UFRJ/ COPPE/ Programa de Engenharia Civil, 2011.

Referências Bibliográficas: p. 144-150.

1. Otimização Multi-objetivo. 2. Rota de Dutos Submarinos. 3. Sistemas Imunológicos Artificiais. I. Jacob, Breno Pinheiro *et al.*. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Titulo.

Aos meus pais, Carlos e Edna
e aos meus orientadores Beatriz e Breno.

AGRADECIMENTOS

A Deus, por criar uma natureza tão bela que nos inspira a matemática e a vida e pela sua bondade imensurável.

Ao professor Breno Pinheiro Jacob e à professora Beatriz S. L. P. de Lima, pela orientação, motivação e apoio, fundamentais na realização desta Tese.

À minha namorada Mariana Vaz pela compreensão das muitas horas dedicadas ao trabalho científico em detrimento da sua companhia.

Ao professor Carl Horst Albrecht, pela constante instrução, amizade e ajuda no desenvolvimento do trabalho.

Aos amigos Juliana Baioco, Daniel Coutinho, Allan Fonseca, Rodrigo Lucerna e Marcelo Zmetek por me darem todo auxílio técnico possível e imaginável além da felicidade de tê-los como amigos.

À minha família, pelo incentivo e apoio incondicionais.

Aos meus amigos, afastados pela realização desta tese.

Aos meus amigos do laboratório LAMCSO (Laboratório de Métodos Computacionais em Sistemas *Offshore*): Mauro, Bruno J., Bruno M., Fabrício, Aldo, Pablo, Fábio...

À Ivete Silva de Araujo Frango por toda ajuda que me prestou com rápidos documentos e rápidas resoluções de problemas.

Ao INPI por ter concedido horário especial de estudante fundamental à realização desta tese;

A todos que, de alguma forma, colaboraram com a elaboração deste trabalho.

E ao CNPq pelo apoio financeiro.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

OTIMIZAÇÃO MULTI-OBJETIVO DE ROTAS DE DUTOS SUBMARINOS ATRAVÉS DE SISTEMAS IMUNOLÓGICOS ARTIFICIAIS

Ian Nascimento Vieira

Setembro/2011

Orientadores: Breno Pinheiro Jacob
Beatriz de Souza Leite Pires de Lima

Programa: Engenharia Civil

Este trabalho apresenta o desenvolvimento de um algoritmo de otimização multi-objetivo baseado em Sistemas Imunológicos Artificiais denominado SIAMEP – Sistema Imunológico Artificial com Memória Externa de Pareto, e sua implementação em uma ferramenta computacional orientada à otimização de rotas de dutos submarinos. O algoritmo é validado através de testes *benchmark* com funções matemáticas e problemas de engenharia. Os resultados alcançados demonstram que o algoritmo desenvolvido foi capaz de resolver de maneira eficiente os problemas de otimização, em particular a otimização multi-objetivo de rotas de dutos submarinos, podendo contribuir no estabelecimento de metodologias inovadoras de projeto destes importantes sistemas de produção de petróleo offshore.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

MULTIOBJECTIVE OPTIMIZATION OF SUBMARINE PIPELINE ROUTES BY ARTIFICIAL IMMUNE SYSTEMS

Ian Nascimento Vieira

September/2011

Advisors: Breno Pinheiro Jacob
Beatriz de Souza Leite Pires de Lima

Department: Civil Engineering

This work presents the development of a multi-objective optimization algorithm referred as SIAMEP - Artificial Immune System with Pareto External Memory, and its implementation on a computational tool oriented towards the optimization of submarine pipeline routes. The algorithm is validated through benchmark mathematical functions and engineering problems. The results demonstrate that the algorithm was able to efficiently solve optimization problems, particularly the multi-objective optimization of submarine pipeline routes, and therefore is able to contribute to the development of innovative design methodologies for such important offshore oil production systems.

SUMÁRIO

CAPÍTULO 1 INTRODUÇÃO 1

1.1 CONTEXTO E MOTIVAÇÃO 1

1.2 OBJETIVOS DO TRABALHO 2

1.3 ROTA DE DUTOS SUBMARINOS 3

1.3.1 Introdução 3

1.3.2 Histórico 4

1.3.3 Metodologia 6

1.4 ORGANIZAÇÃO DA TESE 7

CAPÍTULO 2 OTIMIZAÇÃO MULTI-OBJETIVO 8

2.1 CONCEITOS BÁSICOS 8

2.1.1 Formulação Matemática 8

2.1.2 Soluções Pareto-ótimas 10

2.1.3 Propósito da Otimização Multi-objetivo 12

2.1.4 Diferenças com a Otimização uni-objetivo 12

2.2 ESTADO DA ARTE 13

2.2.1 Algoritmos Genéticos (AG) 13

2.2.1.1 VEGA 14

2.2.1.2 MOGA 15

2.2.1.3 NPGA 15

2.2.1.4 NSGA 16

2.2.1.5 NSGA-II 17

2.2.1.6 PMOGA 17

2.2.2 Estratégias Evolutivas (EE) 17

2.2.2.1 SPEA 18

2.2.2.2 SPEA2 19

2.2.2.3 PAES 19

2.2.2.4 (1+1) PAES-M 20

2.2.3 Sistemas Imunológicos Artificiais (SIA) 21

2.2.3.1 MISA 30

2.2.3.2 MOIA 31

2.2.3.3 MOCSA 31

2.2.3.4 VAIS 32

2.2.3.5 IDCMA 32

2.2.3.6 IFMOA 33

2.2.3.7 Omni-AiNet 33

2.2.3.8 PAIA 34

2.2.3.9 ACSAMO 34

2.2.3.10 CNMOIA 35

2.2.3.11 SIAOM 35

2.2.3.12 EMOIA 36

2.2.3.13 HIMO 37

2.2.3.14 WBMOAIS 38

2.3 MEDIDAS DE DESEMPENHO 38

2.3.1 Métricas de Convergência 40

2.3.1.1 Taxa de Erro (ER) 40

2.3.1.2 Distância Geracional (GD) ... 40
2.3.2 Métricas de Diversidade ... 41
2.3.2.1 Espaçamento (SP) ... 41
2.3.2.2 Espalhamento (Δ) ... 42
CAPÍTULO 3 O ALGORITMO DESENVOLVIDO ... 43
3.1 Algoritmo SIA Multi-objetivo ... 43
3.1.1 Inicializar População ... 44
3.1.2 Avaliar Anticorpos ... 45
3.1.3 Medidas de qualidade e Afinidade ... 45
3.1.4 Clonagem ... 51
3.1.5 Mutação ... 51
3.1.6 Avaliar Clones Mutados ... 51
3.1.7 Gravar na Memória ... 51
3.1.8 Remover Duplicatas ... 52
3.1.9 Selecionar Anticorpos ... 52
3.1.10 Suprimir Memória ... 54
3.1.11 Critério de Parada Satisfeito ... 55
3.2 Tratamento das Restrições ... 55
CAPÍTULO 4 VALIDAÇÃO DO ALGORITMO ... 56
4.1 Função f_1 de Schaffer ... 57
4.2 Função f_2 de Schaffer ... 59
4.3 Problemas ZDT ... 59
4.3.1 ZDT1 ... 61
4.3.2 ZDT2 ... 63
4.3.3 ZDT3 ... 64
4.3.4 ZDT4 ... 66
4.3.5 ZDT6 ... 68
4.3.6 Conclusões sobre os resultados das Funções sem restrição ... 69
4.4 Problemas com Restrição ... 74
4.4.1 Constr-Ex ... 75
4.4.2 TNK ... 77
4.4.3 VIGA bi-apoiada (perfil em I) ... 79
4.4.4 Treliça de duas barras ... 83
4.4.5 Viga Soldada ... 86
4.4.6 Conclusões sobre os resultados das Funções com restrição ... 91
4.5 Comentários ... 91
CAPÍTULO 5 OTIMIZAÇÃO DE ROTA DE DUTOS SUBMARINOS ... 92
5.1 Representação geométrica ... 92
5.2 Codificação no Algoritmo de Otimização ... 99
5.3 Modelagem do Problema ... 100
5.3.1 Tratamento Uni-objetivo ... 100

5.3.2 Tratamento Multi-Objetivo 102
5.4 RESTRIÇÕES 106
5.4.1 Restrição 1 - Cruzamento sobre si mesmo 106
5.4.2 Restrição 2 - Cruzamento de obstáculos 107
5.4.3 Restrição 3 - Comprimento mínimo entre duas curvas 108
5.4.4 Restrição 4 - Comprimento reto mínimo de saída (ponto A) 109
5.4.5 Restrição 5 - Comprimento reto mínimo de chegada (ponto B) 109
5.4.6 Restrição 6 - Limites da batimetria 109
5.4.7 Restrição 7 - Declividade longitudinal; 110
5.4.8 Restrição 8 - Raio de curvatura mínimo 112
CAPÍTULO 6 ESTUDO DE CASOS 114
6.1 CENÁRIOS E PARÂMETROS DO ALGORITMO 114
6.1.1 Cenários 114
6.1.2 Características Físicas e Geométricas do Duto 115
6.1.3 Carregamentos Ambientais 115
6.1.4 Parâmetros do Processo de Otimização 117
6.2 RESULTADOS 120
6.2.1 T01 – Batimetria suave em águas rasas 120
6.2.2 T03 – Batimetria suave em águas intermediárias 125
6.2.3 T05 – Batimetria acidentada em águas intermediárias 130
6.2.4 Estudo adicional em T05 136
6.3 COMENTÁRIOS 139
CAPÍTULO 7 CONCLUSÃO 140
7.1 CONSIDERAÇÕES FINAIS 140
7.2 PROPOSTA PARA TRABALHOS FUTUROS 141
REFERÊNCIAS BIBLIOGRÁFICAS 144
ANEXO A – RESULTADO PARA FUNÇÕES MATEMÁTICAS 151

ÍNDICE DE FIGURAS

Figura 1.1: Sistema conceitual para exploração de petróleo (FOCUS, 2002) 4
Figura 2.1: Exemplo de processo de decisão de compra de um automóvel. 10
Figura 2.2: Exemplo de Fronte de Pareto 11
Figura 2.3: Distribuição das soluções no fronte de pareto 12
Figura 2.4: Princípio da seleção clonal (CASTRO & ZUBEN, 2000) 22
Figura 2.5: Representação do SIA - *CLONALG* 23
Figura 2.6: Taxa de mutação como função da afinidade F^* 25
Figura 2.7: Exemplo de funcionamento do ρ_{Ad} 26
Figura 2.8: Duas metas da otimização multi-objetivo 39
Figura 2.9: Fronte de pareto com (a) boa convergência; (b) boa diversidade 39
Figura 2.10: Comparação entre frontes de pareto 40
Figura 3.1: Fluxograma do Algoritmo SIAMEP 44
Figura 3.2: Exemplo de *camadas de pareto* 46
Figura 3.3: Cálculo da *distância de agrupamento* – modificado de DEB *et al.* (2000a). 47
Figura 3.4: *Distância de agrupamento de indivíduos* – modificado de LUO *et al.* (2008) 49
Figura 4.1: gráfico de g(x) e h(x) com $-6 \le x \le 6$ 57
Figura 4.2: (a) Fronte de pareto verdadeiro; Fronte obtido com (b) T1(conj de pareto = 5) (c) T2 (conj de pareto = 30); (d) T3 (conj de pareto= 70) 58
Figura 4.3: Resultados para Schaffer F_1 em ZITZLER *et al.* (1998) para conjunto de pareto com 5, 30 e 70 soluções e resultado para algoritmo VEGA. 58
Figura 4.4: Resultados para Schaffer f_2 59
Figura 4.5: Resultados para ZDT1 (a) com SIAMEP; (b) com PAES-M 62
Figura 4.6: Resultados para ZDT2 (a) com SIAMEP; (b) com PAES-M 63
Figura 4.7: Resultados para ZDT3 (a) com SIAMEP; (b) com PAES-M 65
Figura 4.8: Resultados para ZDT4 (a) com SIAMEP; (b) com PAES-M 67
Figura 4.9: Resultados para ZDT6 (a) com SIAMEP; (b) com PAES-M 68
Figura 4.10: Resultado da medida de convergência GD para ZDT1 70
Figura 4.11: Resultado da medida de convergência GD para ZDT2 70
Figura 4.12: Resultado da medida de convergência GD para ZDT3 71
Figura 4.13: Resultado da medida de convergência GD para ZDT4 71
Figura 4.14: Resultado da medida de convergência GD para ZDT6 71
Figura 4.15: Resultado da medida de distribuição Δ para ZDT1. 73
Figura 4.16: Resultado da medida de distribuição Δ para ZDT2. 73
Figura 4.17: Resultado da medida de distribuição Δ para ZDT3. 73
Figura 4.18: Resultado da medida de distribuição Δ para ZDT4. 74
Figura 4.19: Resultado da medida de distribuição Δ para ZDT6. 74
Figura 4.20: Resultados para CONSTR-EX: (a) SIAMEP; (b) PAES-M; (c) NSGA-II (DEB *et al.*, 2002); e (d) BCMOA (GUZMÁM *et al.*, 2010) 76

Figura 4.21: Resultados para TNK: (a) com SIAMEP; (b) com PAES-M; (c) VAIS (FRESCHI & REPETTO, 2005); e (d) NSGA-II (DEB *et al.*, 2002);..... 78
Figura 4.22: Viga Bi-apoiada em seção I..... 79
Figura 4.23: Resultado comparativo para o problema da Viga bi-apoiada..... 81
Figura 4.24: Resultado comparativo para o problema da Viga bi-apoiada..... 81
Figura 4.25: Treliça de duas barras..... 84
Figura 4.26: Resultados Treliça de duas barras (DEB *et al.*, 2000b)..... 85
Figura 4.27: Resultados Treliça de duas barras..... 85
Figura 4.28: Viga soldada..... 87
Figura 4.29: Resultados para Viga Soldada (DEB *et al.*, 2000b)..... 89
Figura 4.30: Resultados para Viga Soldada..... 89
Figura 5.1: Elementos do eixo de uma rodovia (LEE, 2000)..... 92
Figura 5.2: Curva circular simples..... 93
Figura 5.3: Representação da rota base..... 94
Figura 5.4: Representação de $PI's$ de uma rota, ainda sem curvas..... 94
Figura 5.5: Representação de trechos em curva..... 95
Figura 5.6: Distribuição de pontos base sobre a reta *AB*..... 96
Figura 5.7: Definição de PI..... 96
Figura 5.8: Cálculo dos pontos sobre a curva..... 97
Figura 5.9: Cálculo do peso de lastro requerido para o duto ficar estável..... 103
Figura 5.10: Forças atuantes no duto, modificado de BAIOCO (2011)..... 104
Figura 5.11: Restrição de cruzamento sobre si mesmo..... 106
Figura 5.12: Seção de Duto Esquemática..... 113
Figura 6.1: Batimetria (a) Acidentada, (b) Suave..... 114
Figura 6.2: Velocidade da corrente próxima ao leito marinho (m/s)..... 116
Figura 6.3: Altura significativa onda H_s..... 117
Figura 6.4: Comprimento (F_1) x Peso total requerido (F_2) para T01..... 120
Figura 6.5: Rotas geradas T01 – todas..... 122
Figura 6.6: T01 – grupo com 5 soluções (visualizando peso requerido)..... 122
Figura 6.7: Conjunto com as cinco soluções T01, uma a uma..... 123
Figura 6.8: Rota gerada T01 em BAIOCO (2011) pelo peso requerido..... 124
Figura 6.9: Comprimento (F_1) x Peso total requerido (F_2) para T03..... 125
Figura 6.10: Rotas geradas T03 – todas..... 127
Figura 6.11: T03 – grupo com 5 soluções (visualizando peso requerido)..... 127
Figura 6.12: Conjunto com as cinco soluções T03, uma a uma..... 128
Figura 6.13: Rota gerada T03 em BAIOCO (2011) pelo peso requerido..... 129
Figura 6.14: Comprimento (F_1) x Peso total requerido (F_2) para T05..... 130
Figura 6.15: Rotas geradas T05 – todas..... 132
Figura 6.16: T05 – grupo com 5 soluções (visualizando peso requerido)..... 132
Figura 6.17: T05 – grupo com 5 soluções (visualizando declividade longitudinal)..... 133

Figura 6.18: Conjunto com as cinco soluções T05, uma a uma 134
Figura 6.19: Rota gerada T05 em BAIOCO (2011) pelo peso requerido 135
Figura 6.20: Comprimento (F_1) x Peso total requerido (F_2) para T05 – Extra 136
Figura 6.21: T05 – Extra, com todas as 40 rotas geradas 137
Figura 6.16: T05 – Extra, grupo com 5 soluções (visualizando peso requerido) 138
Figura 6.16: T05 – Extra, grupo com 5 soluções (visualizando declividade longitudinal) 138

ÍNDICE DE TABELAS

Tabela 1.1: Tratamento x Objetivos 6
Tabela 4.1: Configurações adotadas nos teste de Schaffer's f_1 56
Tabela 4.2: Configurações do algoritmo para os testes ZDT 60
Tabela 4.3: Funções ZDT para testes (todas para minimização) 60
Tabela 4.4: Resultados comparativos ZDT1 62
Tabela 4.5: Resultados comparativos ZDT2 64
Tabela 4.6: Resultados comparativos ZDT3 65
Tabela 4.7: Resultados comparativos ZDT4 67
Tabela 4.8: Resultados comparativos ZDT6 69
Tabela 4.9: Resultados de GD: média (1ª linha) e desvio padrão (2ª linha) 69
Tabela 4.10: Resultados de Δ: Média (1ª linha) e desvio padrão (2ª linha) 72
Tabela 4.11: Configurações do algoritmo para alguns dos testes com Restrição 75
Tabela 4.12: Resultados comparativos CONSTR-EX 77
Tabela 4.13: Resultados comparativos TNK 79
Tabela 4.14: Resultados da viga bi-apoiada para COELLO & CHRISTIANSEN (1998) 82
Tabela 4.15: Valores obtidos pelo SIAMEP 40. 82
Tabela 4.16: Valores obtidos por CASTRO, R. E. (2001). 83
Tabela 4.17: Resultados Treliça de duas Barras – SIAMEP 40 85
Tabela 4.18: Resultados Viga Soldada para CASTRO, R. E. (2001) 89
Tabela 4.19: Resultados Viga Soldada para SIAMEP 40 90
Tabela 6.1: Profundidade das extremidades A e B da rota 115
Tabela 6.2: Propriedades físicas do solo 115
Tabela 6.3: Propriedades físicas e geométricas do duto 115
Tabela 6.4: Velocidade da corrente próxima ao leito marinho (m/s) 116
Tabela 6.5: Período de pico da onda T_P 116
Tabela 6.6: Altura significativa onda H_s 117
Tabela 6.7: Parâmetros do algoritmo SIAMEP 118
Tabela 6.8: Parâmetros da modelagem 118
Tabela 6.9: Penalizações utilizadas para todos os testes 119
Tabela 6.10: Resultado para T01 121
Tabela 6.11: Resultados de BAIOCO (2011) para T01 121
Tabela 6.12: Resultado para T03 125
Tabela 6.13: Resultados de BAIOCO (2011) para T03 126
Tabela 6.14: Resultado para T05 130
Tabela 6.15: Resultados de BAIOCO (2011) para T05 131
Tabela 6.16: Resultado para T05 - Extra 136

LISTA DE SÍMBOLOS

Símbolo	*Significado*
A	Área da seção transversal da viga
a_1	Área da seção transversal do segmento AC em m²
a_2	Área da seção transversal do segmento BC em m²
AC	Ângulo central
$afinidade_{ip}$	Afinidade da solução *ip* na população
A_T	Área da seção transversal do duto
B	Largura das mesas
b	Largura da viga
C	Centro do círculo *C*
$camada_{ip}$	Camada da solução *ip*
CD	*Crowding Distance* – distância de agrupamento
CD_1	CD da solução da borda do intervalo (parte inicial)
CD_{ip}	CD de cada solução *ip* na população
$CD_{n_{indiv}}$	CD da solução da borda do intervalo (parte final)
CR	Taxa de cruzamento pré-definida – Mutação diferencial modificada
D	Altura total da viga
De	Desenvolvimento (comprimento do arco desenvolvido entre PC e PT)
dm_{iq}	Representa o mínimo somatório do valor absoluto da diferença dos valores das funções objetivo entre a solução *iq* e qualquer outra solução em *Q*
$\overline{dm}$	Representa a média dos valores de dm_{iq}
d_{iq}	Para $p=2$, d_{iq} é distância euclidiana entre uma solução *iq* em *Q* e o indivíduo mais próximo de *P**
D_{iq}	Pode ser qualquer métrica de distância (distância Euclidiana ou distância de aglomeramento, por exemplo)
$\overline{D}$	Representa a média dos valores de D_{iq}
D_i^e	É a distância entre soluções extremas de *P** e *Q* correspondendo à *i-ésima* função objetivo
DCD	Dynamic Crowding Distance – distância de agrupamento dinâmica
DCD_{ip}	DCD das soluções *ip* na população
$Decliv_{ino}$	Representa a declividade longitudinal calculada entre o ponto *ino* e um ponto consecutivo na malha do duto
$decliv_{max}$	Valor limite para a maior declividade longitudinal aceita
$densidade_{ip}$	Densidade em torno de cada solução *i*
$dist_{AB}$	Distância entre os pontos A e B da rota a ser otimizada
$discrW$	Incremento aplicado ao peso de lastro no duto para a conseguinte verificação de estabilidade
dp_{ip}	Distância média aos pontos mais próximos de cada lado do ponto particular *ip* ao longo de cada objetivo
E	Módulo de Young, ou módulo de elasticidade
ED	Espaço de decisão
EO	Espaço de objetivos
ER	Taxa de erro (métrica de desempenho)
F	Número real e constante pertencente ao intervalo [0, 2] que controla a amplitude da diferença ponderada – mutação diferencial modificada
F^*	É a afinidade normalizada (entre 0 e 1) – mutação somática
F_h	Carga horizontal
F_{Hor}	Representa a tensão residual no fundo – valor pré-definido igual à tração de lançamento do duto na BGL

$f_i(\boldsymbol{x})$	Função objetivo *i*
$f_i^{*(ie)}$	Valor do *i-ésimo* objetivo da *ie-ésima* solução de *P**
$f_i^{(iq)}$	Valor do *i-ésimo* objetivo da *iq-esima* solução de *Q*
$f_i^{\max}$	Refere-se ao máximo valor de função objetivo encontrado para os objetivos de $1...N_{ob}$
$f_i^{\min}$	Refere-se ao mínimo valor de função objetivo encontrado para os objetivos de $1...N_{ob}$
$f_i(ip)$	O valor do *i-ésimo* objetivo do *ip-ésimo* indivíduo
$F_{mín}^*$	Valor mínimo da afinidade para a função objetivo normalizada
$F_{máx}^*$	Valor máximo da afinidade para a função objetivo normalizada
F_v	Carga vertical
G	Módulo de torção
GD	Distância geracional (métrica de desempenho)
g_j	Representa o valor de violação de cada restrição *j*
$g_j(\boldsymbol{x})$	Função de restrição de comportamento
h	Espessura da solda em polegadas
H	Ponto que divide a reta $\overline{PC\ PT}$ em dois segmentos de igual tamanho
$h_k(\boldsymbol{x})$	Função de restrição de igualdade
$\overline{HPI}$	Comprimento da reta ligando o ponto H ao PI
H_s	Altura significativa da onda
i	Contador do número de objetivos $i = 1,2, ..., N_{ob}$
I	Ângulo de deflexão
$ia1$	Índice do anticorpo 1 selecionado para a mutação diferencial
$ia2$	Índice do anticorpo 2 selecionado para a mutação diferencial
$ia3$	Índice do anticorpo 3 selecionado para a mutação diferencial
ib	Contador do número de pontos base na curva
ic	Contador de clones – um número inteiro entre 1 e o número de clones
ie	Contador para soluções do conjunto externo
iel	Contador do número de elementos
$iGer$	Contador de número da geração
imp	Peso para cada nível de dificuldade de cruzamento
ino	Contador do número de nós na malha do duto
ip	Contador de indivíduo na população atual
iq	Contador do número de soluções no conjunto *Q*
$Iter$	Número da iteração atual
Iy	Momento de inércia em relação ao eixo y
Iz	Momento de inércia em relação ao eixo z
j	Contador de número de restrições $j = 1,2, ..., N_{re}$
k	Contador do número de restrições de igualdade $k = 1,2, ..., N_{ig}$
k_{duto}	Ponderação relativa ao comprimento do duto
k_{lastro}	Ponderação relativa ao lastro
L	Comprimento da viga
l	Comprimento da solda em polegadas
$L_{elem}(iel)$	Comprimento de cada trecho de rota analisado
Lim_{iter}	Limite de iterações para o cálculo do peso requerido
$L_{medidoCurvas}$	Comprimento medido entre curvas
$L_{medidoA_cur}$	Comprimento medido entre o ponto A e o início da primeira curva

$L_{medidoCur_B}$	Comprimento medido entre o final da última curva e o ponto B
$L_{\min Curvas}$	Comprimento mínimo entre curvas
$L_{\min A_cur}$	Comprimento mínimo entre o ponto A e o início da primeira curva
$L_{\min Cur_B}$	Comprimento mínimo entre o final da última curva e o ponto B
L_{rota}	Comprimento da rota gerada entre os pontos A e B
L_v	Comprimento Livre
m	Contador de variáveis livres $m = 1,2, ..., N_{var}$
mk_j	Peso atribuído a cada restrição j
$N(0,\sigma)$	Vetor de números aleatórios independentes que segue uma distribuição gaussiana.
N_{ab}	Número de anticorpos do sistema imunológico artificial
N_{clones}	Número de clones do sistema imunológico artificial
$ncruz_1$	Número de interseções sobre si mesmo
$ncruz_2$	Número de interseções com obstáculos
N_{dom}	Número de indivíduos na população dominados pelo indivíduo n_e
$nDom_{ip}$	Número de soluções dominadas pela solução ip
N_{elem}	Número de elementos que compõe a rota
N_{iq}	Número de restrições de igualdade
n_{indiv}	Número de indivíduos na população
$nNdom$	Número de soluções não-dominadas na população
N_{nos}	Número de nós na malha do duto
N_{ob}	Número de objetivos
N_{pb}	Número de pontos de base
NQ	Número de soluções no conjunto Q
N_{re}	Número de restrições
$nTotDom$	Número total de soluções dominadas por todas as soluções
N_{var}	Número de variáveis livres
O	Coordenadas do centro da curva
OD	Diâmetro externo do duto
p	Constante – métrica de desempenho
P^*	Conjunto de soluções pareto-ótimas
pb_{ib}	Pontos Base ib
PC	Ponto de curva
$P_c(x)$	Esforço resistente ao longo da direção t
$penal_1$	Penalidade de cruzamento sobre si mesmo
$penal_2$	Penalidade de cruzamento de obstáculos
$penal_3$	Penalidade de comprimento mínimo entre duas curvas
$penal_4$	Penalidade de comprimento reto mínimo de saída (ponto A)
$penal_5$	Penalidade de comprimento reto mínimo de chegada (ponto B)
$penal_6$	Penalidade para os limites da batimetria
$penal_7$	Penalidade relativa à declividade longitudinal do duto
$penal_8$	Penalidade para o raio de curvatura
$penal_9$	Penalidade para estabilidade lateral do duto
$penal_{10}$	Penalidade para estabilidade vertical do duto
$penal_j$	Representa o valor de cada penalidade j
PI	Ponto de interseção
PI_{ib}	Ponto de interseção ib
PT	Ponto de tangente
Q	Conjunto de soluções
q, q_1, q_2	Equações parametrizadas em Y
R	Raio da curva

R_{ino}	Raio de curvatura calculado para cada solução em análise
r_m	Um valor aleatório gerado para o parâmetro *m* – mutação diferencial modificada
R_{min}	Raio mínimo de curvatura requerido para o duto
s	Parâmetro da equação de parametrização
SF	Fator de segurança mínimo para o duto (neste trabalho, igual a 1,1)
s_{ip}	Força de uma solução *ip*
SP	Espaçamento (métrica de desempenho)
STD	Desvio padrão
s_u	Resistência não-drenada do solo
t	Altura da viga
T	Tangente externa ou exterior
t_c	Espessura da camada de concreto do duto
t_{dom}	Tamanho do conjunto de comparação
T_f	Espessura das mesas
t_p	Espessura do revestimento anti-corrosivo do duto
T_P	Período de pico da onda
t_s	Espessura da parede de aço do duto
T_w	Espessura da viga
u, u_1, u_2	Equações parametrizadas em X
V_{ip}	Representa a variância de CD_i dos indivíduos
W_e	Empuxo do duto
$W_{req}(iel)$	Peso requerido para estabilizar determinado elemento *iel* da rota
$\overline{W_{req}}$	Peso de lastro do duto
$W_{reqTotal}$	Representa o peso total de lastro da rota analisada
W_{sub}	Peso submerso do duto
W_T	Representa o peso total do duto
W_y	Módulo resistente da seção em relação ao eixo y
W_z	Módulo resistente da seção em relação ao eixo z
$v'_{ic,m}$	Valor que representa cada parâmetro *m* de um clone *ic* modificado em uma dada geração *iGer+1*
$\boldsymbol{v}'_{ic}$	Vetor que representa os parâmetros de um clone *ic* modificado em uma dada geração *iGer+1*
$\boldsymbol{x}$	Vetor de variáveis de decisão $\boldsymbol{x} = (x_1, x_2, ..., x_{N_{\text{var}}})^T$
$\boldsymbol{x}'$	Vetor de variáveis modificadas $\boldsymbol{x}' = (x'_1, x'_2, ..., x'_{N_{\text{var}}})^T$
x_c	Coordenada X do centro do círculo
$\boldsymbol{x}_{ia1}$	Vetor que representa os parâmetros do anticorpo 1
$\boldsymbol{x}_{ia2}$	Vetor que representa os parâmetros do anticorpo 2
$\boldsymbol{x}_{ia3}$	Vetor que representa os parâmetros do anticorpo 3
$x'_{ic,m}$	Valor modificado de um parâmetro *m* de um clone *ic* numa geração *iGer +1*
x_H	Coordenada X do ponto *H*
x_m	Variável livre
$x_m^{(L)}$	Limite superior de cada variável livre *m*
$x_m^{(U)}$	Limite inferior de cada variável livre *m*
$x_{PI_{ib}}$	Coordenada X do PI_{ib}
$y_{PI_{ib}}$	Coordenada Y do PI_{ib}
$x_{pb_{ib}}$	Coordenada X do ponto base pb_{ib}

$y_{pb_{ib}}$	Coordenada Y do ponto base pb_{ib}
$x_{PC_{ib}}$	Coordenada X do PC_{ib}
$y_{PC_{ib}}$	Coordenada Y do PC_{ib}
$x_{PT_{ib}}$	Coordenada X do PT_{ib}
$y_{PT_{ib}}$	Coordenada Y do PT_{ib}
y	Distância vertical entre B e C em metros
y_c	Coordenada Y do centro do círculo
y_H	Coordenada Y do ponto H
$\boldsymbol{\beta}$	Vetor de Parâmetros da mutação somática $\boldsymbol{\beta} = \left(\beta_1, \beta_2, ..., \beta_{N_{\text{var}}}\right)^T$
β_i	Representa o valor do parâmetro β para cada parâmetro m do indivíduo
$\beta_{i\text{-}final}$	Representa o valor final do parâmetro β para cada parâmetro m do indivíduo
γ_s	Peso seco do solo
γ'_s	Peso molhado do solo
γ_{sc_y}	Um fator de segurança calculado para o duto
γ_{sc_y}	Fator de estabilidade lateral
$\gamma_{sc_y,ino}$	Estabilidade lateral referente ao nó *ino* do duto em análise
γ_{sc_z}	Fator de estabilidade vertical
$\gamma_{sc_z,ino}$	Estabilidade vertical referente ao nó *i* do duto em análise
Δ	Espalhamento (métrica de desempenho)
$\delta(x)$	Deslocamento estático máximo
$\delta_{máx}$	Limite máximo de deslocamento da viga
ε	Tolerância a cada restrição de igualdade
η_{ib}	Afastamento de cada ponto base *ib*
θ_{ib}	Ângulo do *ib-ésimo* afastamento com o eixo X
λ	Número de indivíduos gerados por recombinação e mutação
μ	Número de indivíduos da geração anterior
μ_{ag}	Peso específico da água
μ_{solo}	Coeficiente de fricção (ou atrito) do solo marinho – valor pré-definido função do solo em que será instalado o duto
ρ	Parâmetro que controla a suavidade da exponencial inversa – mutação somática
ρ_{Ad}	Valor adpatativo para o parâmetro ρ – mutação somática
ρ_c	Peso específico do concreto
$\rho_{máx}$	Valor máximo para ρ_{Ad} – mutação somática
$\rho_{mín}$	Valor mínimo para ρ_{Ad} – mutação somática
ρ_p	Peso específico do revestimento
ρ_s	Peso específico do aço
$\boldsymbol{\sigma}$	Vetor de desvios padrão $\boldsymbol{\sigma} = \left(\sigma_1, \sigma_2, ..., \sigma_{N_{\text{var}}}\right)^T$
$\sigma(x)$	Tensão normal
σ_{AC}	Tensão na barra AC em *kPa*
σ_{BC}	Tensão na barra BC em *kPa*
$\sigma_{máx}$	Tensão máxima admissível
$\tau(x)$	Tensão cisalhante
$\tau_{máx}$	Tensão cisalhante máxima

Capítulo 1 Introdução

1.1 Contexto e Motivação

Equation Chapter 1 Section 1Num mundo cada vez mais competitivo, a busca por reduzir custos e reduzir tempo no desenvolvimento de projetos, e ainda, apresentar melhores resultados, vem sinalizando cada vez mais o uso de ferramentas de otimização.

Problemas do mundo real envolvem, muitas das vezes, a otimização de múltiplos objetivos simultaneamente e objetivos as vezes conflitantes. Freqüentemente não há apenas uma solução para o problema, mas um conjunto de soluções possíveis. De uma maneira geral uma solução é considerada ótima quando não há no espaço de busca, soluções melhores que ela quando se consideram todos os objetivos (ZITZLER *et al.*, 1998).

A solução usual a estes problemas é a combinação dos múltiplos objetivos em um único critério a ser otimizado, computado de acordo com uma única função que mede a qualidade desta solução (FONSECA & FLEMING, 1993). Com isto o problema multi-objetivo é transformado em uni-objetivo. Para esta função ser desenvolvida, é necessário haver conhecimento prévio do problema a ser abordado e, na maioria das vezes, são necessários inúmeros testes subseqüentes para calibrar os pesos dos critérios escolhidos para decisão. Com isso, esta tarefa torna-se difícil, e pode introduzir erros na solução do problema desde a fase de modelagem.

A forma mais completa de tratamento a estes problemas do mundo real seria o tratamento como múltiplos objetivos, onde um conjunto de soluções possíveis (ou viáveis) seria fornecido ao final do processo de otimização. Este conjunto de soluções pode ser analisado pelo tomador de decisão, visando melhor atender seus objetivos. Desta forma, ao invés de se obter apenas uma única solução ótima ao problema, geralmente formada por uma combinação dos critérios numa função de avaliação, seriam geradas múltiplas soluções avaliadas segundo cada um dos objetivos do problema, que são todas satisfatórias ao problema de otimização. Este conjunto de soluções recebe o nome de *conjunto ótimo de pareto* ou *fronte de pareto*.

Alguns exemplos de problemas de otimização multi-objetivos são: a otimização de uma estrutura em treliça para uma passarela de pedestres, onde se deseja reduzir a seção transversal das barras para reduzir o custo de metal, e deseja-se reduzir o deslocamento da estrutura para aumentar o conforto do pedestre ao passar na passarela.

Neste caso os objetivos são antagônicos – quando se reduz a área das barras, o deslocamento da estrutura aumenta.

Outro exemplo seria o projeto de aviões onde múltiplos objetivos conflitantes coexistem como: redução do peso do avião, aumento do volume útil, turbinas com menor consumo de combustível, asas e corpo do avião com melhor aerodinâmica, estrutura mais resistente e mais leve, envolvendo diferentes materiais e diferentes disciplinas como aerodinâmica, termodinâmica, física, mecânica e materiais.

A otimização multi-objetivo permite um grau maior de flexibilidade em relação à otimização uni-objetivo. É certo que este ganho de flexibilidade vem acompanhado de uma conseqüência: a necessidade de se escolher *a posteriori* qual é a melhor solução tendo como base uma relação de troca (*tradeoff*) onde cada solução é única e atende aos objetivos do problema. O tomador de decisão ficará com a responsabilidade desta decisão (COLLETTE & SIARRY, 2003).

No contexto de otimização utilizando-se algoritmos bio-inspirados existem múltiplas opções de algoritmos: Algoritmos Genéticos, Sistema Imunológico Artificial, Enxame de Partículas, Colônia de Formigas, Algoritmos Evolucionários, Evolução Diferencial, etc. Neste trabalho optou-se pelo Sistema Imunológico Artificial devido à sua capacidade de resolução dos problemas com um algoritmo simples, e capaz de encontrar o ótimo com pequeno número de avaliações conforme visto em VIEIRA (2009). Com isso pode-se obter soluções que satisfaçam às características perseguidas com um bom desempenho computacional.

Este trabalho trata do desenvolvimento de um algoritmo de otimização multi-objetivo, e sua implementação em uma ferramenta de otimização de rotas de dutos submarinos. A escolha por abordar um problema de engenharia de petróleo se dá pela percepção do cenário de crescente demanda de petróleo e pelo fato da sua exploração se dar em águas cada vez mais profundas, avançando em complexidade e custo, sendo a teoria de otimização de grande aplicação neste caso. A motivação para o uso do algoritmo multi-objetivo é que este fornece maior flexibilidade, sendo mais abrangente, uma vez que permite ao tomador de decisão escolher qual a melhor solução dentro de um grupo de soluções viáveis.

1.2 OBJETIVOS DO TRABALHO

Um dos objetivos deste trabalho é desenvolver um algoritmo multi-objetivo simples, inspirado no Sistema Imunológico Artificial (SIA), capaz de resolver problemas multi-objetivos com e sem restrições e que seja baseado no *princípio da seleção clonal* (Algoritmo *CLONALG*) (CASTRO, L. N., 2001). De forma a validar o algoritmo

desenvolvido, este será aplicado a problemas de funções matemáticas de *benchmark* (com e sem restrição) que apresentem diferentes níveis de dificuldades, demonstrando sua capacidade de obter resultados compatíveis com os algoritmos apresentados no estado da arte. Ainda para validar o algoritmo, será apresentada sua aplicação a problemas clássicos de engenharia estrutural.

É também um objetivo deste trabalho a implementação do algoritmo desenvolvido em uma ferramenta de otimização de rotas de dutos submarinos, que é um problema real de engenharia de petróleo que vem se mostrando cada vez mais relevante em virtude da crescente demanda por produtos derivados de petróleo e pelos custos elevados de produção. A ferramenta, denominada *OtimRota*, vem sendo desenvolvida pela equipe do Laboratório de Métodos Computacionais e Sistemas *Offshore* (LAMCSO) da COPPE/ UFRJ, como resultado de um projeto cooperativo realizado em parceria com o Centro de Pesquisas - CENPES da Petrobras (JACOB *et al.*, 2011).

Como será visto a seguir, uma contribuição desta tese é o desenvolvimento de uma mutação híbrida para solucionar o problema da homogeneidade da distribuição das soluções na fronteira de pareto associado ao algoritmo multi-objetivo. A mutação desenvolvida inclui características da mutação da Evolução Diferencial e da mutação somática do Sistema Imunológico Artificial. Pode-se dizer que outra característica inovadora do presente trabalho é a associação do algoritmo multi-objetivo à modelagem do problema de otimização de rotas, onde combinam-se os objetivos (que podem ser antagônicos) de minimizar o comprimento da rota, e minimizar o peso total de lastro requerido para garantir a estabilidade hidrodinâmica do duto apoiado no leito marinho. Com isto, a ferramenta *OtimRota* poderá ser empregada no estabelecimento de metodologias inovadoras de projeto destes importantes sistemas de produção de petróleo *offshore*.

1.3 ROTA DE DUTOS SUBMARINOS

1.3.1 Introdução

Pode-se dizer que a produção de petróleo caminha para um cenário composto por águas cada vez mais profundas com sistemas cada vez mais complexos em virtude de elevadas pressões e temperaturas, e ambientes altamente corrosivos. Nestes sistemas, a otimização passa a ser de importância estratégica para as empresas que atuam no setor.

A exploração e explotação de petróleo no mar envolve um conjunto de equipamentos conhecidos como “Sistemas *Offshore*” e compreendem basicamente quatro grupos: O casco, as linhas, os equipamentos submarinos e os poços. O desenvolvimento das instalações de produção de óleo e gás inclui a instalação de dutos

de óleo e gás apoiados no leito marinho. Estes dutos conectam pontos relevantes do sistema. No caso de um leito marinho com topologia complexa, não é fácil determinar rotas viáveis ou ainda, encontrar a melhor rota viável.

A Figura 1.1 a seguir ilustra um arranjo conceitual de exploração que apresenta dutos submarinos conectando poços a linhas de elevação até a superfície.

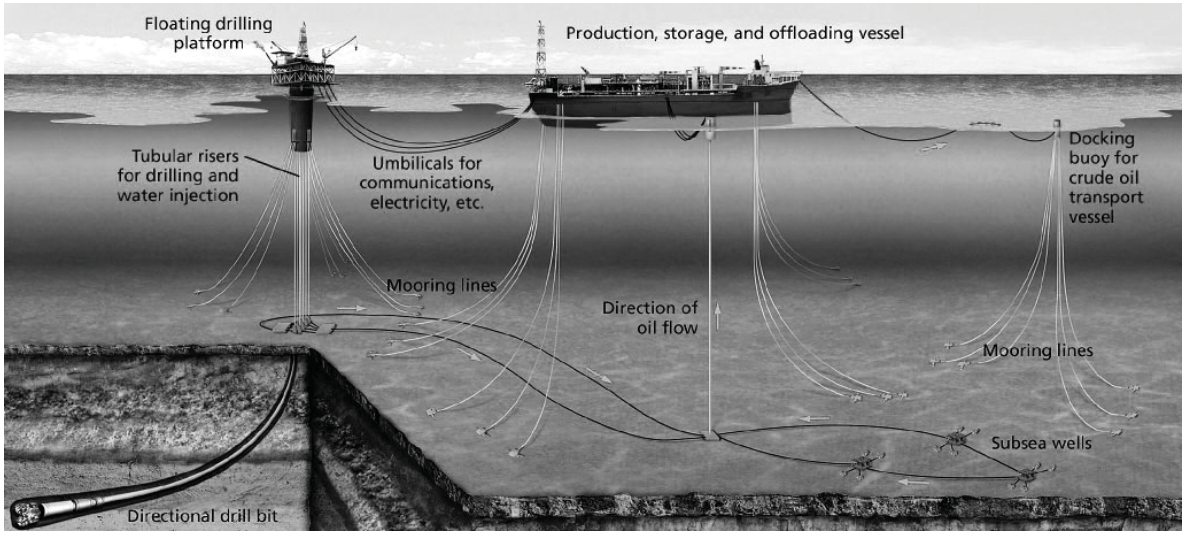


Figura 1.1: Sistema conceitual para exploração de petróleo (FOCUS, 2002)

Neste trabalho, serão abordados aspectos do desenvolvimento de rotas de dutos submarinos levando-se em conta as exigências técnicas, ambientais e operacionais. Tradicionalmente, o planejamento da rota de dutos submarinos é feito manualmente pelo engenheiro, através da inspeção expedita da batimetria de fundo e obstáculos. Eventualmente a avaliação de uma determinada rota é efetuada através de ferramentas de análise, mas de qualquer forma o processo é altamente dependente da experiência do projetista.

Nesta tese, propõe-se tratar a parte de gerações de soluções de rota de maneira automática através de um programa de otimização baseado em um algoritmo multi-objetivo bio-inspirado.

Em tais procedimentos de otimização, cada configuração de rota gerada deve ser avaliada, para determinar sua aptidão (*ou afinidade*) em termos dos critérios incorporados na montagem das funções-objetivo, que leva em consideração todos os aspectos relevantes na escolha de uma rota.

1.3.2 Histórico

Tendo em vista dar um balizamento ao trabalho, serão expostos alguns trabalhos do estado da arte em otimização de rota de dutos.

MEISINGSET *et al.* (2004) desenvolvem estudo visando localizar rotas viáveis possuindo o menor custo de construção. Este trabalho apresenta o desenvolvimento de uma ferramenta para a realização de otimização de rotas de dutos que considera na sua formulação o custo do material para construção do duto e custo de intervenções no processo de instalação do duto, obtendo assim, o duto com o mínimo custo. O algoritmo utilizado foi o algoritmo do cozimento simulado (*simulated annealing*).

Em FERNANDES *et al.* (2008) é apresentado o desenvolvimento de uma função objetivo, que poderia ser associada a algoritmos evolutivos para determinar a rota ótima de dutos em aplicações *offshore* (fora da costa). Tendo em vista que os dutos *offshore* repousam sobre batimetria acidentada, o projeto de otimização levaria em conta dados de batimetria, características do solo, efeitos da instalação e propriedades físicas do duto. Além dessas características outras deveriam ser levadas em consideração como, o comprimento total do duto, quantidade e localização de vãos livres deveriam ser identificadas, raio mínimo de curvatura depois de instalado, etc. Foi realizado um estudo de sensibilidade para identificar os dados mais críticos a serem considerados durante a análise de otimização, objetivando reduzir o custo computacional, evitar interpretações errôneas da rotas, e minimizar o custo de projeto e instalação do duto.

Em VIEIRA *et al.* (2010) são apresentados os passos no desenvolvimento de uma ferramenta computacional para síntese e o otimização de rotas de dutos submarinos. A ferramenta é baseada em algoritmos evolutivos e incorpora critérios relacionados ao projeto de rotas de duto como comprimento do duto, características geofísicas e geográficas de dados de batimetria, incluindo definição de obstáculos e regiões a serem evitadas; número, comprimento e localização de vãos livres; análise e verificação de estabilidade do duto instalado sobre carregamento hidrostático e hidrodinâmico; além de outros aspectos relacionados ao comportamento estrutural do duto. Neste trabalho é dada uma especial atenção à representação geométrica da rota, e aos termos da função objetivo associadas a um processo preliminar de otimização.

LIMA JUNIOR (2010) estudou métodos de análise e critérios de verificação de estabilidade de dutos submarinos apoiados no leito marinho, sob ação de carregamentos hidrodinâmicos de onda e correnteza através de modelo bi-dimensional.

No trabalho de BAIOCO (2011), o problema da otimização de rota de dutos submarino foi abordado de maneira uni-objetivo considerando-se critérios de estabilidade hidrodinâmica. A otimização foi realizada empregando Algoritmos Genéticos e o estudo de caso conduzido mostrou que a ferramenta é capaz de apresentar resultados compatíveis com o esperado por especialistas.

Em LUCENA (2011) é verificada a eficácia de um método de penalização adaptativa aplicada à meta-heurística Sistemas Imunológicos Artificiais, para otimizar uma rota de dutos submarinos, sujeita às restrições de um ambiente real.

1.3.3 Metodologia

No presente trabalho a otimização de rota de dutos será tratada de maneira multi-objetivo dando continuidade à linha de pesquisa apresentada por e LIMA JUNIOR (2010), BAIOCO (2011) e LUCENA (2011). A tabela a seguir ilustra os objetivos a minimizar em cada uma dos casos.

Tabela 1.1: Tratamento x Objetivos

TRATAMENTO	OBJETIVO(S)
Uni-Objetivo	$f =$ Minimizar comprimento
Multi-Objetivo	$f_1 =$ Minimizar comprimento $f_2 =$ Minimizar peso total requerido de lastro

Ainda outros objetivos poderiam ser elencados: $f_3 =$ minimizar o número de vãos livres; $f_4 =$ minimizar a espessura das paredes, etc... No entanto, a utilização de mais de dois objetivos pode complicar ainda mais a análise dos resultados e sua comparação.

Tradicionalmente, em uma tarefa de seleção de rota, os seguintes critérios são levados em consideração:

1. Minimizar o comprimento da rota, minimizando-se a quantidade de material;
2. Minimizar o número de vão livres e quantidade de trabalhos de intervenção para correção destes vãos. Dependendo das condições de correnteza, um vão livre pode levar o duto a um estado de vibrações grandes. Para se evitar isto, os comprimentos dos vãos devem ser controlados;
3. Evitar passar por determinadas áreas devido à existência de outros equipamentos de infra-estrutura e áreas geotecnicamente instáveis;
4. Minimizar a inclinação transversal ao duto para garantia de estabilidade lateral durante instalação e operação;
5. Raio de curvatura horizontal deve ser superior a um limite mínimo. Também são requeridos trechos de reta entre trechos de curva.
6. Minimizar declividade longitudinal para se evitar a variação de pressão na operação do duto reduzindo o risco de bloqueio parcial ou total da linha.

Alguns destes aspectos dependem do comportamento estrutural do duto, sob ação dos carregamentos hidrostáticos e hidrodinâmicos. Em particular, os critérios atualmente adotados para a seleção de rota não incluem a verificação de estabilidade. Esta verificação vem sendo feita *a posteriori*, podendo levar a dutos com pesos submersos elevados, e em alguns casos pode se tornar inviável a instalação do duto.

As questões relativas à estabilidade de dutos apoiados no leito marinho foram estudadas por LIMA JUNIOR (2010) e BAIOCO (2011) e serão aqui apresentadas em forma de referência apenas. Além destas referências, temos: VERITEC RP – E305 (1998); PRCI (2002); DNV - RP - F109 (2007). Como mencionado anteriormente, no trabalho de BAIOCO (2011), o problema da otimização de rota de dutos submarino foi abordado de maneira uni-objetivo; Nesta Tese, os critérios de estabilidade serão incorporados no procedimento de otimização baseado no algoritmo multi-objetivo desenvolvido, referido como SIAMEP - Sistema Imunológico Artificial com Memória Externa de Pareto.

1.4 ORGANIZAÇÃO DA TESE

O trabalho está assim organizado: no *segundo* capítulo será apresentada a formulação matemática, os conceitos básicos para problemas de otimização multi-objetivos e o estado da arte na área de otimização multi-objetivo contendo os seguintes algoritmos: Algoritmos Genéticos, Estratégias Evolutivas e Sistemas Imunológicos Artificiais. Ainda no segundo capítulo são apresentadas as medidas de desempenhos dos algoritmos multi-objetivos quando utilizando frontes de pareto conhecidos.

No *terceiro* capítulo apresenta-se o algoritmo multi-objetivo desenvolvido - SIAMEP; no *quarto* capítulo são apresentados os testes de validação do algoritmo desenvolvido baseados em comparação com os resultados obtidos na bibliografia apresentada.

Mais adiante, no *quinto* capítulo, são abordados os diferentes aspectos relacionados à modelagem do problema de otimização de rota de dutos submarinos na ferramenta *OtimRota*, associada ao algoritmo desenvolvido; no *sexto* capítulo, são apresentados os resultados dos estudos de caso aplicando a ferramenta ao problema da otimização de rota de dutos.

Finalmente, no sétimo capítulo, os comentários finais são apresentados juntos com as sugestões de trabalhos futuros

Capítulo 2 OTIMIZAÇÃO MULTI-OBJETIVO

Equation Chapter (Next) Section 1Neste capítulo serão apresentadas noções básicas de otimização multi-objetivo. A abordagem completa relativa à otimização multi-objetivo pode ser encontrada em DEB (2001). Este capítulo foi dividido em três seções: a primeira seção apresenta os principais conceitos de otimização multi-objetivo, a segunda o estado da arte na área de otimização multi-objetivo, e a terceira trata das medidas de desempenho dos algoritmos multi-objetivos.

2.1 CONCEITOS BÁSICOS

Nesta seção, será apresentada a formulação matemática para um problema multi-objetivo geral, e a definição do que são soluções pareto-ótimas.

2.1.1 Formulação Matemática

Um problema de otimização é definido como sendo a busca pela solução ótima (mínimo ou máximo) de uma função – função objetivo. Em alguns casos pode ocorrer que as variáveis das funções que se deseja otimizar sejam restritas envolvendo uma região definida do espaço de busca. Neste caso, chamamos de problema de otimização com restrição.

A formulação e os termos apresentados aqui seguem a difundida literatura sobre algoritmos evolutivos (DEB, 2001; ZITZLER *et al.*, 1998; SCHAFFER, 1984; HORN *et al.*, 1994; COLLETTE & SIARRY, 2003; e outros).

Um problema multi-objetivo geral possui um número pré-definido de funções objetivo a serem otimizadas (maximizadas ou minimizadas), e também, existem restrições que devem ser satisfeitas por qualquer solução viável. De forma geral se tem o seguinte enunciado (DEB, 2001):

$$\begin{aligned} &\text{Minimizar / maximizar} && f_i(\boldsymbol{x}) = [\, f_1(\boldsymbol{x}),\, f_2(\boldsymbol{x}), ...,\, f_{N_{ob}}(\boldsymbol{x})], && i = 1, 2, ..., N_{ob} \\ &\text{Sujeito a} && g_j(\boldsymbol{x}) \geq 0, && j = 1, 2, ..., N_{re} \\ & && h_k(\boldsymbol{x}) = 0, && k = 1, 2, ..., N_{ig} \\ & && x_m^{(L)} \leq x_m \leq x_m^{(U)}, && m = 1, 2, ..., N_{var} \end{aligned} \tag{2.1}$$

Onde, $\boldsymbol{x}$ é um vetor de N_{var} variáveis de decisão: $\boldsymbol{x} = (x_1, x_2, ..., x_{N_{\text{var}}})^T$; $g_j(\boldsymbol{x})$ e $h_k(\boldsymbol{x})$ são funções de restrição de comportamento; $x_m^{(L)}$ e $x_m^{(U)}$ representam, respectivamente, o limite inferior e superior de cada uma das N_{var} variáveis de decisão x_m, estes limites

definem o *espaço de decisão* ED; $f_i(\boldsymbol{x})$ são as funções objetivo do problema; N_{ob} é o número de funções objetivo; N_{re} é o número de inequações de restrição; N_{ig} é o número de restrições de igualdade. O vetor $\boldsymbol{x}$ será referido também como *solução* do problema, uma vez que ele contém as variáveis a serem otimizadas.

Uma solução será viável ou factível quando não violar nenhumas das $N_{re} + N_{ig}$ restrições. Assume-se que a solução esteja dentro do espaço de busca, atendendo a $2 \cdot N_{\text{var}}$ limites. Caso haja pelo menos uma violação, a solução é considerada não-viável ou não-factível.

As restrições de igualdade $h_k(\boldsymbol{x})$ podem tornar o processo de otimização computacionalmente caro, já que atingir a igualdade numérica tratando-se de números reais em computadores é algo bem difícil. Com isso, estas restrições serão tratadas e transformadas em inequações como as $g_j(\boldsymbol{x})$, bastando-se acrescentar uma tolerância ε a cada restrição de igualdade da seguinte maneira: $h_j \geq \varepsilon$, e, conseqüentemente, $h_j - \varepsilon \geq 0$. Assim sendo, pode-se reescrever a formulação do problema de otimização multi-objetivo omitindo-se as restrições de igualdade e onde agora o j passa a representar o número total de restrições, ou seja, a soma das restrições de igualdade, transformadas, e as inequações:

$$\begin{aligned} &\text{Minimizar / maximizar} \quad f_i(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_{N_{ob}}(\boldsymbol{x})], && i = 1, 2, ..., N_{ob} \\ &\text{Sujeito a} \quad g_j(\boldsymbol{x}) \geq 0, && j = 1, 2, ..., N_{re} \\ &\qquad\qquad x_m^{(L)} \leq x_m \leq x_m^{(U)}, && m = 1, 2, ..., N_{var} \end{aligned} \tag{2.2}$$

Cada uma das N_{ob} funções objetivo $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_{N_{ob}}(\boldsymbol{x})$ pode ser minimizada ou maximizada, sendo que para se trabalhar com os algoritmos de otimização é necessário que todas sejam passíveis de minimização ou maximização. O vetor de funções objetivo $\boldsymbol{f(x)}$ constitui um espaço multidimensional chamado *espaço de objetivos* EO. Para cada solução $\boldsymbol{x}$ no *espaço de decisão* ED, existe um vetor $\boldsymbol{f(x)}$ no espaço de objetivos. Esta é uma diferença fundamental em relação à otimização uni-objetivo, cujo espaço de objetivos é uni-dimensional. Então, nos problemas multi-objetivos o mapeamento acontece entre um vetor $\boldsymbol{x}$ (N_{var}-dimensional) e um vetor $\boldsymbol{f(x)}$ (N_{ob}-dimensional). Se, por exemplo, cada elemento de $\boldsymbol{x}$ e $\boldsymbol{f(x)}$ for representado por números reais, então $\boldsymbol{f(x)}$ estaria mapeada como $\boldsymbol{f(x)}: \Re^{N_{\text{var}}} \rightarrow \Re^{N_{ob}}$.

2.1.2 Soluções Pareto-ótimas

Quando se trata de encontrar as melhores soluções de um problema, serão levados em consideração diversos fatores. Vejamos por exemplo a compra de um automóvel: ao se comprar um automóvel, o comprador estabelece os critérios de tomada de decisão que deverão satisfazer ao seu desejo de compra. Neste caso, foram estabelecidos dois critérios, ou duas medidas de qualidade, que serão avaliadas na hora da escolha de qual o melhor automóvel a se comprar. Estas são: o *conforto* e o *preço do veículo*. Não serão definidas aqui as formas de medida destes critérios, serão utilizadas apenas constatações na visão do comprador. A Figura 2.1 ilustra as opções de compra levantadas.

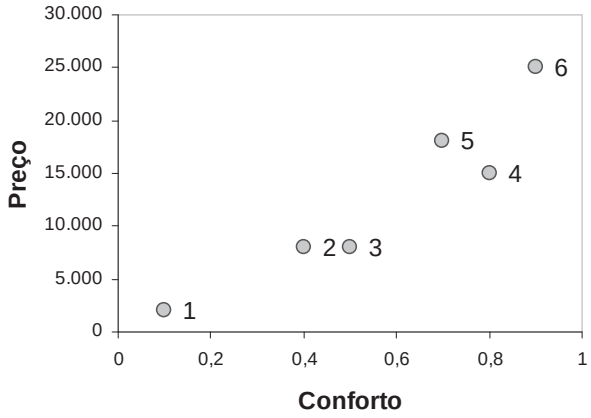


Figura 2.1: Exemplo de processo de decisão de compra de um automóvel.

O objetivo do comprador é de *minimizar* o preço e *maximizar* o conforto do veículo. Neste caso 6 opções de compra estão disponíveis. Com uma primeira análise das opções disponíveis, é possível se descartar algumas das opções: o *modelo* ***2***, com o mesmo preço do *modelo* **3** apresenta um conforto inferior; *modelo* ***5***, com preço superior ao *modelo* **4**, mas com conforto inferior. Com isso, tem-se então, quatro opções restantes: *modelos* **1**, **3**, **4** e **6**. Entre estes, uma relação de troca está estabelecida: quando se aumenta o conforto, aumenta-se também o custo, quando se diminui o conforto, diminui-se também o custo, não sendo possível encontrar qual das soluções melhor satisfaz ao comprador, já que não se tem uma opção em que há o menor custo e o maior conforto. Este exemplo ilustra a situação de um problema multi-objetivo (conforto e preço).

O conjunto de soluções de um problema multi-objetivo consiste de todos os vetores de decisão que não podem ser melhorados em um objetivo sem que haja a degradação de outro – estes vetores são conhecidos como *pareto-ótimos*. O conceito da *optimalidade de pareto* pode ser escrito matematicamente como se segue:

Seja um problema de **minimização** e considere dois vetores de decisão $\boldsymbol{a}, \boldsymbol{b} \in X$. Diz-se que a solução $\boldsymbol{a}$ domina a solução $\boldsymbol{b}$ (também escrito como $\boldsymbol{a} \prec \boldsymbol{b}$) se:

$$\forall i \in \{1, 2, ..., N_{ob}\} : f_i(\boldsymbol{a}) \leq f_i(\boldsymbol{b}) \wedge \exists ii \in \{1, 2, ..., N_{ob}\} : f_{ii}(\boldsymbol{a}) < f_{ii}(\boldsymbol{b}) \quad (2.3)$$

Ou seja, se todos os valores das funções objetivo f_i de $\boldsymbol{a}$ forem menores ou iguais aos valores de f_i de $\boldsymbol{b}$, e existir pelo menos um valor de função objetivo f_{ii} de $\boldsymbol{a}$ que seja simplesmente menor do que o valor da mesma função objetivo f_{ii} em $\boldsymbol{b}$.

Todos os vetores de decisão que não são dominados por outros vetores de decisão são chamados *não-dominados* ou *pareto-ótimos*. O conjunto de todas as soluções não-dominadas é chamado de *conjunto pareto-ótimo*. Este conjunto explicita a relação de troca entre os N_{ob} objetivos.

No exemplo da compra de um automóvel apresentado anteriormente, as soluções pareto-ótimas são os *modelos*: **1**, **3**, **4** e **6**. E, as soluções dominadas são os *modelos* **2** e **5**, já que em pelo menos um dos objetivos suas características são superadas pelas características de outra solução. As soluções pareto-ótimas constituem o *fronte de pareto* que é composto das soluções não-dominadas por nenhuma das outras soluções.

Ainda outro exemplo pode ser visto na Figura 2.2. Neste caso o problema é de minimização; este exemplo ilustra um conjunto de soluções que pertencem ao fronte de pareto, por exemplo as soluções *A* e *B* não-dominadas e um conjunto de soluções dominadas, como por exemplo a solução *C*. A solução *A* domina a solução *C* pois, $f_1(A)<f_1(C)$ e $f_2(A)<f_2(C)$. Assim como a solução *B* domina a solução *C,* pois $f_1(B)<f_1(C)$ e $f_2(B)<f_2(C)$. Com isso, as soluções que não são dominadas por nenhuma outra solução irão compor o fronte de pareto.

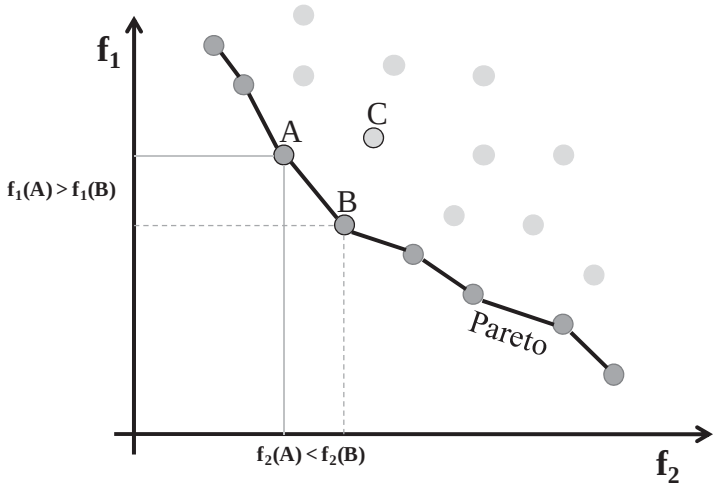


Figura 2.2: Exemplo de Fronte de Pareto

2.1.3 Propósito da Otimização Multi-objetivo

Quando se trata de obter um fronte de pareto, este deverá atender a algumas características que indicam sua qualidade. Um fronte de pareto, que representa a solução do problema de otimização multi-objetivo deverá ter *convergência* e *diversidade*, ou seja, o fronte encontrado deverá estar mais próximo possível das verdadeiras soluções do problema e o conjunto deve estar com as soluções distribuídas da forma mais homogênea possível sobre o fronte de pareto apresentado.

O primeiro propósito (*convergência*) é comum a qualquer procedimento de otimização, mas o segundo (*diversidade*) é particular da otimização multi-objetivo. A Figura 2.3a mostra soluções bem distribuídas na fronteira de pareto, já a Figura 2.3b mostra um fronte com uma distribuição de soluções ruim. Ambas as figuras tratam de um problema de minimização.

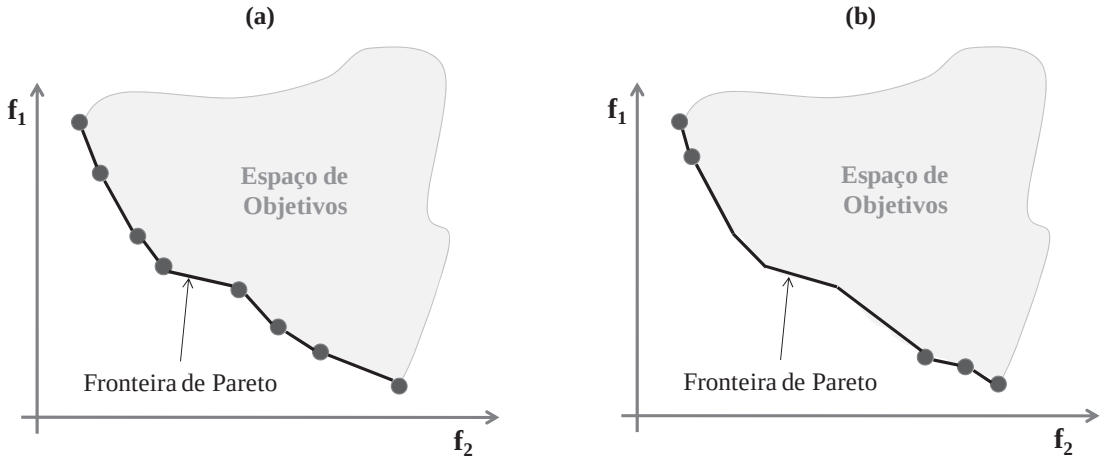


Figura 2.3: Distribuição das soluções no fronte de pareto

Como em um problema de otimização multi-objetivo trabalha-se com o espaço de decisão e o espaço de objetivos, é desejável que as soluções tenham uma boa diversidade nestes espaços. Normalmente, uma boa diversidade em um destes espaços garante também a diversidade no outro. Em alguns problemas isto não acontece.

2.1.4 Diferenças com a Otimização uni-objetivo

Pode-se relacionar as diferenças entre a otimização uni-objetivo e multi-objetivo, (DEB, 2001):

1. Problemas uni-objetivos visam encontrar uma solução ótima (seja minimização ou maximização), em problemas multi-modais podem existir mais de uma solução ótima. Já em otimização multi-objetivo, encontrar a fronteira de pareto é tão importante quanto preservar a sua diversidade. Um algoritmo eficiente deve ser capaz de lidar com ambos os aspectos.

2. Um problema de otimização multi-objetivo trabalha com dois espaços (variáveis e objetivos) no lugar de um. Garantir a diversidade torna-se um problema mais difícil uma vez que duas soluções próximas no espaço de variáveis não implicam em soluções próximas no espaço de objetivos.

3. Os métodos tradicionais de otimização multi-objetivo são baseados em uma função simples a qual pondera cada objetivo. Podem também tratar cada objetivo separadamente, utilizando os demais objetivos como restrições. Assim, um problema multi-objetivo pode ser convertido mediante algumas técnicas, em um problema uni-objetivo.

Assim, definido um problema multi-objetivo, segue-se com a apresentação de algoritmos e aplicações existentes no estado da arte.

2.2 ESTADO DA ARTE

Nesta seção serão apresentados os principais algoritmos disponíveis na literatura. Os algoritmos serão brevemente discutidos em alguns aspectos. Vale lembrar que seria impossível abordar todos os artigos disponíveis na literatura, ficando aqui apenas uma amostra do que existe. Um enfoque especial será dado aos algoritmos desenvolvidos que utilizem o Sistema Imunológico Artificial em virtude da proposta deste trabalho.

2.2.1 Algoritmos Genéticos (AG)

Os Algoritmos Genéticos (AG) são algoritmos inspirados nos princípios de sobrevivência e evolução das espécies (GOLDBERG, 1989).

Segundo *Charles Darwin*[1], os indivíduos mais aptos possuem uma probabilidade maior de reprodução e com isso seus descendentes mantêm o bom material genético na espécie. Este material genético é representado pelo *cromossomo*, que constitui a identidade única de cada indivíduo.

No AG, cada indivíduo, junto com seu material genético único, representa uma solução para o problema. Cada parâmetro (x_m) é codificado por um *gene* utilizando-se de uma representação apropriada, como a representação por números reais ou uma cadeia binária. Cada gene corresponde à representação de um parâmetro ($x_1, x_2, ..., x_{N_{\text{var}}}$) e, o

[1] *Charles Robert Darwin* foi um naturalista britânico que alcançou fama ao convencer a comunidade científica da ocorrência da evolução e propor uma teoria para explicar como ela se dá por meio da reprodução sexual e seleção natural.

cromossomo, é a união de todos os genes, representando uma solução ao problema. Uma população é compreendida por um conjunto de soluções (*cromossomos*).

Durante o processo de evolução, cada indivíduo é submetido ao processo de reprodução. A reprodução é realizada através da *recombinação* para combinar os genes de diferentes pais e produzir seus descendentes. Os descendentes carregam características de ambos os pais, e são submetidos em seguida ao processo de *mutação*, que confere características verdadeiramente novas aos filhos.

Os indivíduos são avaliados pela função objetivo que define o problema. Como resultado da avaliação, cada indivíduo recebe uma medida de aptidão que representa a qualidade da solução. Ao final da geração, somente os indivíduos mais aptos são selecionados para continuar na população, os demais são descartados.

O aprimoramento da população ocorre como conseqüência da repetida seleção dos melhores indivíduos de cada geração, isto se dá, pois estes possuem maior probabilidade de produzir bons descendentes.

A respeito dos algoritmos genéticos serão descritos os algoritmos listados a seguir:

VEGA - *Vector Evaluated Genetic Algorithm* (SCHAFFER, 1984);

MOGA - *Multi-Objective Genetic Algorithm* (FONSECA & FLEMING, 1993);

NPGA - *Niched Pareto Genetic Algorithm* (HORN *et al.,* 1994);

NSGA - *Nondominated Sorting Genetic Algorithm* (SRINIVAS & DEB, 1994);

NSGA-II - *Nondominated Sorting Genetic Algorithm II* (DEB *et al.*; 2000a);

PMOGA - *Pareto Multi-Objective Genetic Algorithm* (CASTRO, R. E., 2001)

2.2.1.1 VEGA

SCHAFFER (1984) propôs o primeiro algoritmo evolutivo destinado a problemas multi-objetivos. O VEGA (***V**ector **E**valuated **G**enetic **A**lgorithm*) se diferencia de um algoritmo genético comum pela forma como é feita a seleção dos melhores indivíduos.

Em VEGA, uma população de n_{indiv} inicializada aleatoriamente é dividida em N_{ob} sub-populações de igual tamanho (n_{indiv} / N_{ob}). Então, o valor da aptidão das soluções é calculado em cada sub-população baseado em cada função objetivo. As soluções são selecionadas pelo método da roleta a partir destas sub-populações para o cruzamento e mutação. O Cruzamento e mutação são realizados na nova população da mesma forma

como é feito no AG uni-objetivo. Este algoritmo não incorpora diretamente o conceito de pareto-otimalidade.

Uma desvantagem deste procedimento foi levantado por RICHARDSON *et al.* (1989) que destacou que a mistura das N_{ob} sub-populações era equivalente a se utilizar uma combinação linear dos componentes do vetor de aptidão tratando o problema como uni-objetivo. O próprio SCHAFFER (1984) havia antecipado isso e chamou este problema de e*specialização.*

2.2.1.2 MOGA

FONSECA & FLEMING (1993) foram os primeiros a incorporar o conceito de pareto-dominância para aferição da qualidade das soluções seguindo as sugestões de GOLDBERG (1989). Toda a população de n_{indiv} é verificada quanto a sua dominância. Os indivíduos recebem uma classe igual a um mais o número de indivíduos da população corrente pelos quais ele é dominado. Os indivíduos não-dominados recebem uma classe igual a um e os indivíduos dominados possuem uma classe com valor menor que o tamanho a da população. A seguir, a população é ordenada de acordo com a classificação recebida. A atribuição de aptidão por camadas, colocando no mesmo nível indivíduos de mesmo grau de dominância, leva a produzir uma grande pressão de seleção que pode causar convergência prematura.

Preocupados com a manutenção da diversidade, FONSECA & FLEMING (1993) propuseram também o uso do método de formação de nichos (compartilhamento da aptidão) para distribuir a população ao longo da região ótima de pareto. Este mecanismo tende a manter a diversidade das soluções no conjunto de valores da função de aptidão. No entanto, em problemas onde diferentes pontos do conjunto pareto-ótimo correspondem aos mesmos valores de aptidão, o MOGA poderá não ser capaz de encontrar um fronte de pareto com múltiplas soluções.

O grande destaque deste trabalho é o ordenamento de soluções por dominância. Uma desvantagem é a lenta convergência e dificuldade de escolha do tamanho do nicho.

2.2.1.3 NPGA

Este algoritmo apresenta uma forma diferente de seleção, particularmente, modifica a operação de seleção por torneio. O processo de seleção por torneio passa a se dar de duas maneiras: primeiro é realizada uma seleção por torneio baseada no critério de dominância de pareto *(pareto domination tournaments*); em seguida, ocorre um torneio entre duas soluções não-dominadas quando o compartilhamento de aptidão é usado para determinar o vencedor. Os torneios ocorrem da seguinte maneira: um

conjunto de indivíduos é escolhido aleatoriamente da população para compor um conjunto para comparação. Dois indivíduos são escolhidos aleatoriamente na população e comparados com estes indivíduos do conjunto de comparação. Se um dos candidatos for dominado pelo conjunto de comparação, e o outro não, então este último é selecionado para reprodução. Caso os dois sejam dominados ou dominem o conjunto de comparação, então é aplicado um critério de compartilhamento de aptidão para a escolha do indivíduo.

O tamanho do conjunto de comparação (t_{dom}) é fundamental e irá determinar a pressão de seleção e, conseqüentemente, o desempenho do algoritmo. Se o conjunto for grande, a população irá convergir prematuramente para apenas uma parte da fronteira de pareto; se for muito pequena, a população final será formada por muitas soluções dominadas.

O conceito de formação de nichos entre os pontos não-dominados é um dos aspectos mais relevantes do trabalho, que apresenta o inconveniente de possuir mais parâmetros a serem configurados, em especial o tamanho do nicho e parâmetros extras para a seleção por torneio.

2.2.1.4 NSGA

O algoritmo NSGA (***N****ondominated* ***S****orting* ***G****enetic* ***A****lgorithm*) é baseado num conceito de ordenamento de soluções não-dominadas. Este difere de um algoritmo genético clássico apenas pela forma como a seleção ocorre.

Antes de a seleção ser executada a população é ordenada com base na dominância de cada indivíduo. Em seguida são identificados os indivíduos não-dominados, que irão compor o primeiro fronte não-dominado, o qual recebe um valor alto de aptidão (*dummy fitness value*). Este valor é atribuído para permitir um alto potencial de reprodução a todas as soluções não-dominadas. Objetivando manter a diversidade da população executa-se o compartilhamento da aptidão para estas soluções. Após o compartilhamento estas soluções são deixadas de lado temporariamente efetuando-se o mesmo procedimento para o segundo fronte e os demais frontes consecutivos. A diferença é que nos frontes consecutivos ao primeiro, o valor atribuído de aptidão (*dummy fitness value*) vai sendo reduzido nos frontes subseqüentes.

Como os indivíduos do primeiro fronte possuem maior valor de aptidão, estes serão selecionados mais vezes que o resto da população, o que permite uma busca tendendo a regiões não-dominadas.

Uma das características mais importante deste trabalho é que praticamente qualquer número de objetivos pode ser usado sem distinção quanto ao problema ser de

minimização ou maximização. O algoritmo, de maneira geral, possui uma rápida convergência. Como ponto fraco apresenta a existência de novos parâmetros de configuração que são sensíveis e provocam forte influência no desempenho da busca, especialmente o tamanho do nicho.

2.2.1.5 NSGA-II

DEB *et al.* (2000a) desenvolveram modificações no NSGA que foram: um procedimento de ordenamento rápido de soluções não-dominadas, um mecanismo de elitismo e um operador de aglomeração sem parâmetros de ajuste para o manutenção da diversidade no conjunto de soluções externas. Este algoritmo é citado como um dos otimizadores com melhor desempenho.

2.2.1.6 PMOGA

CASTRO, R. E. (2001) apresenta em seu trabalho o desenvolvimento de um algoritmo genético para otimização de problemas multi-objetivos. O algoritmo proposto é uma evolução de um algoritmo genético básico com as seguintes características: (i) ordenamento da população de acordo com propriedades de dominância, (ii) utilização de operadores multi-objetivos adequadamente desenvolvidos: exclusão, adição e operador de otimização individual, (iii) definição de um filtro para reter as soluções de pareto, (iv) elitismo deste filtro dentro das população e (v) método de penalização por camadas.

As principais melhorias ou evoluções neste algoritmo foram: modificação do método de seleção; modificações do operador de recombinação; desenvolvimento e introdução dos operadores multi-objetivos e introdução do elitismo do filtro nas populações.

Os resultados encontrados com o algoritmo apresentado foram equivalentes ou superiores aos apresentados na literatura com um bom desempenho computacional. Comprovando a boa integração entre as várias estratégias escolhidas e desenvolvidas para formarem o corpo do algoritmo multi-objetivo proposto.

2.2.2 Estratégias Evolutivas (EE)

As Estratégias Evolutivas são uma subclasse da computação evolutiva em que são aplicados conceitos de evolução de uma população de soluções baseadas em princípios biológicos de evolução como: reprodução, mutação, recombinação e seleção. As estratégias evolutivas foram apresentadas pela primeira vez em RECHENBERG (1965).

Estratégias evolutivas foram desenvolvidas para otimização numérica com vetores de ponto flutuante. Cada indivíduo de uma população é representado por $(\boldsymbol{x}, \boldsymbol{\sigma})$, em que

$\boldsymbol{x}$ é um vetor que representa um ponto no espaço de busca $\boldsymbol{x} = \left(x_1, x_2, ..., x_{N_{\text{var}}} \right)^T$, que deverá otimizar uma ou mais funções objetivas no domínio do problema, e $\boldsymbol{\sigma}$ é um vetor de desvios padrões $\boldsymbol{\sigma} = \left(\sigma_1, \sigma_2, ..., \sigma_{N_{\text{var}}} \right)^T$ com a dimensão de $\boldsymbol{x}$. Cada nova geração pode ser obtida por operações de cruzamento e de mutação. A recombinação não é essencial em Estratégias Evolutivas. Já a mutação é um elemento crucial. No caso de não haver recombinação, a mutação deve dar conta tanto do aspecto de busca quanto de diversidade. No caso mais simples ela é aplicada sobre o(s) indivíduo(s) genitor(es) como se segue: $\boldsymbol{x}' = \boldsymbol{x} + N(0, \boldsymbol{\sigma})$, onde $\boldsymbol{x}'$ é o vetor das variáveis modificado, e $N(0, \boldsymbol{\sigma})$ é um vetor de números aleatórios independentes que segue uma distribuição Gaussiana (BÄCK *et al.*, 1991).

A seguir são apresentados alguns algoritmos para resolução de problemas de otimização multi-objetivos inspirados no conceito de algoritmos evolutivos:

SPEA - *Strength Pareto Evolutionary Algorithm* (ZITZLER *et al.*, 1998);

SPEA2 - *Strength Pareto Evolutionary Algorithm 2* (ZITZLER *et al.*, 2001);

PAES - *Pareto Archived Evolution Strategy* (KNOWLES & CONRE, 2000);

(1+1) PAES-M - *Estratégia Evolutiva com Pareto Arquivado Modificado* (BARDANACHVILI, 2006);

2.2.2.1 SPEA

ZITZLER *et al.* (1998) apresentam um algoritmo evolutivo multi-objetivo elitista com conceitos de pareto-dominância visando manter a população diversificada pela formação de nichos. Este algoritmo combina o uso de uma população externa para o armazenamento de soluções não-dominadas (ótimas) encontradas a cada passo do tempo, com técnicas baseadas em pareto-dominância para a atribuição de aptidão de cada indivíduo e o uso de técnicas de agrupamento para reduzir o armazenamento de soluções não-dominadas.

Objetivando alcançar uma distribuição uniforme da população na fronteira de pareto (conjunto externo), a atribuição de aptidão é realizada da seguinte maneira: primeiro, a cada solução no conjunto de pareto (externo) é atribuído um valor representando a "força de pareto" (do inglês "*pareto strength*"), sendo este valor proporcional ao número de soluções dominadas por cada indivíduo. A "força de pareto" de cada solução é também a sua aptidão. A "força de pareto" de uma solução ie do conjunto externo será igual ao número de indivíduos N_{dom} na população que são dominados por ou iguais ao indivíduo ie com respeito aos valores das funções objetivo,

dividido pelo tamanho da população somado de um. Subseqüentemente, um indivíduo ip na população corrente recebe a aptidão como sendo o resultado da soma da "força de pareto" de todas as soluções do conjunto externo que a dominam ou são iguais ao indivíduo ip, adicionando-se um ao resultado.

A diversidade da população é mantida através de uma técnica de agrupamento chamada método de ligação média (*average linkage method*). Neste método a distância entre os dois grupos é definida como a média das distâncias entre todos os pares de objetos, onde um membro do par é de cada um dos grupos. Este método também utiliza informações sobre todos os pares de distância, não apenas as distâncias mínima e máxima. Esta estratégia reflete a idéia de preferir indivíduos próximos à fronteira de pareto e ao mesmo tempo distribuídos ao longo da superfície de busca.

Segundo ZITZLER *et al.* (2001) alguns problemas surgem deste algoritmo: indivíduos dominados pelos mesmos indivíduos do conjunto externo possuem aptidão idênticas, portanto, se o conjunto externo possui somente um elemento então todos os indivíduos terão a mesma aptidão (o que faz a busca se comportar como aleatória); pouca ou nenhuma informação pode ser extraída a partir da dominância se muitos indivíduos são indiferentes, ou seja, não dominam um ao outro (situação comum em problemas com mais de dois objetivos); o algoritmo de agrupamento pode destruir soluções no fronte de pareto.

2.2.2.2 SPEA2

O algoritmo SPEA2 (ZITZLER *et al.*, 2001) surge como um melhoramento do algoritmo predecessor SPEA (ZITZLER *et al.*, 1998), com intuito de diminuir ou acabar com suas fraquezas. Neste, a atribuição da aptidão foi redesenhada, passando a levar em consideração quantos indivíduos dominam e são dominados por cada indivíduo (considerando população corrente e conjunto externo). O mecanismo de agrupamento foi trocado por uma nova forma de truncamento do conjunto externo que garante a preservação das soluções no fronte de pareto. Foi adicionada uma técnica para estimativa da densidade baseada no *k-ésimo vizinho mais próximo* onde a densidade de cada ponto é uma função (decrescente) da distância deste ponto ao *k-ésimo* ponto mais próximo.

2.2.2.3 PAES

KNOWLES & CONRE (2000) desenvolveram um algoritmo simples baseado em estratégia evolutiva chamado de PAES (***P**areto **A**rchived **E**volution **S**trategy*). Em PAES, uma solução genitora gera uma solução filha, através de mutação; esta solução filha é comparada à genitora: se a filha dominar a genitora, esta é aceito como genitora para a

Um conjunto de soluções não-dominadas é utilizado para manter a diversidade ao longo do fronte de pareto. Cada nova solução filha gerada é comparado ao conjunto para verificar se esta domina algum membro deste conjunto; caso a filha domine uma solução do conjunto, então a solução filha é adicionada ao conjunto e usada como genitora na próxima geração, e as soluções dominadas são eliminadas do arquivo; caso contrário, a filha não domine nenhuma solução do conjunto, então, é medida a distância entre a genitora e a filha a cada membro do conjunto, buscando-se a solução mais próxima. Se a filha residir em uma região com pouca densidade de soluções no espaço dos parâmetros, então esta é aceita como genitora na próxima geração e uma cópia é adicionada ao conjunto.

Em CHEN & LU (2008) o algoritmo PAES é utilizado para comparação com outros algoritmos do estado da arte nas funções matemáticas ZDT1, ZDT2, ZDT3, ZDT4 e ZDT6.

2.2.2.4 (1+1) PAES-M

A notação (1+1) refere-se a uma situação particular de uma notação abrangente ($\mu + \lambda$), sendo μ o número de indivíduos da geração anterior e λ o número de indivíduos gerados por recombinação e mutação. Caso o algoritmo seja elitista, são removidos os indivíduos menos ajustados de $\mu + \lambda$, representado por ($\mu + \lambda$)-ES; caso seja não-elitista, são removidos os menos ajustados de somente λ indivíduos, com a notação (μ, λ)-ES, voltando-se a μ indivíduos em ambos os casos. Há ainda variantes como (1+λ)-ES em que, de apenas um pai, são gerados λ descendentes. A tese de BARDANACHVILI (2006) desenvolve um dos mais simples algoritmos para otimização multi-objetivo capaz de gerar um conjunto diversificado de soluções no fronte de pareto. O trabalho é baseado no método (1+1)-PAES (KNOWLES & CONRE, 2000) que gera um arquivo de referência para armazenar as soluções previamente encontradas no fronte. O arquivo é então utilizado para estabelecer o *ranking* de dominância aproximado entre a solução atual e a solução candidata gerada. Em (1+1)-PAES-M, foram introduzidas modificações ao algoritmo original que são: pesquisa global do (1+1)-ES feita através da utilização de dois níveis de desvio padrão para mutação: um global e outro local; o algoritmo realiza tanto simulações individuais como conjuntas das variáveis do problema para acelerar a convergência; permite que as restrições sejam tratadas, sem o emprego

Seu algoritmo foi testado em diversas funções matemáticas (ZDT1, ZDT2, ZDT3, ZDT4 e ZDT6; TNK, MEC e outros) obtendo níveis de precisão, robustez e eficiência comparáveis aos melhores algoritmos AG e EE encontrados na literatura.

2.2.3 Sistemas Imunológicos Artificiais (SIA)

Este algoritmo será descrito em mais detalhes por ser parte importante do trabalho desta tese. Dentre outras coisas, são apresentados aspectos do desenvolvimento deste algoritmo, importantes para o funcionamento como multi-objetivo.

Os sistemas imunológicos artificiais surgiram devido a tentativas da engenharia e da ciência da computação em simular um mecanismo particular do sistema imune, com o objetivo de criar sistemas artificiais para resolver problemas de engenharia. Essas tentativas foram função da analogia existente entre o reconhecimento de padrão do sistema imunológico e o reconhecimento de padrão em computação, detecção e eliminação de vírus no organismo e detecção de invasão em redes de computadores.

O Sistema Imunológico Artificial (SIA) tem como inspiração biológica o comportamento das células de defesa do organismo dos animais (anticorpos), chamados linfócitos, quando em contato com um agente estranho ao organismo (antígeno). Os linfócitos fazem parte do sistema imune adaptativo e podem ser de dois tipos principais: linfócitos B (células B) ou linfócitos T (células T) (CASTRO, L. N., 2001). O princípio da seleção clonal, que é capaz de explicar as funções básicas de respostas do sistema imune ao estímulo de um antígeno, inspirou o desenvolvimento de poderosas ferramentas computacionais (CASTRO & ZUBEN, 2001). Este princípio é associado às funções básicas do sistema imune adaptativo em resposta ao estímulo de um antígeno. Ele estabelece que somente a célula capaz de reconhecer certo estímulo antigênico irá proliferar.

Quando é exposto a um antígeno, células do sistema imune irão atuar buscando eliminar estes invasores. Como resultado serão liberadas substâncias que podem ser percebidas pelos receptores das células T. Estas células uma vez ativas se dividem e secretam outro tipo de substâncias que mobilizam outros componentes do sistema imunológico. Diferentemente dos receptores das células T, as células B são capazes de reconhecer partes livres solúveis dos antígenos, sem necessitar das substâncias geradas pela resposta do sistema inato. Assim, quando ativadas, as células B se dividem e se diferenciam em plasmócitos, secretando anticorpos em altas taxas. Anticorpos são formas solúveis dos receptores das células B. Cada célula secreta somente um tipo de anticorpo

que é relativamente específico ao antígeno encontrado. Isto significa uma resposta imune específica para cada antígeno (CASTRO, L. N., 2001).

As células B, além de se proliferar e diferenciar em plasmócitos, também podem se diferenciar em células B de memória, que permanecerão no organismo garantindo uma reposta mais rápida a futuras invasões do mesmo antígeno. As Células T não secretam anticorpos, mas realizam um papel central na regulação da resposta das células B e são importantes nas respostas imunes mediadas por células (CASTRO, L. N., 2001, CASTRO & ZUBEN, 2000).

O princípio da seleção clonal pode ser ilustrado através da Figura 2.4.

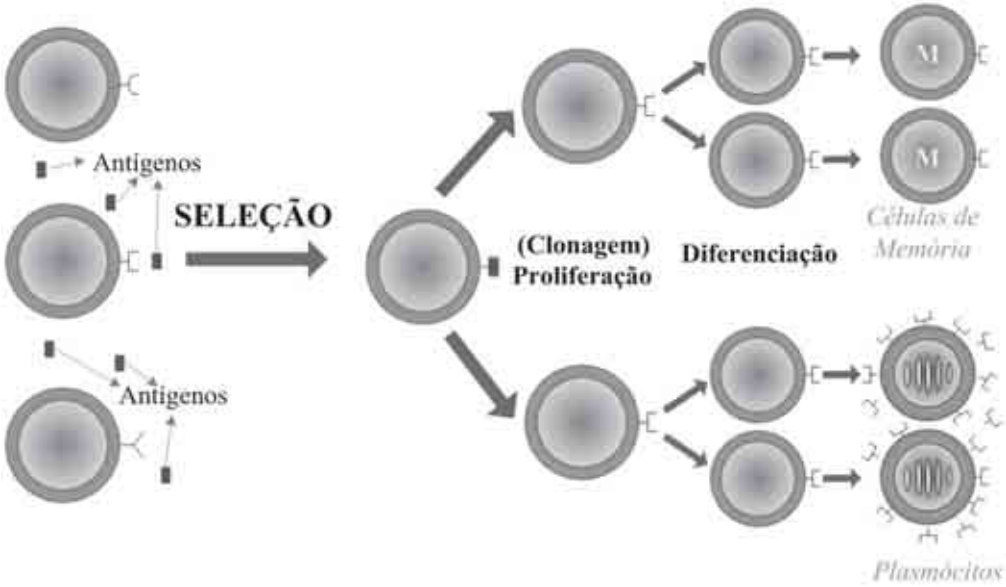


Figura 2.4: Princípio da seleção clonal (CASTRO & ZUBEN, 2000)

O processo de otimização baseado no princípio da seleção clonal (CASTRO & ZUBEN, 2001) do sistema imunológico artificial (SIA – CLONALG) será descrito a seguir. Este algoritmo foi inicialmente proposto para atender a problemas de aprendizagem de máquina e reconhecimento de padrões, e depois modificado para o problema de otimização *uni-objetivo*. Neste algoritmo, *afinidade* tem o mesmo significado que a aptidão de um algoritmo genético (AG). Este processo pode ser visualmente representado através da Figura 2.5 a seguir.

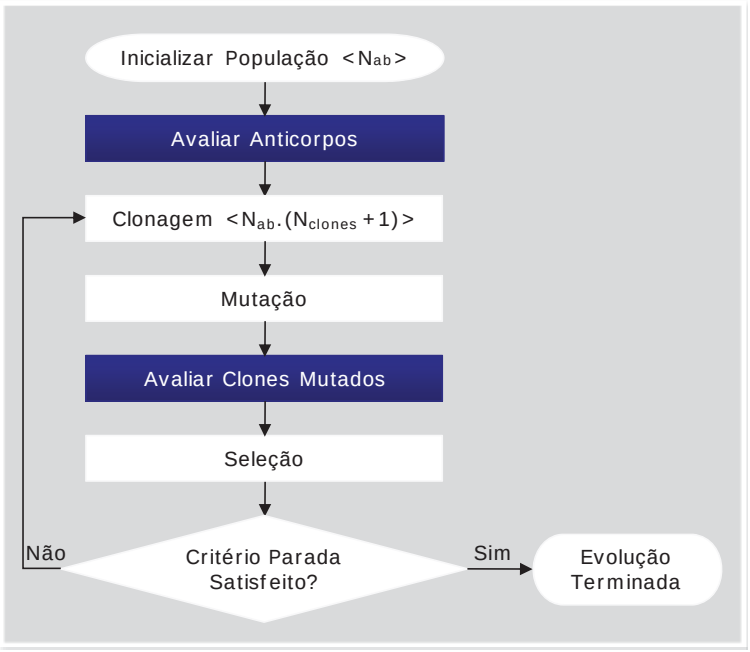


Figura 2.5: Representação do SIA - *CLONALG*

1) *Inicialização da população*: a população de soluções iniciais é gerada aleatoriamente. Para cada parâmetro (variável) é gerado um valor aleatório respeitando o limite inferior e superior do espaço de busca de cada variável.

2) *Avaliação*: a população de anticorpos será avaliada (cálculo da função objetivo de cada anticorpo).

3) *Clonagem*: toda a população de anticorpos é selecionada para a realização da clonagem. A clonagem gera cópias idênticas de cada anticorpo. A população nesta etapa encontra-se com o tamanho igual a: $(N_{clones} + 1)N_{ab}$. Onde N_{clones} é igual ao número de clones e N_{ab} é igual ao número de anticorpos.

4) *Mutação*: este operador é responsável pela modificação dos clones gerados permitindo a realização da busca local e global no caso do SIA. A mutação poderá ser de vários tipos, por exemplo: somática, não-uniforme, diferencial etc.

5) *Avaliação*: a população de clones será avaliada (cálculo da função objetivo cada clone).

6) *Seleção*: etapa em que os anticorpos com maior afinidade são selecionados para que venham a compor a população da nova geração, ou solução do problema em si. O processo de seleção compara cada anticorpo com seu grupo de clones (gerados no passo 3 e mutados no passo 4), caracterizando uma espécie de seleção por família. Por exemplo, numa população de 2 anticorpos e 3 clones por anticorpo, o anticorpo 1 será comparado aos seus 3 clones já mutados e será escolhido o de maior afinidade, que substituirá o anticorpo 1, o mesmo acontece com o anticorpo 2.

7) *Avaliação do critério de parada*: Repita os passos 3-6 até que o critério de parada previamente definido seja alcançado.

A mutação estabelecida no passo 4 pode ser de várias formas. Neste trabalho são apresentadas duas delas: a *hipermutação* somática e a mutação diferencial modificada. A mutação é a responsável pela maturação das células B do sistema imune.

O processo de seleção estabelecido no passo 6 permite que a evolução do algoritmo seja realizada dentro de cada grupo, ou família, separadamente, permitindo explorar simultaneamente diferentes posições do espaço de busca. Esta característica, juntamente com a mutação, são responsáveis pela correta evolução do algoritmo uni-objetivo.

Hipermutação Somática:

A *hipermutação* somática é dependente do valor da afinidade, quanto melhor o indivíduo (afinidade mais alta, para o caso de maximização, por exemplo), menor a taxa de mutação que será aplicada, e vice-versa. A *hipermutação* somática se dá de acordo com a Eq. (2.4) onde as variáveis em negrito são vetores.

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\beta} \cdot \exp(-\rho \cdot F^*) . \tag{2.4}$$

Onde $\boldsymbol{x}'$ é o vetor de N_{var} variáveis de decisão modificado: $\boldsymbol{x}' = \left(x_1', x_2', ..., x_{N_{\text{var}}}'\right)^T$; $\boldsymbol{x}$ é o vetor de N_{var} variáveis de decisão: $\boldsymbol{x} = \left(x_1, x_2, ..., x_{N_{\text{var}}}\right)^T$ original; ρ é um parâmetro que controla a taxa de decaimento da exponencial inversa (valor pré-definido); F^* é a afinidade normalizada (entre 0 e 1), que pode ser determinada por $F^* = F / F_{\max}$, onde F é a afinidade do anticorpo, e $F_{\max}$ é a máxima afinidade dos anticorpos; e $\boldsymbol{\beta} = \left(\beta_1, \beta_2, ..., \beta_{N_{\text{var}}}\right)$ é um vetor de parâmetros pré-definidos que controlam a ordem de grandeza da mutação em cada variável.

É importante apontar que os valores de $\boldsymbol{\beta}$ deverão ser pré-definidos e sua escolha irá impactar na ordem de grandeza da mutação em cada variável livre. Para conseguir cobrir uma região maior de valores, na atual implementação é utilizada uma equação de decaimento linear em função da geração (ou número de avaliações, dependendo do critério de parada escolhido) para os valores de $\boldsymbol{\beta}$. Assim no início do

processo evolutivo, β assume o valor inicialmente pré-definido e irá decaindo linearmente até um valor pré-definido no final do processo. De maneira geral, um valor que leva a bons resultados é utilizar um β igual à metade do intervalo da variável, ou seja, igual à diferença entre o limite superior e inferior para cada variável dividido por dois. Outros valores também podem ser utilizados apresentando bons resultados. Adotando-se a configuração apresentada acima, o decaimento linear fará com que se reduza a ordem de grandeza do valor da taxa de mutação imposta ao algoritmo permitindo mutações grandes e pequenas.

Objetivando aumentar o desempenho do algoritmo, a relação entre a afinidade normalizada $F*$ e a taxa de mutação $\exp(-\rho \cdot F^*)$ foi estudada, conforme é mostrado na Figura 2.6 pelas curvas de diferentes valores de ρ. Pode-se perceber que os menores valores de ρ levam aos maiores níveis de variação da curva. Por exemplo, para ρ = 1, até mesmo o melhor indivíduo (F^* =1) apresentará uma alta taxa de mutação (0,37) prejudicando a busca local. Para valores maiores de ρ, a sensibilidade da busca local é aumentada, mas a capacidade de exploração é diminuída.

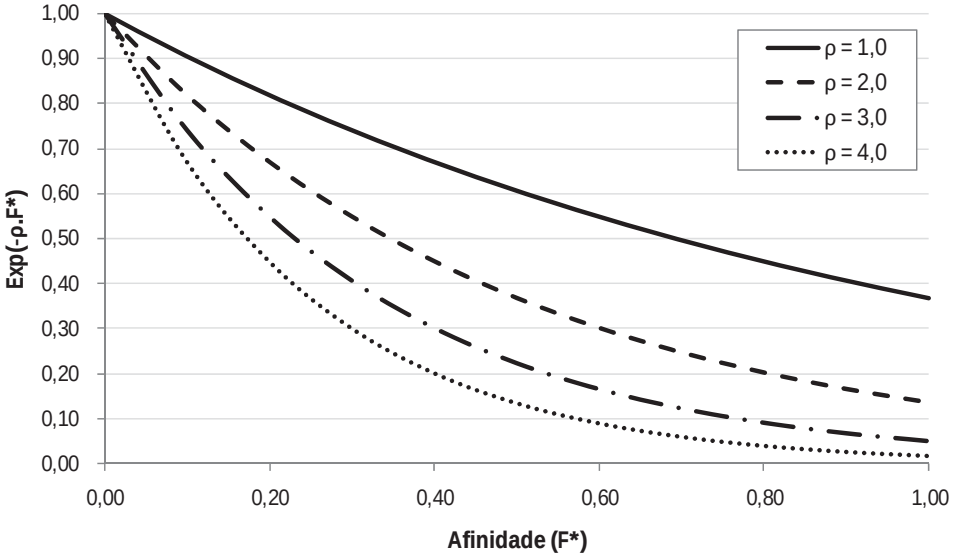


Figura 2.6: Taxa de mutação como função da afinidade $F*$

Nota-se pela análise da figura anterior que quanto menor o ρ, menos sensível as variações será a curva da taxa de mutação.

Segundo esta linha de raciocínio, seria razoável associar o valor de ρ à qualidade de cada indivíduo, para definir quando este deve se comportar como uma busca local ou uma busca global. Então, é proposta, uma variação linear de ρ, onde na Eq. (2.4), ρ é substituído por um parâmetro adaptativo definido como uma função do valor da afinidade de cada anticorpo.

$$\rho_{Ad} = \left[\frac{\rho_{máx} - \rho_{mín}}{F^*_{máx} - F^*_{mín}} \right] \cdot F^* + \left\{ \rho_{máx} - F^*_{máx} \cdot \left[\frac{\rho_{máx} - \rho_{mín}}{F^*_{máx} - F^*_{mín}} \right] \right\}. \quad (2.5)$$

Nesta expressão, $\rho_{mín}$ e $\rho_{máx}$ são parâmetros pré-fixados, representando respectivamente valor mínimo e máximo para ρ_{Ad} (valor adaptativo de ρ); F^* é o valor da afinidade normalizada de cada anticorpo, $F^*_{mín}$ e $F^*_{máx}$ são respectivamente o valor mínimo e máximo de afinidade normalizada entre todos os anticorpos na geração atual. Desta forma, o anticorpo com a menor afinidade receberá o menor valor de ρ_{Ad} e aquele com o maior valor de afinidade receberá o maior valor de ρ_{Ad}. Como os valores de $F^*_{mín}$ e $F^*_{máx}$ serão iguais a 0 e 1 respectivamente, a equação pode ser simplificada, ficando:

$$\rho_{Ad} = \left[\rho_{máx} - \rho_{mín} \right] \cdot F^* + \left\{ \rho_{máx} - \cdot \left[\rho_{máx} - \rho_{mín} \right] \right\}. \quad (2.6)$$

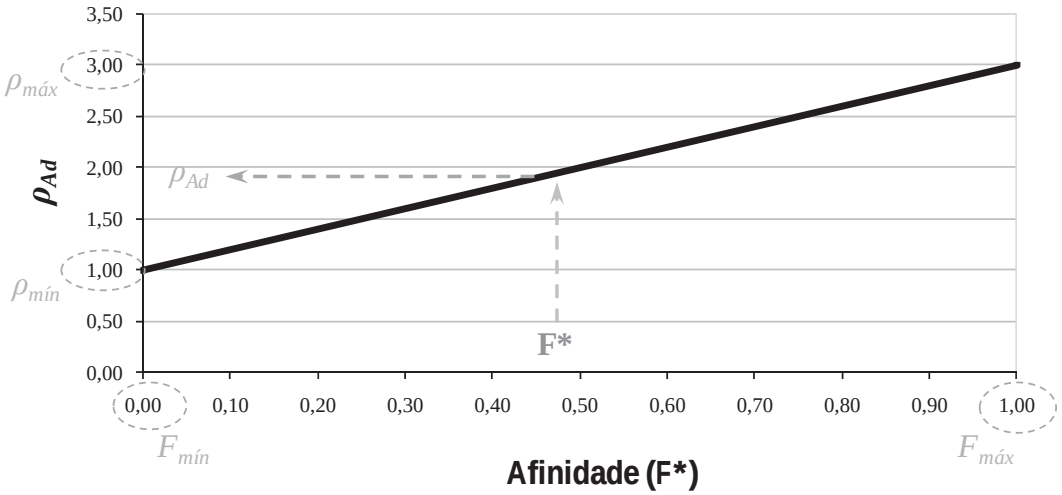


Figura 2.7: Exemplo de funcionamento do ρ_{Ad}

A Figura 2.7 traz um exemplo de funcionamento do ρ_{Ad}, onde o valor de $F^*_{mín}$ é 0,00; $F^*_{máx}$ igual a 1,00; $\rho_{mín}$ igual a 1,00 e $\rho_{máx}$ igual a 3,00. Pode-se perceber que quanto pior o anticorpo, menor será o valor de ρ_{Ad} e conseqüentemente, maiores serão as taxas de mutação, e quanto melhor o anticorpo, maior o valor de ρ_{Ad}, indicando menores taxas de mutação.

Esta modificação tem a intenção de dar a cada anticorpo uma maior habilidade de busca, permitindo que os melhores sejam responsáveis pela busca local e os piores pela busca global. Os efeitos práticos foram apreciados em VIEIRA *et al.* (2009).

Mutação Diferencial Modificada:

Esta mutação foi desenvolvida para o presente trabalho e teve como inspiração o algoritmo da *Evolução Diferencial* (PRICE, *et al*., 2005, OLIVEIRA, *et al.*, 2007, FEOKTISTOV, 2006). Como o foco do presente trabalho não é a abordagem do algoritmo da *Evolução Diferencial*, este não será aqui apresentado. Apenas será exposta a mutação desenvolvida a partir das idéias e conceitos presentes no algoritmo da *Evolução Diferencial.* Caso seja de interesse, as referências supracitadas são perfeitamente completas na apresentação do referido algoritmo.

A mutação desenvolvida se resume basicamente em um mecanismo de interpolação entre as soluções no espaço de busca. Após algumas rodadas com o SIA multi-objetivo utilizando a mutação diferencial modificada em funções matemáticas, foi possível perceber que esta mutação apresenta uma limitação quando se trata de exigir taxas de mutação com pequena ordem de grandeza. Outro problema ocorre quando os parâmetros a serem otimizados possuem grande ordem de grandeza, como por exemplo 10^3, e apresentam importante variação no valor da função objetivo para variações de pequena ordem de grandeza, como por exemplo 10^{-3}, exigindo ao mesmo tempo taxas de mutação grandes e pequenas.

A capacidade de interpolação é o destaque desta mutação. Esta característica é importante para que ao final do processo de otimização as soluções apresentem uma distribuição homogênea, ou seja, com soluções bem distribuídas. O problema com as pequenas ordens de grandeza decorre do fato da mutação ser aditiva (ou subtrativa) e proporcional aos valores dos parâmetros dos indivíduos selecionados, fazendo com que seja difícil a percepção de valores da ordem de grandeza 10^{-2}, por exemplo. A mutação se dá da seguinte maneira:

Seja $iGer$ o número da geração atual; $x'_{ic,m}$ o valor de cada parâmetro $m \in [1,...,N_{\text{var}}]$ de um clone com índice ic numa geração $(iGer+1)$; e $v'_{ic,m}$ o resultado de um *vetor doador* que representa um parâmetro $m \in [1,...,N_{\text{var}}]$ modificado numa geração $(iGer+1)$. Então, durante a evolução do algoritmo, para cada clone gerado ic (sendo ic um número inteiro entre um e o número de clones), são selecionados três anticorpos distintos entre si com índices $ia1, ia2\ e\ ia3$; então, para cada parâmetro $m \in [1,...,N_{\text{var}}]$ deste clone ic verifica-se se um valor aleatório r_m gerado no intervalo [0,1] é menor do que uma taxa de cruzamento pré-definida CR; caso seja afirmativo, o valor do parâmetro m deste clone $x'_{ic,m}$ será modificado e será igual ao valor de $v'_{ic,m}$ calculado como apresentado abaixo na Eq. (2.8); caso contrário, o valor do parâmetro m

deste clone $x'_{ic,m}$ na geração $(iGer+1)$ será igual a próprio valor do parâmetro deste clone, $x_{ic,m}$ na geração $iGer$. Matematicamente o processo fica apresentado como se segue:

$$x'_{ic,m} = \begin{cases} v'_{ic,m}, & se\ r_m \leq CR \\ x_{ic,m}, & se\ r_m > CR \end{cases}. \tag{2.7}$$

Sejam os vetores $\boldsymbol{x}_{ia1}, \boldsymbol{x}_{ia2}\ e\ \boldsymbol{x}_{ia3}$ a representação de indivíduos escolhidos aleatoriamente na população e distintos entre si. Na geração $iGer$ um par de vetores $(\boldsymbol{x}_{ia2}, \boldsymbol{x}_{ia3})$ define uma diferença $\boldsymbol{x}_{ia2} - \boldsymbol{x}_{ia3}$. Essa diferença é multiplicada por um $F > 0$, sendo denotada por diferença ponderada, e é usada para perturbar o terceiro vetor (ou indivíduo) $\boldsymbol{x}_{ia1}$. Este processo que resulta num *vetor doador* $\boldsymbol{v'}_{ic}$ que pode ser escrito matematicamente como:

$$\boldsymbol{v'}_{ic} = \boldsymbol{x}_{ia1} \pm F \cdot (\boldsymbol{x}_{ia2} - \boldsymbol{x}_{ia3}). \tag{2.8}$$

Onde os índices aleatórios $ia1, ia2\ e\ ia3\ \in [1,...,N_{Ab}]$ são números inteiros distintos entre si; o número de anticorpos na população N_{Ab} deve ser superior a quatro; F é um número real e constante pertencente ao intervalo [0, 2] que controla a amplitude da diferença ponderada; o vetor $\boldsymbol{x} = \left(x_1, x_2, ..., x_{N_{\text{var}}}\right)^T$ representa uma solução e possui N_{var} variáveis; o sinal $\pm$ é sorteado cada vez que a operação é realizada.

Assim, os clones terão como resultado, uma espécie de interpolação entre os anticorpos existentes na população atual, permitindo o melhoramento do espalhamento das soluções no fronte de pareto.

Conforme apresentado em OLIVEIRA *et al.* (2007) os valores de F e CR usados no presente trabalho serão respectivamente 0,8 e 0,5. PRICE *et al*. (2005) utilizam em boa parte dos testes 0,9 para F e CR.

Mutação Híbrida:

Notou-se que em alguns problemas a utilização da mutação diferencial modificada resultava em estagnação da solução antes do final do processo evolutivo, e que a hipermutação somática não era capaz de alcançar um fronte de pareto homogêneo ao final da evolução. Então, utilizando-se da essência do funcionamento de cada mutação,

optou-se por criar uma mutação *híbrida* que considera o uso de duas mutações no mesmo processo evolutivo: hipermutação somática e mutação diferencial modificada.

Esta mutação simplesmente faz com que durante um percentual do número de avaliações ou gerações (dependendo do critério de parada) utilize-se a hipermutação somática e em seguida, a mutação diferencial modificada. Ou seja:

1. Primeiro, durante um determinado número de avaliações (ou gerações, dependendo da escolha do critério de parada) será realizada a *hipermutação somática*. Esta mutação visa uma ampla exploração do espaço de busca, permitindo-se a busca nas diferentes ordens de grandezas dos parâmetros.

2. Em seguida, nas avaliações (ou gerações) restantes será realizada a mutação *diferencial modificada*. Esta mutação visa aproveitar-se da capacidade de interpolação da mutação diferencial e desta forma, refinar a distribuição das soluções sobre o fronte de pareto.

Desta forma é possível combinar uma excelente capacidade de busca, pelo uso da hipermutação somática, com uma ótima distribuição do fronte, pelo uso da mutação diferencial modificada.

Para todos os testes com a mutação híbrida o critério utilizado para alternar entre a hipermutação somática e a mutação diferencial modificada foi que durante 80% das gerações (ou avaliações) utiliza-se a hipermutação somática e nos 20% restantes a mutação diferencial modificada. Estes valores foram adotados após o estudo de várias combinações de percentuais aonde se chegou à conclusão de que 80% em busca global e 20% de refinamento de espalhamento era um número razoável para gerar bons resultados. Mesmo assim cabe-se ressaltar que quaisquer outros valores poderiam ter sido adotados como critério para a alternância entre as operações de mutação.

Pode-se destacar como vantagem da mutação híbrida o fato desta permitir uma boa exploração global e local associada a uma boa capacidade de distribuição das soluções na fronteira de pareto, sendo isto feito através de uma mutação simples reunindo dois conceitos já existentes.

A seguir serão apresentados alguns algoritmos inspirados no comportamento do sistema imune natural no intuito de fornecer uma visão ampla de diferentes aplicações em otimização multi-objetivo:

MISA - *Multiobjective Immune System Algorithm* (COELLO e CORTÉS, 2002);

MOIA - *Multiobjective Immune Algorithm* (LUH, CHUEH & LIU, 2003)

MOCSA - *Multiobjective Clonal Selection Algorithm* (CAMPELO *et al.*, 2004)

VAIS - *Vector Artificial Immune System* (FRESCHI & REPETTO, 2005);

IDCMA - *Immune Dominance Clonal Multiobjective Algorithm* (JIAO, *et al.*, 2005)

IFMOA - *Immune Forgetting Multiobjective Optimization Algorithm* (LU, *et al.*, 2005)

Omni-AiNet - *Artificial Immune Network for Omni-optimization* (COELLO & VON ZUBEN, 2006);

PAIA - *Population Adaptive Based Immune Algorithm* (CHEN & MAHFOUF, 2006)

ACSAMO - *Adaptative Clonal Selection Algorithm for Multiobjective Optimization* (WANG & MAHFOUF, 2006)

CNMOIA - *Constrained Nonlinear Multiobjective Immune Algorithm* (ZHANG, 2006 e 2007);

SIAOM - *Sistema Imunológico Artificial para Otimização Multi-objetivo* (BERBERT, 2008);

EMOIA - *Evolutionary Multiobjective Optimzation Immune Algorythm* (TAN, K. C. *et al.*, 2008);

HIMO - *Hybrid Immune Multiobjective Optimization Algorithm* (CHEN, J. *et al.*, 2010);

WBMOAIS - A novel Artifical Immune System for Multiobjective Optimization (GAO, J. & WANG, J., 2010)

2.2.3.1 MISA

O algoritmo MISA (***M**ultiobjective **I**mmune **S**ystem **A**lgorithm*) foi baseado no princípio da seleção clonal, modelando a característica de que somente os anticorpos com maior afinidade irão proliferar. Neste sistema imunológico artificial, o conceito da dominância de pareto foi utilizado para gerar soluções não-dominadas. Também é utilizado um conjunto externo (chamado de *secondary memory*) para guardar as soluções não-dominadas encontradas ao longo do processo evolutivo.

O algoritmo opera da seguinte maneira: Inicialmente é gerada uma população com distribuição uniforme; o conjunto externo é inicializado vazio; determina-se a dominância dos indivíduos na população; determina-se a afinidade de cada anticorpo; os melhores serão copiados para o conjunto externo; executa-se um conjunto de operações no caso do conjunto externo ter atingido o tamanho máximo; é realizada a clonagem dos melhores anticorpos; opera-se a mutação nos clones de uma forma que os genes em cada cadeia de cromossomos sejam iguais ao número de variáveis de decisão do

problema; aplica-se uma mutação não-uniforme aos melhores anticorpos encontrados; aplica-se o cruzamento a uma fração dos anticorpos no conjunto externo; por fim, repete-se o processo desde a determinação da dominância até que se atinja um critério de parada pré-determinado.

Nota-se o uso de um conjunto externo funcionando como um mecanismo de elitismo garantindo as melhores soluções ao longo do processo.

2.2.3.2 MOIA

Em MOIA (***M**ulti**O**bjective **I**mmune **A**lgorithm*) os anticorpos são codificados em uma cadeia binária. As cadeias são de dois tipos: *cadeias leves* (*bits* menos significantes) e *cadeias pesadas* (*bits* mais significantes). Os objetivos são avaliados e são definidos índices de classificação, que medem a camada de dominância de cada solução. Os anticorpos não-dominados são escolhidos para sofrerem *hipermutação*, que ocorre somente nas *cadeias leves*, para garantir somente mutação local. Após esta fase, as células não-dominadas são guardadas na memória. A supressão e geração de novos anticorpos é um processo complexo no MOIA. Em primeiro lugar um índice cumulativo escalar, chamado avidez, é calculado como um produto da afinidade anticorpo-antígeno (o mesmo que o índice de classificação) e afinidade de anticorpos-anticorpos (o mesmo que uma medida de distância). Este índice combinado é usado para selecionar os anticorpos para a construção de bibliotecas de DNA em células germinais, que requer também um percentual pré-definido de células de memória. Então, por meio um processo de diversificação/recombinação, é criada a população para a próxima iteração.

Os Autores ampliaram seu trabalho para lidar com restrições em um artigo sucessivo (LUH & CHUEH, 2004). O problema restrito é transformado em um problema irrestrito, associando a cada anticorpo um valor de interleucina (*interleukine*). Em processos biológicos, a interleucina (*interleukine*) pode estimular ou reprimir a promoção de anticorpos. Em suas experiências *in silico*[2], a interlucina (*interleukine*) é um índice global de violação de restrição que é somado ao índice de classificação.

2.2.3.3 MOCSA

MOCSA (***M**ulti**O**bjective **C**lonal **S**election **A**lgorithm*) é uma versão codificada em números reais de uma versão multiobjetivo do algoritmo *CLONALG* (CAMPELO, *et al.,*

[2] *In silico* é uma expressão usada no âmbito da simulação computacional e áreas correlatas para indicar algo ocorrido "em ou através de uma simulação computacional". A expressão foi cunhada a partir das expressões latinas *in vivo* e *in vitro*, frequentemente usadas na Biologia. *In silico* é, originalmente, usada apenas para denotar simulações computacionais que modelam um processo natural ou de laboratório e não para cálculos computacionais genéricos.

2004). Os anticorpos são avaliados em relação aos objetivos e classificados por ranqueamento de soluções não-dominadas, ou seja, indivíduos são classificados em sucessivos frontes de soluções não-dominadas. Em seguida, os clones são gerados para toda a população. O número de clones para cada anticorpo é inversamente proporcional ao seu índice de classificação. Uma perturbação aleatória com desvio padrão fixo é adicionada aos clones, a fim de explorar localmente o espaço de busca. Toda a população de anticorpos original e clones mutados é classificada em vários frontes e são copiados para a memória. Então, é aplicada a supressão de acordo com um procedimento de nichos no espaço da função objetivo. Em cada iteração uma dada percentagem de anticorpos é gerada aleatoriamente para substituir os anticorpos com posições mais baixas na classificação, assegurando a manutenção da diversidade da população.

2.2.3.4 VAIS

O algoritmo multi-objetivo proposto por FRESCHI & REPETTO (2005) foi baseado no comportamento do sistema imunológico, especificamente no princípio da seleção clonal. O algoritmo segue as etapas padrão do algoritmo CLONALG (CASTRO *et al*., 2000) modificando o cálculo da afinidade, que passa a ser baseada no conceito de dominância de pareto. O cálculo da afinidade foi inspirado no SPEA2 (*Simple Strenght Aproach, SiSA*), para cada solução não-dominada, a afinidade é igual a uma força (*Strength*), como definida no SPEA2. Para soluções dominadas, a afinidade é o número de soluções que a dominam. Todos os anticorpos são clonados. Quanto maior a afinidade do anticorpo, menor será a taxa de mutação, indicando a existência de uma busca mais localizada em torno de soluções ótimas e uma busca mais exploratória em torno de soluções menos promissoras.

Objetivando obter um fronte de pareto mais uniformemente distribuído, aplica-se um operador de supressão, que a partir da distância Euclidiana das células de memória, reduz o número de soluções que estejam próximas uma das outras. Se for gerado um indivíduo infactível, ele é descartado. Se a infactibilidade ocorre após a mutação, reduz-se progressivamente a amplitude da mutação até que o clone se torne factível.

2.2.3.5 IDCMA

O algoritmo IDCMA (***I**mmune **D**ominance **C**lonal **M**ultiobjective **A**lgorithm*) introduz o conceito de grau diferencial do sistema imune como medida de afinidade de anticorpo-anticorpo. Este índice é usado para reduzir o tamanho de anticorpos não-dominados na memória antes da fase de clonagem. Um anticorpo é selecionado aleatoriamente a partir da memória e é usado para ativar uma nova população de anticorpos aleatórios gerados

em cada iteração. Estes anticorpos são classificados por sua afinidade com o anticorpo de ativação (com base no grau de correnpondência da cadeia binária) e estão divididos em dois grupos: uma população de *dominance clonal antibody* e uma população de *immune dominance antibody*. O primeiro será submetido a recombinação e mutação binária e é recombinado com o último conjunto e memória para fornecer a população para a próxima iteração. Como as variáveis são codificadas como *cadeias* binárias, a extensão para problemas combinatórios é simples.

2.2.3.6 IFMOA

Os mesmos autores do IDCMA propoem o algoritmo IFMOA (***I**mmune **F**orgetting **M**ultiobjective **O**ptimization **A**lgorithm*) para problemas com variáveis reais. Neste caso, a afinidade é um índice escalar que contém tanto informações de dominância de Pareto como informação sobre a densidade. A afinidade antígeno-anticorpo é medida pela força de Pareto (*pareto strength*), como na abordagem do SPEA, enquanto que a afinidade do anticorpo-anticorpo é inversamente proporcional à soma das duas menores distâncias Euclidianas entre um anticorpo e o resto da população. Um operador é introduzido para emular a não ativação de certas células quando expostas aos antígenos. Durante a evolução, parte da população é suprimida (de acordo com o valor de afinidade) e substituída por anticorpos da memória.

2.2.3.7 Omni-AiNet

COELLO & VON ZUBEN (2006) apresentam um algoritmo inspirado no sistema imune desenvolvido para resolver problemas de otimização com um ou múltiplos objetivos em situações uni ou multi-modais. Este algoritmo incorpora mecanismos apresentados pelo dopt-aiNet (*Artificial Immune Network for Dynamic Optimization*). O algoritmo apresenta um conjunto de anticorpos capaz de ajustar seu tamanho durante a execução do algoritmo. Essa redução é definida por um limiar de supressão pré-definido (que descarta indivíduos similares) e um mecanismo de grade (*grid*) que visa manter a diversidade da população garantindo um espalhamento das soluções no espaço de busca.

O algoritmo é iniciado com a geração aleatória (codificado em variáveis reais) de um repertório inicial. Os principais passos subseqüentes são: clonagem, *hipermutação* polinomial, ordenamento do repertório em classes (segundo o conceito e ϵ-dominância restrita), seleção e duplicação genética (mecanismo para introdução de variabilidade genética).

2.2.3.8 PAIA

Em seu algoritmo, CHEN & MAHFOUF (2006) tentam emular a adaptatividade de concentração de anticorpos no sangue, por meio de uma população com tamanho adaptativo. A principal característica do PAIA (***P**opulation **A**daptive **B**ased **I**mmune **A**lgorithm*) está no processo de ativação. A partir de um anticorpo escolhido aleatoriamente na população, duas medidas de afinidade são calculadas para soluções dominadas e não-dominadas. Estes valores estão relacionados com a distância do anticorpo identificado no espaço de busca e eles são definidos para garantir que os anticorpos não-dominados sempre possuam a menor afinidade. O índice definido até o momento lida com informações sobre a afinidade entre anticorpo-anticorpo e anticorpo-antígeno. O conjunto de clones é selecionado com referência à afinidade, então, um número variável de clones é gerado, especificando o número total de clones permitido. Os anticorpos não selecionados são clonados uma vez. A mutação é uma perturbação aleatória com desvio padrão proporcional à afinidade com os pais. Ao final da iteração, a supressão é aplicada à população para detecção de anticorpos cuja distância a outras soluções é maior do que um limiar pré-definido.

PAIA não implementa uma população externa para coletar células de memória, mas evolui anticorpos não-dominados mantendo-os na população, dinamicamente adaptados à complexidade do problema.

2.2.3.9 ACSAMO

Em seu algoritmo ***A**daptative **C**lonal **S**election **A**lgorithm for **M**ultiobjective **O**ptimization*, WANG & MAHFOUF (2006) mantêm traços das duas melhores soluções que atuam como antígenos: a melhor população anterior e a melhor solução global na população atual. Para identificar essas melhores soluções, é utilizada uma abordagem ponderada dinamicamente ajustada, com os pesos sendo alterados aleatoriamente em cada iteração. A afinidade de um anticorpo é então definida como o soma das distâncias Euclidianas de antígenos (melhor população anterior e a melhor solução global), medido no espaço de busca. Este índice é usado também para impulsionar a amplitude da mutação, de modo que quanto melhor o indivíduo, menor a sua mutação. A seleção das células para a próxima geração é baseada em uma classificação não-dominada (*nondominated sorting*) e, a supressão eventual de anticorpos levando em conta a distância de aglomeração (*crowding distance*). A memória externa somente é alimentada de soluções não-dominadas.

2.2.3.10 CNMOIA

Em CNMOIA (***C**onstrained **N**onlinear **M**ulti**O**bjective **I**mmune **A**lgorithm*), ZHANG (2006, 2007) calcula dois indicadores escalares para cada anticorpo: a afinidade anticorpo-antígeno e densidade de anticorpos. O primeiro valor mede a proximidade a soluções não-dominadas através da definição de um produto interno e é devidamente acoplado ao grau de violação de restrição. O segundo conta o número de soluções ao redor do anticorpo. Essas informações contribuem para criar também um índice escalar que mede o nível de estimulação de um anticorpo. O número de clones para cada anticorpo e sua probabilidade de mutação são avaliados de acordo com o valor de afinidade, enquanto que os anticorpos são dinamicamente selecionados para a próxima geração, com uma probabilidade que depende do nível de estimulação. Em CNMOIA, a população de antígenos é a população da memória.

2.2.3.11 SIAOM

BERBERT (2008) apresenta um ***S**istema **I**munológico **A**rtificial* elaborado com intuito de ser capaz de abranger problemas de ***O**timização **M**ulti-objetivos* com características que geralmente dificultam a utilização de métodos exatos: não-linearidade, não-convexidade e presença de restrições. Este algoritmo faz uso do conceito da dominância de pareto e visa guiar a busca pela Fronteira de Pareto Global e manter a diversidade das soluções não-dominadas encontradas.

Neste algoritmo o cálculo da afinidade é feito como no SPEA2, acrescido de uma consideração sobre a factibilidade da solução, alterando a forma de cálculo da afinidade em função disso, além disso, foi estabelecida uma medida para saber o quanto uma solução está próxima de ser factível chamada de *limiar de factibilidade*. A clonagem é realizada nos anticorpos factíveis e os que são infactíveis, mas se apresentam dentro de um limiar de factibilidade também serão clonados. O número de clones por anticorpo é proporcional à qualidade do anticorpo, quanto melhor, maior o número de clones, quando infactível, é gerado um número mínimo de clones. A maturação apresenta uma parcela inspirada no algoritmo PSO (*Particle Swarm Optimization*) que tenta controlar a direção da mutação numa região promissora em função da informação obtida do grupo de anticorpos. Os clones de baixa afinidade sofrerão mutações maiores, e os de alta afinidade sofrem mutações menores. O processo de seleção irá selecionar os melhores anticorpos dentro de cada grupo de anticorpos, sendo assim um método elitista e naturalmente com memória, pois a solução nunca poderá piorar. Isto ocorre, pois a seleção sucessivamente irá selecionar indivíduos melhores que os anteriores. No final do processo ocorrerá a deleção que consiste da remoção de anticorpos infactíveis e/ou

aqueles que não obtiveram melhora dentro de um determinado número de gerações pré-definidas.

Para encontrar o maior número de soluções no fronte de pareto é utilizado um arquivo externo de memória que armazena um número determinado de soluções não-dominadas factíveis. O tamanho deste conjunto é limitado e a supressão se dá baseada num limiar de aglomeração que utiliza uma distância de aglomeração.

De certo, um dos grandes inconvenientes deste algoritmo é o grande número de parâmetros a se ajustar em virtude de estratégias de controle de tamanho de população e de diversidade.

2.2.3.12 EMOIA

Neste trabalho TAN *et al.* (2008) apresentam um sistema imunológico artificial evolucionário para otimização multi-objetivo combinando a capacidade de busca global dos algoritmos evolucionários com a capacidade de aprendizado do sistema imunológico artificial.

O algoritmo incorpora características da seleção clonal e de memória do sistema imune para melhorar a busca do algoritmo evolucionário identificando potenciais regiões para exploração e ao mesmo tempo evitando enfatizar exageradamente alguma região do espaço de busca. Um esquema baseado na entropia é utilizado para estimativa da densidade para se distribuir uniformemente as soluções não-dominadas ao longo do fronte de pareto descoberto.

Os estímulos antigênicos e os anticorpos são definidos respectivamente como o problema a se resolver e as potenciais soluções ao problema. A afinidade dos anticorpos está associada a quão bem este resolve o problema e é definida em termos de pareto dominância. A memória do sistema imune foi implementada na forma de um arquivo de tamanho fixo e as células de memória representam os anticorpos não-dominados. De forma a promover a diversidade e facilitar a troca de informação entre os anticorpos, estes são submetidos a operadores genéticos como cruzamento e mutação após o processo da seleção clonal e expansão.

O processo evolutivo se inicia com a geração aleatória de soluções para compor a população de anticorpos. Em seguida os anticorpos são avaliados (baseados nas respectivas funções objetivos e dominância). Os anticorpos são copiados para o arquivo como células de memória. O arquivo é atualizado a cada ciclo (somente soluções não-dominadas são adicionadas ao conjunto de células de memória, e todas as soluções dominadas são removidas do arquivo). Quando o tamanho máximo do arquivo é atingido executa-se um procedimento para truncamento baseado na entropia de cada solução

(EDAS - *Entropy-based Density Assessment Scheme*) onde as soluções menos efetivas são removidas do arquivo. Após o arquivamento de soluções, anticorpos apropriados são selecionados para uma população intermediária que tem o mesmo tamanho da população corrente. O processo de seleção se dará em dois estágios: O primeiro envolve a clonagem das células de memória a partir do arquivo por meio de um esquema de seleção clonal, estas comporão parte das soluções na população intermediária. O segundo envolve uma seleção por torneio binário na população corrente para compor o restante das soluções no conjunto de reprodução. Esta seleção por torneio é realizada com base num esquema de dominância-restrita (*constrained-dominance*). Este conjunto é então submetido a cruzamento e mutação uniforme. O processo é repetido até que o número máximo de avaliações seja atingido.

O algoritmo foi testado em sete funções de teste *benchmark*: ZDT1, ZDT4, ZDT6, FON, KUR, DTLZ3 (5 objetivos), CTP7 (com restrição). Os resultados se mostraram bastante efetivos em termos de convergência, diversidade e distribuição no fronte.

2.2.3.13 HIMO

Neste trabalho, CHEN *et al.* (2010) desenvolveram um algoritmo hibrido multi-objetivo baseado no princípio da seleção clonal do sistema imune. Em HIMO, foi proposto um operador de mutação hibrido que é uma combinação de uma mutação Gaussiana e uma mutação Polinomial (GP-HM). O operador GP-HM adota uma mudança seletiva adaptativa entre as mutações Gaussiana e Polinomial, visando aumentar a velocidade de convergência em direção ao fronte de pareto. O operador GP-HM combina as vantagens da mutação Gaussiana (mutação mais grosseira, ou seja, mais global) com as vantagens da mutação Polinomial (mutação mais refinada, ou seja, mais local).

O algoritmo HIMO possui as seguintes etapas: a população é inicializada aleatoriamente; em seguida é avaliada e os indivíduos não-dominados são colocados no arquivo externo; um operador de clonagem é aplicado, clonando-se indivíduos proporcionalmente à sua afinidade, quanto maior a afinidade, maior o número de clones por indivíduo; é aplicado um operador de recombinação (SBX – *Simulated Binary Crossover*) capaz de escapar de ótimos locais; aplica-se o operador GP-HM e em seguida os indivíduos são avaliados e as soluções não-dominadas são armazenadas num arquivo externo. Repete-se o processo até que atinja um critério de parada.

O algoritmo foi testado em funções matemáticas de benchmark (ZDT1, ZDT2, ZDT3, ZDT4, ZDT6) apresentando resultados promissores quanto a convergência do algoritmo e resultados razoáveis quanto a diversidade do fronte obtido comparados aos algoritmos NSGA-2, PAES, SPEA e SPEA2;

2.2.3.14 WBMOAIS

O trabalho de GAO & WANG (2010) apresenta um algoritmo multi-objetivo baseado em sistema imunológico artificial, especificamente no algoritmo do *opt-aiNET*. O WBMOAIS segue a estrutura elementar do *opt-aiNET,* mas possui as seguintes características distintivas: **(1)** uma soma aleatória de pesos de múltiplos objetivos é usada como uma função de custo ou *afinidade*. Como resultado, a tarefa de atribuição da *afinidade* requer uma complexidade computacional muito menor que a baseada na classificação em pareto dominância; **(2)** os indivíduos da população são obtidos a partir de uma memória, que é constituída por um conjunto de soluções elite, e um procedimento de busca local é utilizado para facilitar a exploração do espaço de busca; e **(3)** em adição ao algoritmo da supressão clonal semelhante ao usado em *opt-aiNET,* um novo algoritmo de truncamento é apresentado buscando-se eliminar indivíduos similares na memória e obter-se um fronte com um bom espalhamento de soluções não-dominadas.

O algoritmo WBMOAIS é comparado com outros algoritmos como VIS (*Vector Immune Algorithm*), NSGA-II (*Elitist Non-dominated Sorting Genetic Algorithm*) que são representativos do estado da arte. As simulações foram executadas em sete funções matemáticas de benchmark (ZDT6, SCH2, DEB, KUR, POL, FON, e VNT). Os resultados do algoritmo WBMOAIS foram superiores aos algoritmos comparados, mostrando que o algoritmo pode ser uma alternativa aos algoritmos tradicionalmente utilizados para a resolução de problemas multi-objetivos.

2.3 MEDIDAS DE DESEMPENHO

Para a comparação dos resultados dos algoritmos evolutivos em otimização multi-objetivo, foi necessário o desenvolvimento de métricas capazes de medir características desejadas dos algoritmos desenvolvidos. Neste sentido, existem basicamente duas metas que um algoritmo multi-objetivo deve alcançar (ver Figura 2.8):

(i). (convergência) = Encontrar as soluções mais próximas possíveis ao verdadeiro fronte de pareto; e

(ii). (diversidade) = Que as soluções encontradas estejam o mais espalhadas possível, de forma a cobrir toda a região do verdadeiro fronte de pareto.

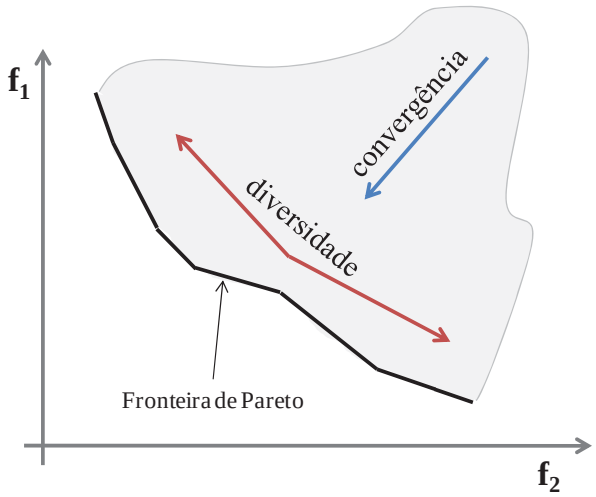


Figura 2.8: Duas metas da otimização multi-objetivo

De alguma forma, os dois objetivos podem ser considerados *ortogonais*. O primeiro requer uma busca em direção à região pareto-ótima, enquanto o segundo requer uma busca ao longo do fronte pareto-ótimo como mostrado na Figura 2.8.

A Figura 2.9a apresenta um caso hipotético onde os resultados do algoritmo A tem boa convergência, mas possuem pouca diversidade, ao contrário do B (Figura 2.9b) que possui uma convergência ruim, mas boa diversidade. Esta é uma situação que ilustra a necessidade de pelo menos duas medidas de desempenho para poder se comparar o resultado de dois algoritmos distintos.

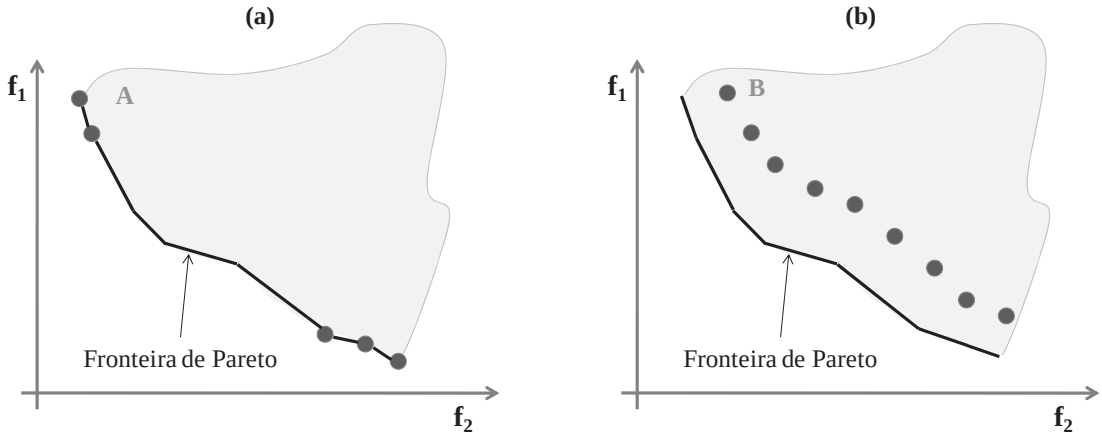


Figura 2.9: Fronte de pareto com **(a)** boa convergência; **(b)** boa diversidade

Outra situação pode ser ilustrada na Figura 2.10. No primeiro caso, Figura 2.10a, é fácil visualizar que o algoritmo A é melhor do que o B. No entanto, no segundo caso, Figura 2.10b, é difícil determinar qual algoritmo tem o melhor desempenho. A comparação dependerá fortemente da métrica utilizada.

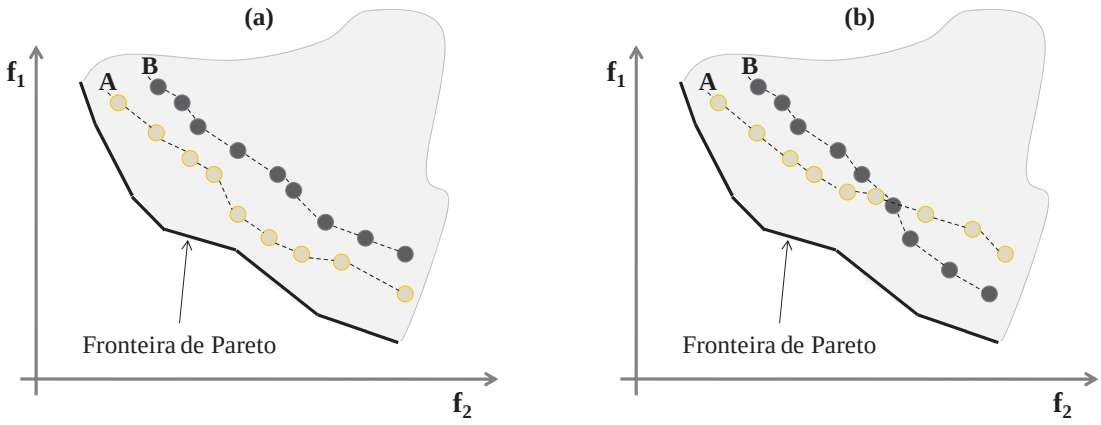


Figura 2.10: Comparação entre frontes de pareto

A seguir serão apresentadas algumas métricas de desempenho classificadas em dois grupos: convergência e diversidade.

2.3.1 Métricas de Convergência

Este tipo de métrica computa explicitamente a distância de um conjunto Q de NQ soluções a um conjunto conhecido de soluções pareto-ótimo *P** (DEB, 2001). Estas métricas são aplicáveis **somente** quando o **conjunto pareto-ótimo é conhecido**.

2.3.1.1 Taxa de Erro (ER)

Esta métrica conta as soluções de Q que não estão no conjunto de soluções pareto-ótimas *P** (VELDHUIZEN, 1999):

$$ER = \frac{\sum_{iq=1}^{|Q|} e_{iq}}{|Q|} \tag{2.9}$$

Onde, e_{iq}=0 se $iq \in P^*$, e $e_{iq} = 1$, caso contrário. Quanto menor o valor de *ER*, melhor será a convergência. Se *ER = 0*, significa que $Q \subseteq P^*$. Se *ER = 1* significa que nenhuma solução pertence a *P**. Uma desvantagem desta métrica é que embora uma solução de Q seja pareto-ótima, se esta solução não existe em *P**, a solução deve ser contada na Eq (2.9) como sendo não pareto-ótima. Com isso o conjunto *P** deverá ser suficientemente grande para representar o máximo de soluções no fronte de pareto.

2.3.1.2 Distância Geracional (GD)

Esta métrica encontra uma distância média entre as soluções de Q e *P** (VELDHUIZEN, 1999):

$$GD = \frac{\left(\sum_{iq=1}^{|Q|}\left(d_{iq}\right)^{p}\right)^{1/p}}{|Q|} \tag{2.10}$$

Para *p = 2*, o parâmetro d_{iq} é a distância Euclidiana (no espaço da função objetivo) entre uma solução $iq \in Q$ e o indivíduo mais próximo de *P**: Foi utilizada nesta tese ***p = 2****,* pois se entende que a distância Euclidiana é suficiente para representar a distância entre as soluções.

$$d_{iq} = \min_{ie=1}^{|P^*|} \sqrt{\sum_{i=1}^{N_{ob}} (f_i^{(iq)} - f_i^{*(ie)})^2} \tag{2.11}$$

Onde, $f_i^{*(ie)}$ é o valor do i*-ésimo* objetivo da ie*-ésima* solução de *P** e $f_i^{(iq)}$ é o valor do i*-ésimo* objetivo da iq*-ésima* solução de Q.

A dificuldade com esta métrica consiste no fato de que se existe conjunto Q para o qual há uma grande flutuação na medida de distância, a métrica pode não revelar a distância real. Neste caso será necessário a medida da variância de *GD*. Além disso, se as funções objetivo diferirem em magnitude, elas deverão ser normalizadas antes de calculadas as distâncias. Para tornar os cálculos confiáveis recomenda-se um grande número de soluções em *P** (DEB, 2001).

Quanto menor o valor de *GD*, melhor a convergência das soluções encontradas do conjunto de soluções fornecidas *P**.

2.3.2 Métricas de Diversidade

Estas métricas computam a distribuições das soluções de um conjunto Q.

2.3.2.1 Espaçamento (SP)

SCHOTT (1995) sugeriu uma métrica em que é calculada a distância relativa entre soluções consecutivas do conjunto obtido de soluções não-dominados.

$$SP = \sqrt{\frac{1}{|Q|}\sum_{iq=1}^{|Q|}(dm_{iq} - \overline{dm})^2} \tag{2.12}$$

Onde,

$$dm_{iq} = \min_{iq \in Q \; e \; iq \neq i} \sum_{i=1}^{N_{ob}} \left| f_i^{(iq)} - f_i^{*(ie)} \right|, \text{ e } \overline{dm} = \frac{\sum_{iq=1}^{|Q|} dm_{iq}}{|Q|} \tag{2.13}$$

A variável dm_{iq} representa o mínimo somatório do valor absoluto da diferença dos valores das funções objetivo entre a solução iq e qualquer outra solução em Q. Esta distância é diferente da distância Euclidiana mínima entre duas soluções. Por outro lado, $\overline{dm}$ representa a média dos valores de dm_{iq}. Quanto menor for o valor de *SP*, melhor distribuídas estarão as soluções em Q (DEB, 2001).

2.3.2.2 Espalhamento (Δ)

DEB *et al.* (2000a) propôs uma métrica onde: D_{iq} pode ser qualquer métrica de distância (distância Euclidiana ou distância de aglomeramento, por exemplo) entre soluções vizinhas e $\overline{D}$ é o valor da média destas distâncias; D_i^e é a distância entre soluções extremas de *P** e Q correspondendo à i*-ésima* função objetivo.

$$\Delta = \frac{\sum_{i=1}^{N_{ob}} D_i^e + \sum_{iq=1}^{|Q|} \left| D_{iq} - \overline{D} \right|}{\sum_{i=1}^{N_{ob}} D_i^e + \left| Q \right| \overline{D}} \tag{2.14}$$

O presente trabalho utilizou a distância Euclidiana. Conforme as soluções se aproximam da distribuição ideal, o valor de Δ se aproxima de zero.

Capítulo 3 O ALGORITMO DESENVOLVIDO

Equation Chapter (Next) Section 1 Neste capítulo serão desenvolvidos os temas relacionados ao desenvolvimento do algoritmo multi-objetivo. Em primeiro lugar será apresentado o fluxograma geral do algoritmo e em seguida cada uma de suas etapas serão discutidas em detalhes.

O algoritmo desenvolvido é denominado SIAMEP acrônimo para **S**istema **I**munológico **A**rtificial com **M**emória **E**xterna de **P**areto. Pois diferentemente do SIA *CLONALG*, sua evolução dependerá agora, de uma memória que está externa à população e não mais da evolução de cada grupo de anticorpos independentes como era com a seleção em família.

3.1 ALGORITMO SIA MULTI-OBJETIVO

O algoritmo SIAMEP foi desenvolvido a partir do SIA uni-objetivo e também inspirando-se em conceitos de outros algoritmo já existentes no estado da arte, conforme será apresentado a seguir. Com o intuito de tornar o entendimento mais simples, inicialmente é apresentada uma visão geral (ver Figura 3.1) do algoritmo, e em seguida, as etapas do algoritmo serão pormenorizadas.

Pode-se dizer que as características mais importantes dos algoritmos evolutivos são: o *operador de seleção* e *o operador de exploração do espaço de busca*. O primeiro porque o operador de seleção é a ferramenta capaz de identificar, entre as soluções apresentadas, aquela solução que é a mais adequada em respeito a uma determinada característica perseguida. O segundo, porque para se conseguir novas soluções melhores que as anteriores, o algoritmo precisa ter a capacidade de explorar corretamente o espaço de busca, e quem responde por esta função é o operador de exploração. Exemplos de operadores de exploração do espaço de busca são: mutação, cruzamento associado com a mutação, etc. Em relação ao algoritmo *CLONALG* uni-objetivo apresentado no item 2.2.3 pode-se destacar as seguintes características:

1. *Mutação proporcional à qualidade da solução* (afinidade) – esta permite que haja mais exploração local em torno de soluções promissoras e mais exploração global em torno de soluções não tão boas. O algoritmo faz isso variando a taxa de mutação, ver Eq. (2.4).
2. *Seleção por família* – esta permite que um grupo de anticorpos tenha sua evolução independente dos demais grupos de anticorpos (ou famílias). Ou seja, durante o processo evolutivo, a informação de um grupo de anticorpos não

influencia o outro, permitindo que estes evoluam em direções distintas, fazendo com que o algoritmo cubra bem o espaço de busca.

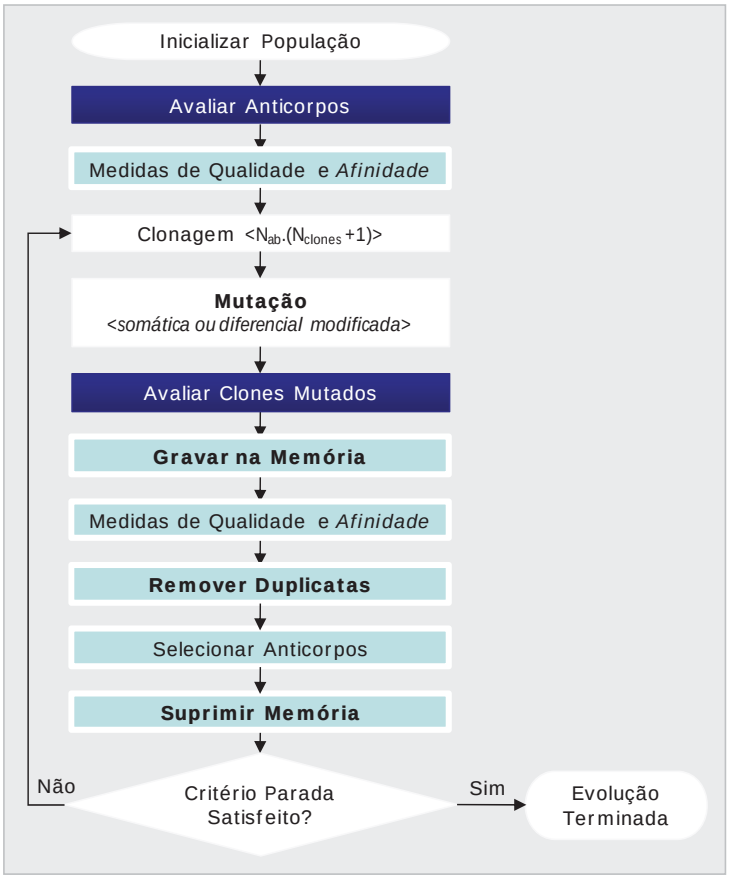


Figura 3.1: Fluxograma do Algoritmo SIAMEP

Para que o algoritmo SIA consiga gerar novas soluções sucessivamente melhores do que as anteriores, é fundamental que haja distinção entre as soluções, ou seja, que a medida de qualidade (afinidade) represente suficientemente bem o objetivo perseguido pelo algoritmo (convergência e diversidade). Desta forma, durante a execução do algoritmo, algumas soluções têm que ser melhores do que outras para que as soluções geradas pelo algoritmo possam ser progressivamente melhoradas. Então, surge o primeiro problema na construção do algoritmo SIA multi-objetivo. As soluções do fronte de pareto não possuem distinção de qualidade, todas, por definição, são igualmente boas.

Nos itens seguintes estão apresentadas como são feitas todas as etapas do algoritmo desenvolvido e como foi possível contornar os problemas apresentados.

3.1.1 Inicializar População

Deve-se inicializar uma população de soluções contendo N_{ab} Anticorpos. A população de soluções iniciais é gerada aleatoriamente. Para cada parâmetro (variável) x_m, $m \in [1,...,N_{\text{var}}]$, ($N_{\text{var}}$ sendo o número de parâmetros, ou variáveis livres) é gerado

3.1.2 Avaliar Anticorpos

Nesta etapa serão calculados os valores das funções objetivos e a violação das restrições para cada um dos anticorpos existentes na população.

3.1.3 Medidas de qualidade e Afinidade

Conforme descrito anteriormente, para o algoritmo SIA poder evoluir, é necessário haver distinção entre as soluções. Segundo a estratégia de pareto uma característica de distinção é se a solução é dominada ou não-dominada. No entanto, esta é *uma relação restrita de distinção* que não faz distinção entre os elementos dentro de um mesmo grupo. Ou seja, entre as soluções não-dominadas, qual é a melhor? Sabe-se que as soluções não-dominadas são, por definição, igualmente boas, mas para o processo evolutivo é necessário haver distinção e medição de qualidade para guiar a busca em direção ao ótimo. Mas esta classificação já é capaz de gerar um primeiro nível de distinção: soluções dominadas e soluções não-dominadas.

Objetivando-se obter uma melhor distinção entre as soluções e inspirando-se em FONSECA & FLEMING (1993) utilizou-se aqui o conceito de classificação em camadas de pareto para uma primeira distinção entre os elementos. Isto se dá da seguinte maneira:

Verifica-se a dominância das soluções para toda a população corrente, então, aquelas soluções que primeiramente foram classificadas como não-dominadas são classificadas como sendo pertencentes da camada um, e removidas da população. Em seguida, o processo de verificação da dominância é executado novamente, com a população restante e aquelas soluções que obtiverem o status de não-dominadas serão agora pertencentes à camada dois e são removidas da população. O processo é repetido até que não ocorram mais modificações nas características de dominação das soluções indicando que se atingiu a última camada.

Este processo pode ser ilustrado pela Figura 3.2 que mostra o resultado de uma iteração do algoritmo, onde estão impressas as soluções num gráfico onde o eixo x representa o valor de f_1 para as soluções encontradas e o eixo y representa o valor de f_2 para as soluções encontradas. Neste caso supondo um problema multi-objetivo de minimização com duas funções objetivo f_1 e f_2.

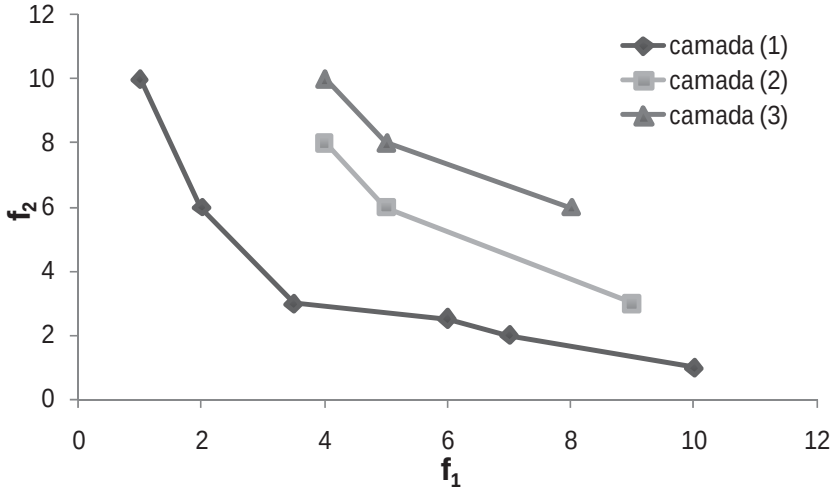


Figura 3.2: Exemplo de *camadas de pareto*

Tendo em vista melhorar a distinção entre as soluções na população, foi utilizado, de maneira simplificada, o conceito de força (*strength*) de uma solução conforme descrito em ZITZLER *et al.* (1998) e ZITZLER *et al.* (2001). A Eq. (3.1) apresentada a seguir resume tal conceito:

$$s_{ip} = \left(\frac{nDom_{ip}}{nTotDom + 1} \right) \tag{3.1}$$

Onde s_{ip} representa a força de uma solução ip; $nDom_{ip}$ é o número de soluções dominadas pela solução ip; $nTotDom$ é o número total de soluções dominadas por todas as soluções, ou seja, é o somatório do número de soluções dominadas por cada uma das soluções. O conceito aqui é utilizado de forma simplificada, pois é utilizado em conjunto com outras características de distinção e no trabalho de ZITZLER *et al.* (1998, 2001) esta é a única forma de distinção encontrada.

Quanto maior o valor de s_{ip}, mais dominante é o indivíduo, ou seja, mais indivíduos ele domina, indicando que ainda há indivíduos em camadas de número maiores que um. Como o objetivo é que, ao final da evolução, todas as soluções façam parte do fronte de pareto, então, ter muitas soluções pertencendo a camadas maiores do que um, não é bom. Quando todas as soluções apresentam a parcela s_{ip} com valor reduzido, ou igual a zero, há poucas soluções dominadas ou nenhuma, indicando que as soluções estão distribuídas no fronte de pareto (ZITZLER *et al.*,1998).

No intuito de medir a qualidade da diversidade das soluções no fronte de pareto, foi utilizado o conceito de densidade estabelecido em DEB *et al.* (2000a) e ZITZLER *et al.* (2001) como elemento de distinção entre as soluções. Mesmo durante a evolução do

algoritmo, é interessante que as soluções já apresentem diversidade, permitindo que a evolução cubra toda a região que define a fronteira de pareto. Sendo assim, a medida de densidade, irá permitir que as soluções mais espalhadas façam parte do fronte de pareto. Isso se dá da seguinte maneira:

Para se obter uma estimativa da densidade das soluções ao redor de uma solução particular ip na população, calcula-se a distância média aos dois pontos mais próximos de cada lado deste ponto particular ip ao longo de cada objetivo. Esta medida de distância dp_{ip} serve como estimativa do Hiper-cubo (Figura 3.3*)* formado usando-se como vértices os vizinhos mais próximos. Esta distância é chamada de *distância de agrupamento* (ou *crowding distance*). Na Figura 3.3, a distância de agrupamento da ip-*ésima* solução do fronte (marcado em círculos cinza escuro cheios) é a média da lateral do Hiper-cubo (marcado em tracejado). Na figura, os círculos cinza escuro cheios pertencem à mesma camada de pareto.

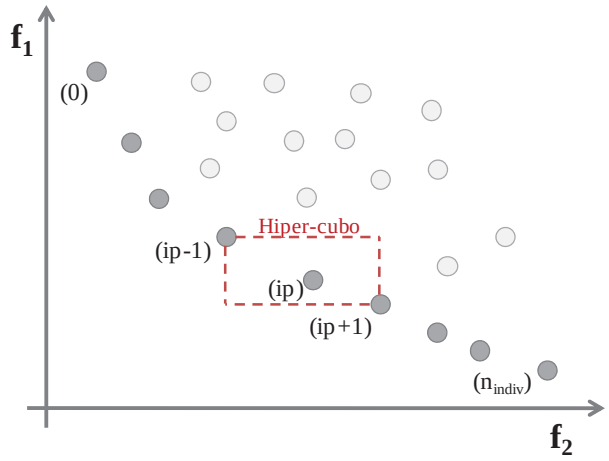


Figura 3.3: Cálculo da *distância de agrupamento* – modificado de DEB *et al.* (2000a).

O cálculo da *distância de agrupamento* (doravante, CD – *Crowding Distance)* requer o ordenamento da população de acordo com o valor de cada função objetivo em ordem crescente. Posteriormente, para cada função objetivo, as soluções da fronteira (soluções com menores e maiores valores de função objetivo) recebem a distância igual a infinito. Todas as demais soluções intermediárias recebem a distância dada pelo valor absoluto da diferença normalizada do valor das funções objetivos entre duas soluções adjacentes. A *distância de agrupamento* geral é calculada pela soma das distâncias individuais correspondentes a cada objetivo para cada indivíduo. O valor de cada função objetivo é normalizado antes de calcular a *distância de agrupamento* e esta distância será usada para a estimativa da densidade em torno de cada solução. O algoritmo apresentado a seguir ilustra o cálculo (DEB *et al.,* 2000a):

Procedimento 1: Cálculo da *distância de agrupamento*

01: **Para** cada solução ip , no conjunto de soluções, **faça**

02: $CD_{ip} = 0$

03: **Fim** do **Para**

04: **Para** cada função objetivo i , **faça**

05: Ordenar população pelo valor da função objetivo i

06: Atribuir $CD_1 = CD_{n_{indiv}} = \infty$ [soluções da borda do intervalo]

07: **Para** as soluções de $ip = 2$ até $(n_{indiv} - 1)$, **faça**

08: $$CD_{ip} = CD_{ip} + \frac{\left[f_i(ip+1) - f_i(ip-1) \right]}{f_i^{\max} - f_i^{\min}}$$

09: **Fim** do **Para**

10: **Fim** do **Para**

11: **somar**, para cada solução ip , os valores de CD_{ip} em cada objetivo i

Onde, CD_{ip} refere-se ao valor da distância média para a solução mais próxima no espaço da função objetivo em torno de cada solução; $f_i(ip+1)$ refere-se ao valor da função objetivo de $1...N_{ob}$ para o indivíduo posterior ao ip*-ésimo*; $f_i(ip-1)$ refere-se ao valor da função objetivo de $1...N_{ob}$ para o indivíduo anterior ao ip*-ésimo*; $f_i^{\max}$ refere-se ao máximo valor de função objetivo encontrado para os objetivos de $1...N_{ob}$; e $f_i^{\min}$ refere-se ao mínimo valor de função objetivo encontrado para os objetivos de $1...N_{ob}$.

LUO *et al.* (2008) notaram que esta distância não considera a diferença existente entre os espalhamentos em cada uma das funções objetivos. Para melhor compreensão veja a Figura 3.4 a seguir.

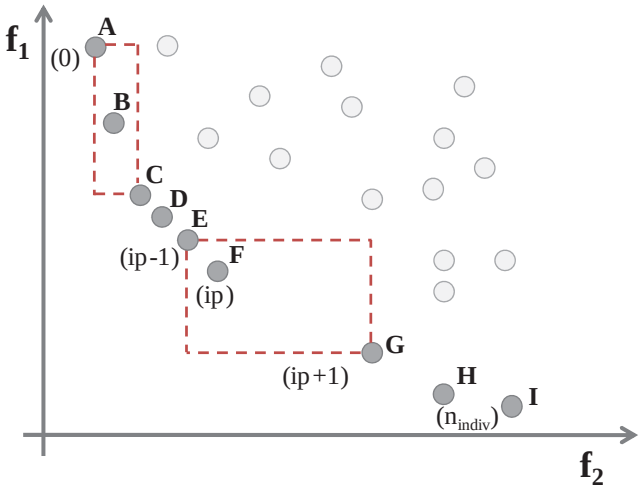


Figura 3.4: *Distância de agrupamento de indivíduos* – modificado de LUO *et al.* (2008)

Seja uma população de tamanho n_{indiv}, o número de soluções não-dominadas na população é $nNdom$, se $nNdom > n_{indiv}$, então o algoritmo deverá usar um operador de truncamento para reduzir ($nNdom - n_{indiv}$) indivíduos do conjunto de soluções não-dominadas. Nessa redução, pode-se levar em conta o valor da distância de agrupamento para a escolha de quem deve ser removido. A partir da Figura 3.4, dois problemas básicos podem ser percebidos no uso da distância de agrupamento (*CD*) para remover indivíduos da população: (1) devido ao fato da *CD* dos indivíduos C, D e E ser pequena, todos estes seriam removidos da população. Resultando em um fronte não uniforme. (2) o valor do *CD* de alguns indivíduos, como o B, é pequeno devido ao fato de que um dos lados do retângulo é pequeno, embora o outro lado seja grande. Entretanto, o *CD* de F será grande, pois o comprimento das laterais do retângulo são quase iguais. Então, se um indivíduo tivesse que ser removido entre B e F, o removido seria o B, no entanto a diversidade de B é maior do que a de F (LUO *et al*., 2008).

Assim sendo, optou-se pelo cálculo de uma distância de agrupamento dinâmica (*Dynamic Crowding Distance*, doravante - *DCD*) conforme proposto por LOU *et al*. (2008). O cálculo da DCD se dá da seguinte maneira:

$$DCD_{ip} = \frac{CD_{ip}}{\log(1/V_{ip})} \tag{3.2}$$

Onde CD_{ip} é a distância de agrupamento como calculada como no Procedimento 1; V_{ip} está descrito na expressão (3.3) a seguir e representa a variância de CD_{ip} dos indivíduos

vizinhos do ip-*ésimo* indivíduo. V_{ip} é capaz de fornecer informação sobre o grau da diferença entre o CD_{ip} em cada um dos objetivos.

$$V_{ip} = \frac{1}{N_{ob}} \sum_{i=1}^{N_{ob}} \left(\left| f_i(ip+1) - f_i(ip-1) \right| - CD_{ip} \right)^2 \tag{3.3}$$

Onde N_{ob} é o número de objetivos, $f_i(ip)$ é o valor do i-*ésimo* objetivo do ip-*ésimo* indivíduo.

Para resolver o problema (1), a cada remoção de um indivíduo, o valor de *DCD* de todos os restantes é recalculado. E para resolver a falha (2) o cálculo de *DCD* é feito como apresentado em (3.2) considerando a variância. Assim, os resultados serão superiores aos apresentados simplesmente utilizando-se a distância de agrupamento (*CD*).

A densidade será então o inverso do valor de DCD_{ip} para cada indivíduo.

$$densidade_{ip} = \frac{1}{(1 + DCD_{ip})} \tag{3.4}$$

Após todos os indivíduos possuírem um valor para estimativa da densidade pode-se efetuar a comparação entre aqueles de mesma camada, para identificar os que agregam maior informação ao conjunto. Uma solução com menor valor de *DCD* indica que, de algum modo, está mais densamente cercada por outras soluções, e a solução com maior valor de *DCD* estaria menos cercada. Logo, se o objetivo é homogeneizar a distribuição de soluções no fronte, deve-se minimizar a *DCD* para povoar as regiões não povoadas com soluções.

A **afinidade** é finalmente calculada a partir dos dados obtidos acima.

$$afinidade_{ip} = \left[s_{ip} \right] + \left[densidade_{ip} \right] + \left[camada_{ip} \right] \tag{3.5}$$

Comentários acerca desta função de *afinidade*:

- A parcela s_{ip} é proporcional ao número de soluções dominadas por ip, quando esta parcela possui um valor reduzido, esta indica que não há soluções inferiores a ela levando a formação da fronteira de pareto.

- A parcela $densidade_{ip}$ é a densidade entre as soluções mais próximas de ip, quando esta possui um valor reduzido indica que há uma baixa densidade resultando em boa informação, quando o valor é grande, indica que há um espaço muito povoado resultando em pouca informação.
- A parcela $camada_{ip}$: indica que quanto maior a camada, maior será a afinidade, logo, pior será a solução num problema de minimização.
- Nota-se que esta somente funcionará para **problemas de minimização**, pois o melhor valor de afinidade será o menor.
- Seria possível a adoção de pesos individuais para cada uma das parcelas desta função, mas acredita-se que isto traria uma dificuldade adicional desnecessária ao algoritmo;

Resumidamente, nesta etapa verifica-se a dominância em camadas na população *de memória*, é calculado o valor do *DCD* de cada solução e sua respectiva densidade, é calculado o valor da força (*strength*) de cada solução. E por fim calcula-se a *afinidade*.

3.1.4 Clonagem

Nesta etapa, toda a população de anticorpos é selecionada para a realização da clonagem. A clonagem gera cópias idênticas de cada anticorpo. A população nesta etapa encontra-se com o tamanho igual a: $\left(N_{clones}+1\right)N_{ab}$. Onde N_{clones} é igual ao número de clones e N_{ab} é igual ao número de anticorpos. Analisando-se sistematicamente nota-se que a população é aumentada para realizar a exploração do espaço de busca.

3.1.5 Mutação

A mutação é a etapa responsável pela geração de novas soluções, ou seja, representa, o operador de busca. Este operador é responsável pela modificação dos clones gerados permitindo a realização da busca local e global. A mutação poderá ser de vários tipos, por exemplo: hipermutação somática, diferencial modificada, híbrida, etc.

3.1.6 Avaliar Clones Mutados

Nesta etapa serão calculados os valores das funções objetivos e a violação das restrições para cada um dos clones que sofreu a mutação.

3.1.7 Gravar na Memória

No algoritmo do SIA uni-objetivo, a memória da população pode ser simplesmente a própria população de anticorpos. Isto ocorre porque a seleção, como processo elitista, não permite que as soluções sejam degeneradas durante a evolução do algoritmo, então,

a solução atual de cada anticorpo representa a melhor solução encontrada até então para aquele anticorpo. Esta forma de abordagem funciona corretamente quando se deseja encontrar apenas uma solução ótima. No caso do problema multi-objetivo, onde se deseja encontrar um conjunto de soluções – fronte de pareto, não é suficiente que a memória fique simplesmente na população corrente. A criação do conjunto externo de pareto com as soluções do conjunto de pareto demanda também a criação de operadores para controlar o tamanho deste conjunto, que poderia crescer excessivamente durante o processo evolutivo.

O algoritmo desenvolvido utiliza-se de conjunto externo de pareto e o procedimento para alocação dos indivíduos neste conjunto é simples: após a etapa da mutação, todos os indivíduos da população serão colocados no conjunto de pareto e o algoritmo passará a realizar todas as operações em cima deste conjunto externo de pareto, e não mais na população de anticorpos + clones. Após a seleção, o algoritmo volta a utilizar a população de anticorpos para suas operações.

3.1.8 Remover Duplicatas

Depois de alocados no conjunto externo de pareto, a população ainda pode conter indivíduos muito próximos ou idênticos, por isso realiza-se uma simples verificação de proximidade pela distância Euclidiana entre todos os indivíduos para verificar se há dois indivíduos no mesmo lugar físico, ou muito próximos. Caso isso ocorra, uma das cópias é removida. O algoritmo aceita uma tolerância na proximidade na ordem de *0,0001*, ou seja, indivíduos com distância inferior a este número serão considerados indivíduos iguais e serão removidos da população.

3.1.9 Selecionar Anticorpos

Depois de reunidas as características de distinção estabelecidas anteriormente (item 3.1.3), é necessário escolher a melhor solução, ou seja, a definição de um *operador de seleção*.

No item 2.2.3 foi esclarecido como ocorre, entre outras coisas, a seleção no SIA uni-objetivo. Para funcionar dentro de um novo contexto de comparação que é o multi-objetivo, a seleção teve que ser modificada em relação ao SIA uni-objetivo.

A seleção em primeiro lugar levará em conta se a solução é dominada ou não-dominada. O alvo são as soluções não-dominadas. Para isto temos a medida da camada de pareto, qualquer indivíduo que seja da camada um será não-dominado. Com isso, um dos objetivos é sempre alcançar as soluções na primeira camada de pareto. Se todas as soluções forem da primeira camada de pareto? Então é acionado o segundo elemento de comparação: *a densidade*. O objetivo é encontrar as soluções menos aglomeradas, ou

seja, que estão localizadas em regiões menos densas do espaço de busca, para privilegiar lugares ainda não explorados, com isso, escolhem-se as de maior *DCD* (ou seja, de menor densidade).

Resumidamente, o operador de seleção busca as seguintes características para o caso de problemas sem restrição:

- Soluções da menor camada de pareto; e
- Soluções de menor densidade (ou maior valor de *DCD*);

Caso o problema possua restrições, o operador de seleção será diferenciado. Em primeiro lugar, este deverá privilegiar as soluções totalmente viáveis, ou seja, aquela em que o somatório da violação das restrições é igual a zero. Em seguida será feito como descrito anteriormente. No caso de todas as soluções apresentadas serem não-viáveis, serão privilegiadas aquelas com o menor valor do somatório da violação das restrições.

Resumidamente, o operador de seleção busca as seguintes características para o caso de problemas com restrição:

- Soluções viáveis (ou com menor somatório de violação das restrições);
- Soluções da menor camada de pareto; e
- Soluções de menor densidade (ou maior valor de *DCD*);

O procedimento da seleção pode ser escrito da seguinte maneira:

Procedimento 2: Operador de seleção

01: **Para** cada anticorpo $ip = 1$ até N_{ab}, **faça** [percorrer anticorpos]

02: **Se** *contaViavel > 0,* **então** [conta o número de soluções viáveis]

03: *iPos* [é uma função que retorna a posição de uma solução no conjunto de soluções da memória externa que atenda a característica de *ser viável; camada = 1; possuir menor valor de densidade*] <valores entre 0 e o número de indivíduos no conjunto pareto>

04: **Se** *iPos = -1*, **então** [nenhum indivíduo foi encontrado com as características desejadas]

05: *iPos* [soluções da memória externa que atenda a característica de *ser viável; camada > 1; possuir menor valor de densidade*] <valores entre 0 e o número de indivíduos no conjunto pareto>

06: **Fim** do **Se**

07: **Caso contrário** [problemas com restrição]

08: *iPos* [soluções da memória externa que atenda a característica de ter o menor somatório de violação das restrições] <valores entre 0 e o número de indivíduos no conjunto pareto>

09: **Fim** do **Se**

10: *selecionaIndiv* [indivíduo na posição *iPos* no conjunto de pareto é selecionado para ser anticorpo na geração subseqüente]

03: **Fim** do **Para**

Desta forma percorre-se a população de anticorpos (que está na memória externa) procurando-se constituir a nova população com o tamanho igual ao número de anticorpos (previamente estabelecido). Então, verifica-se o número de soluções existentes na memória que são viáveis, ou seja, que não possuem violações nas restrições. Em seguida, caso este número seja maior do que zero, localiza-se o anticorpo que possui as características perseguidas, apresentadas no Procedimento 2 ilustrado acima, este é selecionado e colocado na população corrente. O procedimento é repetido até que se atinja o tamanho desejado para a população, respeitando-se as situações onde não existem mais soluções viáveis – quando se escolhem aquelas soluções com menor valor de violação das restrições.

Com isso haverá três formas de distinção que deverão ser capazes de diferenciar as soluções, e o operador de seleção que, em conjunto com as características apresentadas deverá ser capaz de fazer o algoritmo evoluir para as melhores soluções.

3.1.10 Suprimir Memória

Como nas etapas anteriores indivíduos foram alocados no conjunto externo de pareto (ou memória externa de pareto), quando se atinge um determinado tamanho pré-estabelecido, é necessário verificar quais indivíduos deverão permanecer neste conjunto, uma vez que este conjunto não pode crescer indefinidamente. Sendo assim, foram

estabelecidos critérios para remoção dos indivíduos que são: somente permanecerão na população indivíduos que pertencem à primeira camada; e indivíduos que foram selecionados para compor a nova população corrente. Caso mesmo assim, o tamanho da população externa ultrapasse um limite pré-definido, os indivíduos com menor valor de *DCD* serão removidos, ou seja, indivíduos em região com maior densidade serão removidos. A cada remoção de indivíduos, o valor de *DCD* de todos os restantes deverá ser recalculado. Este processo é repetido até se atingir o tamanho limite pré-estabelecido para o conjunto de pareto. Um detalhe importante é que não serão removidos os indivíduos da borda do conjunto uma vez que estes possuem os maiores valores de *DCD*.

Desta forma, somente existirão indivíduos não-dominados, e que apresentam o melhor espalhamento, uma vez que aqueles que foram identificados como sendo indivíduos que representam pouca informação (baixo DCD), foram removidos.

3.1.11 Critério de Parada Satisfeito

Verifica-se se o critério de parada escolhido foi atingido, caso verdadeiro, pare o algoritmo, caso falso, voltar ao passo da clonagem e seguir o processo até que se atinja o critério de parada.

O critério de parada poderá ser o número de avaliação da função objetivo, o número de gerações, entre outros.

3.2 TRATAMENTO DAS RESTRIÇÕES

Para o tratamento das restrições optou-se por realizar, um dos mais simples dos tratamentos que é: adiciona-se ao valor da afinidade, o somatório do valor das violações das restrições ao quadrado.

$$afinidade_{ip} = \left[s_{ip} \right] + \left[densidade_{ip} \right] + \left[camada_{ip} \right] + \sum_{j=1}^{N_{re}} \left(mk_j \cdot g_j \right)^2 \qquad (3.6)$$

Onde N_{re} é o número de restrições; g_j representa o valor de cada violação das restrições j; e mk_j é o peso atribuído a cada restrição. Nota-se que a equação (3.6) somente atende a problemas de minimização.

Cabe ressaltar que novas implementações podem ser feitas para aperfeiçoar o tratamento das restrições das funções multi-objetivos, como por exemplo, os artigos de YEN (2009) e BARBOSA & LEMONGE (2003, 2008).

Capítulo 4 VALIDAÇÃO DO ALGORITMO

Equation Chapter (Next) Section 1Neste capítulo serão tratados os testes de validação do algoritmo desenvolvido. Inicialmente são abordados testes em funções matemáticas com apenas dois objetivos e funções sem restrição. Por fim, são apresentados testes em funções matemáticas com restrição e problemas de engenharia também com restrição já testados e apresentados em artigos descritos no estado da arte.

O algoritmo SIAMEP foi testado em diversas funções matemáticas, com grau crescente de dificuldade. Os testes iniciais foram executados com 250 gerações e usando a hipermutação somática. As configurações adotadas para os testes iniciais estão expostas na Tabela 4.1.

Tabela 4.1: Configurações adotadas nos teste de Schaffer's f_1

PARÂMETROS	T1	T2	T3
Número de anticorpos (N_{ab})	4	10	10
Número de clones (N_{clones})	9	3	3
Tamanho População	*40*	*40*	*40*
Par. mutação somática (ρ)	2	2	2
Par. mutação somática (β_m)	6	6	6
Par. mutação somática ($\beta_{m-final}$) (*)	$\beta_m \cdot 10^{-2}$	$\beta_m \cdot 10^{-2}$	$\beta_m \cdot 10^{-2}$
Escopo mutação somática	Todos os parâmetros		
Indivíduos no conjunto pareto	5	30	70

(*) $\beta_{m-final}$ representa o valor de β_m após sofrer um decaimento linear com o num. de avaliações ou gerações.

Os parâmetros do algoritmo foram escolhidos de forma a garantir uma boa exploração global combinado com uma boa capacidade de busca local. Para uma melhor exploração global, aumenta-se o número de anticorpos, proporcionando mais indivíduos para a busca. Para uma maior busca local utiliza-se um número maior de clones. Em resumo, a relação entre estes dois parâmetros irá influenciar na capacidade exploratória do algoritmo. A escolha por um ρ fixo é função da baixa complexidade da função e é usado um ρ igual a dois pela suavidade da curva da mutação proporcionada por este coeficiente. O parâmetro β_m adotado segue a recomendação geral de metade do intervalo de cada variável. E por fim, o número de indivíduos no conjunto de pareto, assim como o tamanho da população e o número de gerações, foram os mesmos apresentados nas referências bibliográficas, objetivando-se permitir a comparação dos resultados.

Uma tabela com resultados completos dos valores de F_1 e F_2 no fronte de pareto obtidos pelo algoritmo SIAMEP para as funções ZDT1, ZDT2, ZDT3, ZDT4 e ZDT6, CONSTR_EX e TNK estão disponíveis no anexo A deste trabalho (Tabela A.1, Tabela A.2 e Tabela A.3).

4.1 FUNÇÃO F_1 DE SCHAFFER

A primeira função avaliada foi apresentada em SCHAFFER (1984). É uma função básica para teste de problemas multi-objetivos, não apresenta dificuldade, servindo apenas como teste de validação da implementação.

$$f_1 = \big(g(x), h(x)\big), \text{ onde } g(x) = x^2 \text{ e } h(x) = (x-2)^2 \tag{4.1}$$

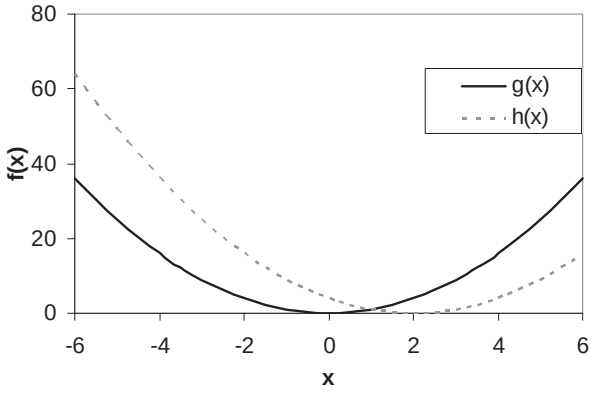
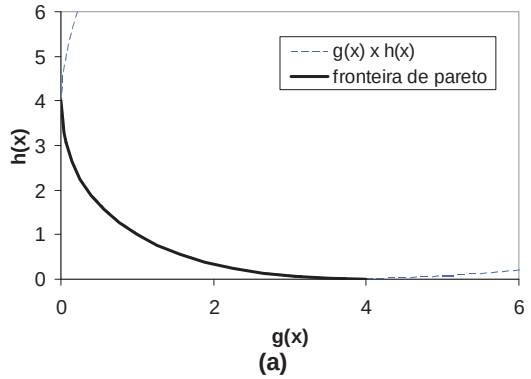
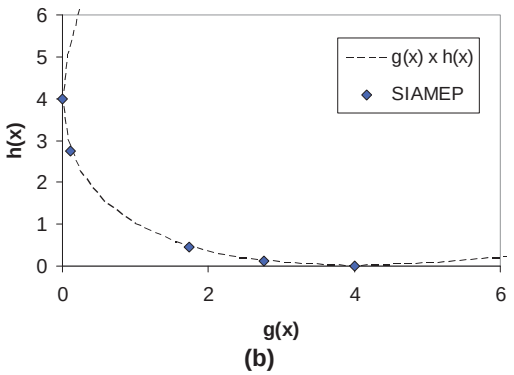


Figura 4.1: gráfico de g(x) e h(x) com $-6 \le x \le 6$.

A seguir são apresentados os resultados do algoritmo proposto comparados com os obtidos em ZITZLER *et al.* (1998).



(a)



(b)

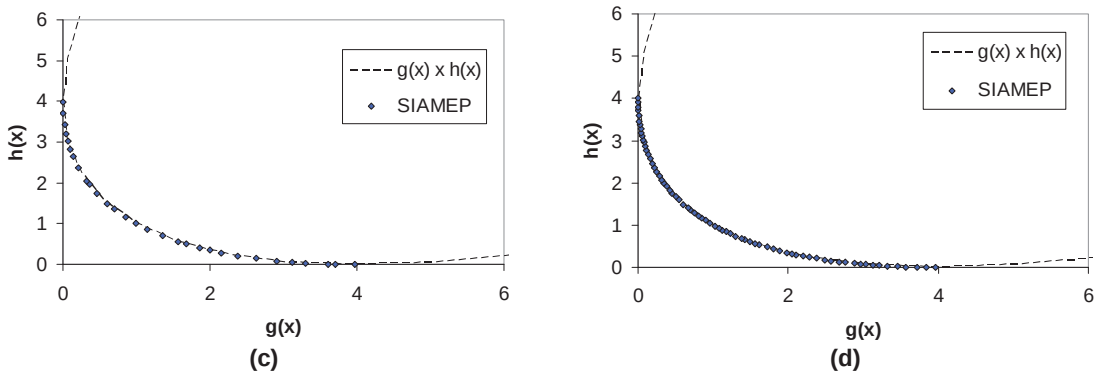


Figura 4.2: (a) Fronte de pareto verdadeiro**;** Fronte obtido com **(b**) T1(conj de pareto = 5) **(c)** T2 (conj de pareto = 30)**; (d)** T3 (conj de pareto= 70)

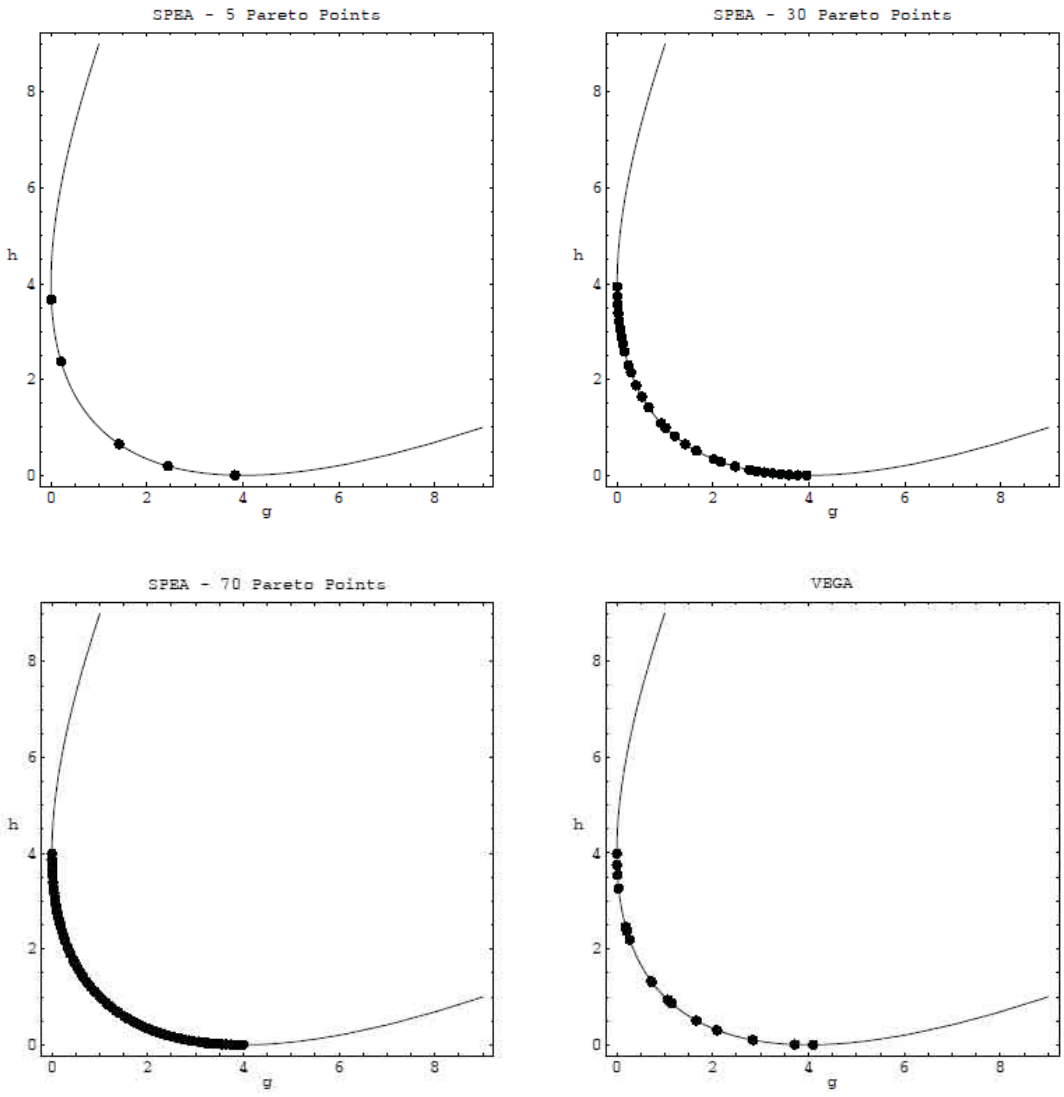


Figura 4.3: Resultados para Schaffer F_1 em ZITZLER *et al.* (1998) para conjunto de pareto com 5, 30 e 70 soluções e resultado para algoritmo VEGA.

Como se pode notar, o algoritmo proposto foi visualmente compatível com os resultados encontrados na literatura.

4.2 FUNÇÃO F_2 DE SCHAFFER

A segunda função avaliada foi apresentada em SCHAFFER (1984). É uma função básica para teste de problemas multi-objetivos, já apresenta um grau de dificuldade um pouco maior por ter um fronte interrompido.

$$f_1 = \begin{cases} -x & se\ x \le 1 \\ -2 + x & se\ 1 < x \le 3 \\ 4 - x & se\ 3 < x \le 4 \\ -4 + x & se\ x > 4 \end{cases}, \text{ onde } x \in [-5,10] \qquad (4.2)$$

$$f_2 = (x-5)^2$$

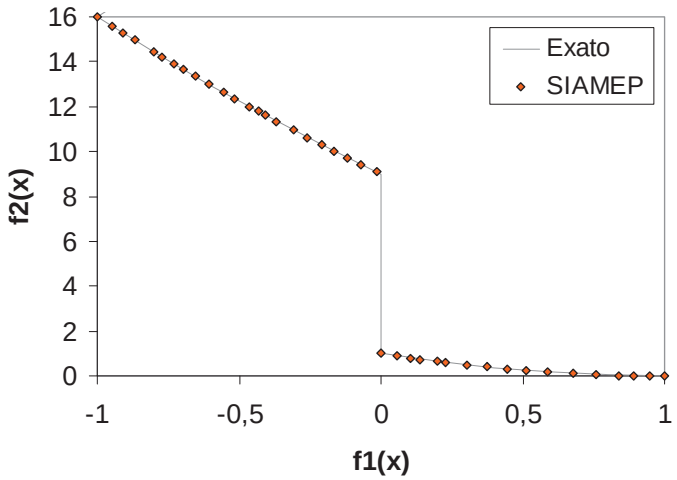


Figura 4.4: Resultados para Schaffer f_2

Como se pode notar, o algoritmo proposto obteve um fronte bem coincidente com o exato, com isso o estudo de caso segue para funções matemáticas do conjunto proposto por Zitzler-Deb-Thiler.

4.3 PROBLEMAS ZDT

As funções do problema ZDT são para minimização e não possuem restrições (funções apresentadas na Tabela 4.3). Para os testes com estas funções matemáticas foram utilizadas as configurações apresentadas na Tabela 4.2 com o critério de parada sendo o número máximo de avaliações e mutação híbrida. Para comparação com algoritmos do estado da arte, foram adotadas 24.000 avaliações da função objetivo, muito embora, o algoritmo apresente resultados satisfatórios com um número de avaliações inferior para algumas das funções apresentadas. Todos os testes do algoritmo proposto foram repetidos por 30 vezes para a obtenção dos resultados estatísticos e, para o cálculo das métricas de convergência, foi utilizado um fronte exato com 500 soluções

uniformemente espaçadas, valor este sendo o mesmo adotado nas referências citadas. Um dos objetivos deste teste foi comparar os resultados do algoritmo desenvolvido com resultados encontrados na bibliografia no intuito de verificar a capacidade do algoritmo para resolução de problemas multi-objetivos com diferentes graus de dificuldade. Para os testes do problema ZDT o conjunto de pareto possui 40 indivíduos.

Tabela 4.2: Configurações do algoritmo para os testes ZDT

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
Num. de Anticorpos (N_{ab}) e Clones (N_{clones})	[15 e 3]	[15 e 3]	[15 e 3]	[16 e 3]	[15 e 3]
Tamanho População	*60*	*60*	*60*	*64*	*60*
Parâmetro mutação somática (β_m)	$m = 1,...,30$ $\beta_m = 1$	$m = 1,...,30$ $\beta_m = 1$	$m = 1,...,30$ $\beta_m = 1$	$\beta_1 = 1$ $m = 2,...,10$ $\beta_m = 3$	$m = 1,...,10$ $\beta_m = 1$
Parâmetro mutação somática ($\beta_{m-final}$) (*)	$\beta_m \cdot 10^{-3}$	$\beta_m \cdot 10^{-3}$	$\beta_m \cdot 10^{-3}$	$\beta_m \cdot 10^{-5}$	$\beta_m \cdot 10^{-4}$
Parâmetro mutação somática (ρ)	2	2	2	2	2
Escopo mutação somática	1 param.	1 param.	2 param.	1 param.	1 param.
Parâmetro mutação diferencial (*F*)	0,8				
Escopo mutação diferencial	Todos os parâmetros podem sofrer mutação				

(*) $\beta_{m-final}$ representa o valor de β_m após sofrer um decaimento linear com o num. de avaliações, ou gerações.

A tabela a seguir apresenta a formulação matemática das funções utilizadas.

Tabela 4.3: Funções ZDT para testes (todas para minimização)

NOME	LIMITE	FUNÇÕES OBJETIVO	SOLUÇÃO ÓTIMA	COMENTÁRIOS
ZDT1 *N_{var}=30*	[0,1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{x_1 / g(x)}\right]$ $g(x) = 1 + 9\left[\left(\sum_{m=2}^{N_{var}} x_m\right) / (N_{\text{var}} - 1)\right]$	$x_1 \in [0,1]$ $x_m = 0,$ $m = 2,..., N_{\text{var}}$	convexa
ZDT2 *N_{var}=30*	[0,1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \left(x_1 / g(x)\right)^2\right]$ $g(x) = 1 + 9\left[\left(\sum_{m=2}^{N_{var}} x_m\right) / (N_{\text{var}} - 1)\right]$	$x_1 \in [0,1]$ $x_m = 0,$ $m = 2,..., N_{\text{var}}$	não-convexa
ZDT3 *N_{var}=30*	[0,1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{x_1 / g(x)} - \frac{x_1}{g(x)}\sin(10\pi x_1)\right]$ $g(x) = 1 + 9\left[\left(\sum_{m=2}^{N_{var}} x_m\right) / (N_{\text{var}} - 1)\right]$	$x_1 \in [0,1]$ $x_m = 0,$ $m = 2,..., N_{\text{var}}$	convexa desconectada

NOME	LIMITE	FUNÇÕES OBJETIVO	SOLUÇÃO ÓTIMA	COMENTÁRIOS
ZDT4 $N_{var}=10$	$x_1 \in [0,1]$, $x_m \in [-5,5]$, $m= 2,...,N_{var}$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1-\sqrt{x_1/g(x)}\right]$ $g(x) = 1+10(N_{\text{var}}-1)+\sum_{m=2}^{N_{\text{var}}}\left(x_m^2-10\cos(4\pi x_m)\right)$	$x_1 \in [0,1]$ $x_m = 0,$ $m = 2,...,N_{\text{var}}$	convexa
ZDT6 $N_{var}=10$	[0,1]	$f_1(x) = 1-\exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)\left[1-\left(f_1(x)/g(x)\right)^2\right]$ $g(x) = 1+9\left[\left(\sum_{m=2}^{N_{\text{var}}} x_m\right)/(N_{\text{var}}-1)\right]^{0,25}$	$x_1 \in [0,1]$ $x_m = 0,$ $m = 2,...,N_{\text{var}}$	não-convexa, não-uniformemente espaçada

Onde N_{var} representa o número de variáveis.

As tabelas comparativas que se seguem, foram construídas com sete colunas: a primeira coluna mostra o nome dos algoritmos com o número de avaliações da função objetivo expresso entre parênteses logo abaixo, ou ao lado, do nome do algoritmo; na segunda e terceira colunas estão expressos respectivamente os resultados da média e desvio padrão para a medida de desempenho GD; na quarta e quinta colunas estão expressos respectivamente os resultados da média e desvio padrão para a medida de desempenho SP; na sexta e sétima colunas estão expressos respectivamente os resultados da média e desvio padrão para a medida de desempenho Δ. Nas linhas das tabelas temos os algoritmos sendo utilizado como referência os seguintes trabalhos: PAES-M (BARDANACHVILI, 2006); NSGA-II real, NSGA-II binário, e PAES (DEB *et al*., 2002); HIMO e SPEA-II (CHEN *et al*., 2010); ACSAMO (WANG & MAHFOUF, 2006); MOEO (CHEN & LU, 2008); NSPSO (GUZMÁN *et al.*, 2010); CSS-MOPSO e MOPSO (KAVEN & LAKNEJADI, 2011); MOOCEM (BEKKER & ALDRICH, 2011).

4.3.1 ZDT1

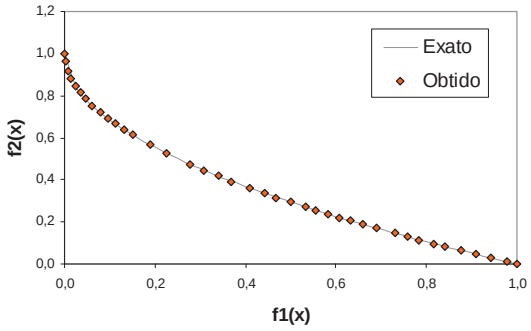
É possível reescrever a equação dada na Tabela 4.3 para uma forma mais simplificada dada pela expressão a seguir (4.3).

$$ZDT1:\begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)\cdot h(f_1(x), g(x)) \\ g(x) = 1+\left[9/(N_{\text{var}}-1)\right]\cdot\sum_{m=2}^{N_{\text{var}}} x_m;\ N_{\text{var}} = 30 \\ h(f_1, g) = 1-\sqrt{f_1/g} \\ 0 \le x_m \le 1 \end{cases} \tag{4.3}$$

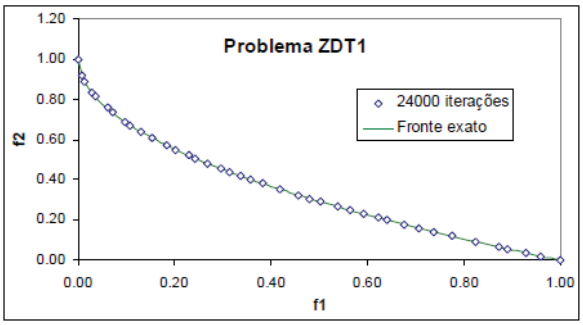
Neste caso, o fronte é convexo e a expressão para a solução exata do fronte é obtida igualando-se x_2 a x_{30}, a zero, o que faz com que $g(x)$ seja igual a 1, atingindo seu menor valor, e é dada por:

$$f_1(x) = x_1, (0 \leq x_1 \leq 1)$$
$$f_2(x) = 1 - \sqrt{f_1(x)} \tag{4.4}$$

Na Figura 4.5 são mostrados os resultados dos algoritmos SIAMEP e PAES-M para uma execução com 24.000 avaliações.



(a)



(b)

Figura 4.5: Resultados para ZDT1 **(a)** com SIAMEP; **(b)** com PAES-M

Como se pode perceber pela Tabela 4.4, com o mesmo número de avaliações de PAES-M, o algoritmo proposto foi capaz de encontrar um fronte mais próximo do fronte verdadeiro, indicado por um valor mais baixo de GD. No entanto, PAES-M obteve um fronte ligeiramente mais uniforme, conforme se pode perceber pelo valor de Δ. Em relação a SP nota-se que o SIAMEP apresentou-se superior aos demais algoritmos com um menor valor para SP.

Tabela 4.4: Resultados comparativos ZDT1

ALGORITMOS	GD		SP		Δ	
	média	***STD***	**média**	***STD***	**média**	***STD***
SIAMEP (24.000)	**0,000041**	*0,000016*	**0,001415**	*0,000787*	0,160824	*0,018660*
PAES (25.000)	0,082085	*0,093161*	-	-	1,229794	*0,069563*
PAES-M (24.000)	0,000143	*0,000200*	-	-	**0,137407**	*0,020873*
NSGA-II binário (25.000)	0,000894	*0,000000*	-	-	0,463292	*0,204015*
NSGA-II real (25.000)	0,033482	*0,068920*	-	-	0,390307	*0,043313*
HIMO (25.000)	0,000733	*0,006550*	-	-	0,346411	*0,176482*
SPEA-II (25.000)	0,001448	*0,017804*	-	-	0,472254	*0,311564*
ACSAMO (50.000)	0,000232	*0,005797*	-	-	0,306300	*0,143527*
MOEO (25.000)	0,001277	*0,026401*	-	-	0,327140	*0,255623*

ALGORITMOS	GD		SP		Δ	
	média	*STD*	média	*STD*	média	*STD*
NSPSO (25.000)	0,000437	*0,000184*	-	-	0,638000	*0,051865*
MOPSO (10.000)	0,096400	*0,011428*	0,756200	0,145703	-	-
CSS-MOPSO (10.000)	0,003048	*0,000288*	0,199631	0,048466	-	-
MOOCEM (10.000)	0,001200	-	0,003900	-	-	-

4.3.2 ZDT2

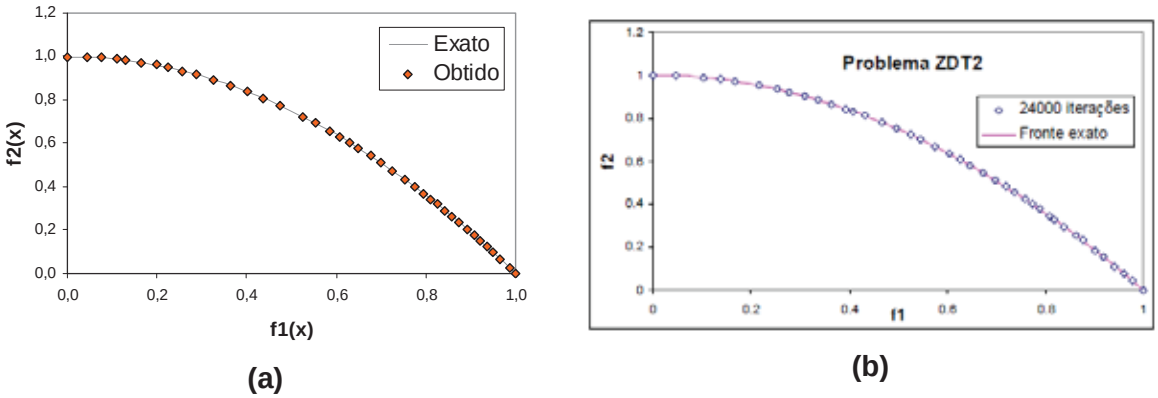
Pode-se reescrever a equação dada na Tabela 4.3 para uma forma mais simplificada dada pela expressão a seguir (4.5).

$$ZDT2: \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x) \cdot h(f_1(x), g(x)) \\ g(x) = 1 + \left[9/(N_{\text{var}} - 1)\right] \cdot \sum_{m=2}^{N_{\text{var}}} x_m; \;\; N_{\text{var}} = 30 \\ h(f_1, g) = 1 - (f_1/g)^2 \\ 0 \le x_m \le 1 \end{cases} \tag{4.5}$$

O fronte é não-convexo e a expressão para a solução exata do fronte é obtida igualando-se x_2 a x_{30}, a zero, o que faz com que $g(x)$ seja igual a 1, atingindo seu menor valor, e é dada por:

$$\begin{aligned} f_1(x) &= x_1, (0 \le x_1 \le 1) \\ f_2(x) &= 1 - \left[f_1(x)\right]^2 \end{aligned} \tag{4.6}$$

Na Figura 4.6 são mostrados os resultados dos algoritmos SIAMEP e PAES-M para uma execução com 24.000 avaliações



(a) **(b)**

Figura 4.6: Resultados para ZDT2 **(a)** com SIAMEP; **(b)** com PAES-M

Como se pode perceber pela Tabela 4.5, para ZDT2 o algoritmo proposto ficou ligeiramente atrás do algoritmo PAES-M, tanto nos valores de GD como também de Δ. No entanto, cabe ressaltar que a diferença no valor de GD é bastante pequena, pois a mesma ocorre na quinta casa decimal, não sendo possível afirmar que o algoritmo SIAMEP é inferior ao PAES-M. Em relação a SP, SIAMEP foi melhor do que MOOCEM, destacando-se que este apresenta um menor número de avaliações, comprometendo a comparação.

Tabela 4.5: Resultados comparativos ZDT2

ALGORITMOS	GD		SP		Δ	
	média	***STD***	**média**	***STD***	**média**	***STD***
SIAMEP (24.000)	0,000023	*0,000004*	**0,000524**	*0,000034*	0,165728	*0,017843*
PAES (25.000)	0,126276	*0,192034*	-	-	1,165942	*0,087647*
PAES-M (24.000)	**0,000012**	*0,000022*	-	-	**0,137666**	*0,024722*
NSGA-II binário (25.000)	0,000824	*0,000000*	-	-	0,435112	*0,156867*
NSGA-II real (25.000)	0,072391	*0,178014*	-	-	0,430776	*0,068710*
HIMO (25.000)	0,000707	*0,006450*	-	-	0,348676	*0,172029*
SPEA-II (25.000)	0,000743	*0,009127*	-	-	0,473808	*0,306431*
ACSAMO (50.000)	0,000097	*0,003050*	-	-	0,326200	*0,109545*
MOEO (25.000)	0,001355	*0,029950*	-	-	0,285062	*0,238701*
NSPSO (25.000)	0,000350	*0,000128*	-	-	0,580000	*0,032863*
MOPSO (10.000)	-	-	-	-	-	-
CSS-MOPSO (10.000)	-	-	-	-	-	-
MOOCEM (10.000)	0,001200	-	0,002700	-	-	-

4.3.3 ZDT3

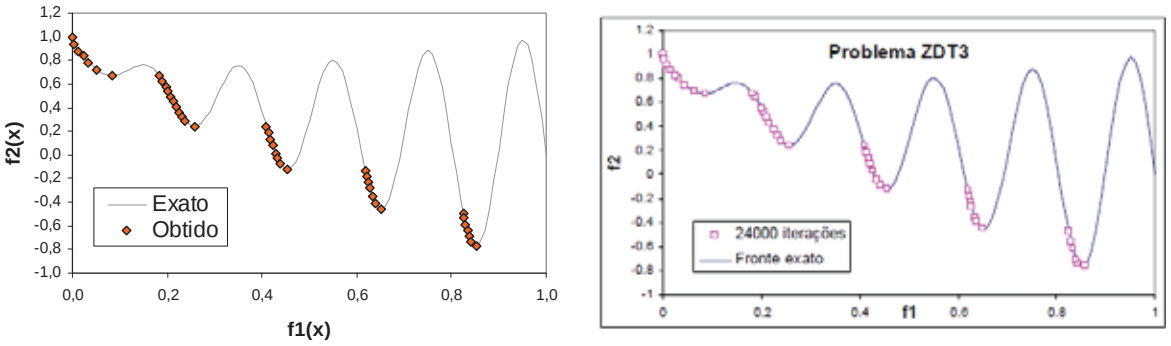
Pode-se reescrever a equação dada na Tabela 4.3 para uma forma mais simplificada dada pela expressão a seguir (4.7).

$$ZDT3: \begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x) \cdot h(f_1(x), g(x)) \\ g(x) = 1 + \left[9/(N_{\text{var}} - 1)\right] \cdot \sum_{m=2}^{N_{\text{var}}} x_m; \;\; N_{\text{var}} = 30 \\ h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g) \cdot \sin(10\pi f_1) \\ 0 \le x_m \le 1 \end{cases} \tag{4.7}$$

O fronte é descontínuo (contendo cinco trechos) e convexo, e a expressão para a solução exata do fronte é obtida igualando-se x_2 a x_{30}, a zero, o que faz com que $g(x)$ seja igual a 1, atingindo seu menor valor, e é dada por:

$$f_1(x) = x_1, (0 \le x_1 \le 1)$$
$$f_2(x) = 1 - \sqrt{f_1(x)} - f_1(x) \cdot \sin\left(10\pi f_1(x)\right) \tag{4.8}$$

Na Figura 4.7 são mostrados os resultados dos algoritmos SIAMEP e PAES-M para uma execução com 24.000 avaliações



(a) **(b)**

Figura 4.7: Resultados para ZDT3 **(a)** com SIAMEP; **(b)** com PAES-M

Como se pode notar o algoritmo SIAMEP conseguiu alcançar um fronte razoavelmente bem distribuído e com valor de GD ligeiramente pior do que em PAES-M (ou seja, ligeiramente maior) ficando em segundo lugar na comparação. E também, o SIAMEP perde na homogeneidade do espalhamento por apresentar um valor de Δ maior do que o de NSPSO e PAES-M. Neste caso a maior dificuldade foi em apresentar os pontos da função no trecho inferior a direita. Em relação à media SP, SIAMEP apresenta um valor superior ao de MOOCEM, no entanto MOOCEM não apresenta desvio padrão e possui um número inferior de avaliações. Neste caso a comparação fica comprometida.

Tabela 4.6: Resultados comparativos ZDT3

ALGORITMOS	GD		SP		Δ	
	média	***STD***	**média**	***STD***	**média**	***STD***
SIAMEP (24.000)	0,000727	*0,000202*	0,003851	*0,000548*	0,376682	*0,019474*
PAES (25.000)	0,023872	*0,003162*			0,789920	*0,040657*
PAES-M (24.000)	**0,000572**	*0,000753*			0,157638	*0,025408*
NSGA-II binário (25.000)	0,114500	*0,089107*			0,738540	*0,140378*
NSGA-II real (25.000)	0,043411	*0,006481*			0,575606	*0,071260*
HIMO (25.000)	0,001300	*0,009094*			0,529629	*0,145598*
SPEA-II (25.000)	0,003716	*0,024207*			0,606826	*0,437500*
ACSAMO (50.000)	0,000628	*0,005148*			0,626400	*0,115326*
MOEO (25.000)	0,004385	*0,043704*			0,965236	*0,216698*
NSPSO (25.000)	0,000616	*0,000172*	-	-	**0,007770**	*0,008485*
MOPSO (10.000)	0,068005	*0,013964*	0,794325	*0,070546*	-	-

ALGORITMOS	GD		SP		Δ	
	média	***STD***	**média**	***STD***	**média**	***STD***
CSS-MOPSO (10.000)	0,004781	*0,000141*	0,301347	*0,042321*	-	-
MOOCEM (10.000)	0,003900	-	**0,002400**	-	-	-

4.3.4 ZDT4

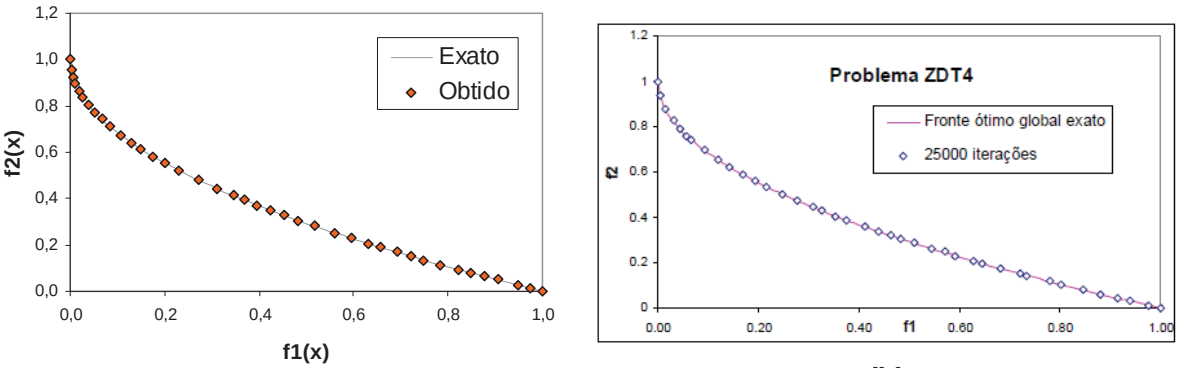
Pode-se reescrever a equação dada na Tabela 4.3 para uma forma mais simplificada dada pela expressão a seguir (4.9).

$$ZDT4:\begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)\cdot h(f_1(x), g(x)) \\ g(x) = 1+10\left(N_{\text{var}}-1\right)+\sum_{m=2}^{N_{\text{var}}}\left(x_m^2-10\cos\left(4\pi x_m\right)\right);\ N_{\text{var}}=10 \\ h(f_1, g) = 1-\sqrt{f_1/g} \\ 0 \le x_1 \le 1,\ \text{e}\ -5 \le x_m \le 5,\ m = 2,...,10 \end{cases} \tag{4.9}$$

O fronte é convexo, com a peculiaridade de possuir 100 frontes ótimos locais, devendo o algoritmo encontrar o fronte ótimo global. Com as variáveis x_2 a x_{10} no intervalo [-5,5] e x_1 no intervalo [0,1], há 21^9 soluções locais Pareto-ótimas, obtidas com passo 0,5 para as 9 variáveis x_2 a x_{10}, independentes. Essas soluções formam 100 frontes Pareto-ótimos. A expressão para a solução exata do fronte é obtida igualando-se x_2 a x_{30}, a zero, o que faz com que $g(x)$ seja igual a 1, atingindo seu menor valor, e é dada por:

$$\begin{aligned} f_1(x) &= x_1, (0 \le x_1 \le 1) \\ f_2(x) &= 1-\sqrt{f_1(x)} \end{aligned} \tag{4.10}$$

Na Figura 4.8 são mostrados os resultados dos algoritmos SIAMEP e PAES-M para uma execução com 25.000 avaliações.



(a) **(b)**

Figura 4.8: Resultados para ZDT4 **(a)** com SIAMEP; **(b)** com PAES-M

Neste caso o algoritmo SIAMEP mostrou-se superior ao PAES-M, obtendo um fronte com menor valor de GD e com um bom resultado de Δ (segundo melhor). Destaca-se que ACSAMO apresenta um valor de GD menor do que SIAMEP, no entanto ele consegue isto com um número maior de avaliações, o que não é desejável. Em SP o SIAMEP apresenta o melhor resultado comparado à demais soluções destacando-se que os demais algoritmos apresentam número inferior de avaliações. Nota-se que a maior dificuldade nesta função é a importância das variações dos parâmetros em diferentes ordens de grandeza. Isto demonstra a importância do decaimento linear dos valores do parâmetro da mutação somática β_m, que permite o algoritmo explorar tanto os grandes valores de mutação como os pequenos, mais ao final da evolução.

Tabela 4.7: Resultados comparativos ZDT4

ALGORITMOS	GD		SP		Δ	
	média	***STD***	**média**	***STD***	**média**	***STD***
SIAMEP (24.000)	**0,000407**	*0,000250*	**0,002072**	*0,000746*	0,197123	*0,024809*
PAES (25.000)	0,854816	*0,726112*	-	-	0,870458	*0,318432*
PAES-M (24.000)	0,003185	*0,009310*	-	-	**0,139004**	*0,023742*
NSGA-II binário (25.000)	0,513053	*0,344180*	-	-	0,702612	*0,254260*
NSGA-II real (25.000)	3,227636	*2,703263*	-	-	0,479475	*0,099202*
HIMO (25.000)	0,003248	*0,044385*	-	-	0,312165	*0,178818*
SPEA-II (25.000)	0,028492	*0,217904*	-	-	0,705629	*0,515909*
ACSAMO (50.000)	**0,000300**	*0,006892*	-	-	0,295800	*0,160312*
MOEO (25.000)	0,008145	*0,004011*	-	-	0,275664	*0,428607*
NSPSO (25.000)	0,405000	*0,207846*	-	-	0,975000	*0,017263*
MOPSO (10.000)	0,228776	*0,070243*	1,081437	*0,195925*	-	-
CSS-MOPSO (10.000)	0,003462	*0,000380*	0,175199	*0,031783*	-	-
MOOCEM (10.000)	-	-	-	-	-	-

4.3.5 ZDT6

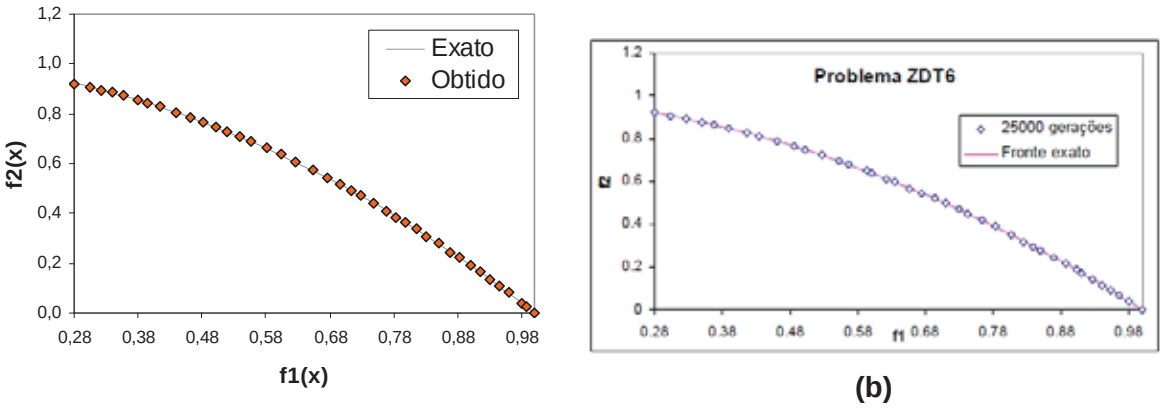
Pode-se reescrever a equação dada na Tabela 4.3 para uma forma mais simplificada dada pela expressão a seguir (4.11).

$$ZDT6: \begin{cases} f_1(x) = 1 - \exp(-4x_1) \cdot \sin^6(6\pi x_1) \\ f_2(x) = g(x) \cdot h(f_1(x), g(x)) \\ g(x) = 1 + (N_{\text{var}} - 1) \cdot \left[\left(\sum_{m=2}^{N_{\text{var}}} x_m^2 \right) \Big/ (N_{\text{var}} - 1) \right]^{0,25} \;;\; N_{\text{var}} = 10 \\ h(f_1, g) = 1 - \left(f_1 / g \right)^2 \\ 0 \le x_m \le 1 \end{cases} \tag{4.11}$$

O fronte é mais uma vez não-convexo e a expressão para a solução exata do fronte é obtida igualando-se x_2 a x_{30}, a zero, o que faz com que $g(x)$ seja igual a 1, atingindo seu menor valor, e é dada por:

$$\begin{aligned} f_1(x) &= x_1, (0 \le x_1 \le 1) \\ f_2(x) &= 1 - \left[f_1(x) \right]^2 \end{aligned} \tag{4.12}$$

Na Figura 4.9 são mostrados os resultados dos algoritmos SIAMEP e PAES-M para uma execução com 24.000 avaliações.



(a) **(b)**

Figura 4.9: Resultados para ZDT6 **(a)** com SIAMEP; **(b)** com PAES-M

Neste caso o algoritmo SIAMEP conseguiu alcançar um fronte visivelmente bem distribuído e apresentando um valor de GD menor do que em HIMO que foi o segundo melhor neste caso. O SIAMEP também ganhou no quesito homogeneidade do espalhamento, obtendo um valor inferior ao de PAES-M no valor de Δ, e também possui o menor valor de SP comparado aos demais algoritmos.

Tabela 4.8: Resultados comparativos ZDT6

ALGORITMOS	GD		SP		Δ	
	média	***STD***	**média**	***STD***	**média**	***STD***
SIAMEP (24.000)	**0,000019**	*0,000004*	**0,000557**	*0,000021*	**0,169381**	*0,017677*
PAES (25.000)	0,085469	*0,081633*	-	-	1,153052	*0,062578*
PAES-M (24.000)	0,022886	*0,048919*	-	-	0,319951	*0,326727*
NSGA-II binário (25.000)	7,806798	*0,040829*	-	-	0,644477	*0,187195*
NSGA-II real (25.000)	0,296564	*0,114608*	-	-	0,668025	*0,099614*
HIMO (25.000)	0,000740	*0,007287*	-	-	0,444738	*0,143882*
SPEA-II (25.000)	0,011643	*0,048959*	-	-	0,670549	*0,277505*
ACSAMO (50.000)	-	-	-	-	-	-
MOEO (25.000)	0,000630	*0,000033*	-	-	0,225468	*0,033884*
NSPSO (25.000)	-	-	-	-	-	-
MOPSO (10.000)	0,019078	*0,009291*	3,707637	*0,849501*	-	-
CSS-MOPSO (10.000)	0,026345	*0,014011*	3,179233	*1,351519*	-	-
MOOCEM (10.000)	-	-	-	-	-	-

4.3.6 Conclusões sobre os resultados das Funções sem restrição

Os resultados são apresentados resumidamente a seguir em duas tabelas. A primeira tabela contendo o resultado quanto à qualidade da convergência, medida de desempenho GD (Tabela 4.9); e uma segunda tabela que apresenta a medida de espalhamento das soluções Δ (Tabela 4.10). As tabelas são ilustradas pelos gráficos subseqüentes para permitir melhor visualização dos resultados.

Pela análise das tabelas a seguir pode-se perceber que em termos de convergência (GD), o algoritmo desenvolvido neste trabalho (SIAMEP) foi capaz de superar em 3 dos 5 testes os resultados do melhor algoritmo da comparação que era o PAES-M. Destacando-se que em ZDT2 e ZDT3 o algoritmo perde por muito pouco.

Tabela 4.9: Resultados de GD: média (1ª linha) e desvio padrão (2ª linha)

	ZDT1	**ZDT2**	**ZDT3**	**ZDT4**	**ZDT6**
SIAMEP	**0,000041**	0,000023	0,000727	**0,000407**	**0,000019**
STD	*0,000016*	*0,000004*	*0,000202*	*0,00025*	*0,000004*
PAES	0,082085	0,126276	0,023872	0,854816	0,085469
STD	*0,093161*	*0,192034*	*0,003162*	*0,726112*	*0,081633*
PAES-M	0,000143	**0,000012**	**0,000572**	0,003185	0,022886
STD	*0,000200*	*0,000022*	*0,000753*	*0,009310*	*0,048919*
NSGA-II (Bin)	0,000894	0,000824	0,043411	3,227636	7,806798
STD	*0,000000*	*0,000000*	*0,006481*	*2,703263*	*0,040829*
NSGA-II (real)	0,033482	0,072391	0,043411	3,227636	0,296564
STD	0,068920	0,178014	0,006481	2,703263	0,114608

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
HIMO	0,000733	0,000707	0,001300	0,003248	0,000740
STD	*0,006550*	*0,006450*	*0,009094*	*0,044385*	*0,007287*
SPEA-II	0,001448	0,000743	0,003716	0,028492	0,011643
STD	*0,017804*	*0,009127*	*0,0242074*	*0,217904*	*0,048959*
ACSAMO	0,000232	0,000097	0,000628	0,000300	-
STD	*0,005797*	*0,003050*	*0,005148*	*0,006892*	-
MOEO	0,001277	0,001355	0,004385	0,008145	0,000630
STD	*0,026401*	*0,029950*	*0,043704*	*0,004011*	*0,000033*
NSPSO	0,000437	0,000350	0,000616	0,405000	-
STD	*0,000184*	*0,000128*	*0,000172*	*0,207846*	-
MOPSO	0,096400	-	0,068005	0,228776	0,019078
STD	*0,011428*	-	*0,013964*	*0,070243*	*0,009291*
CSS-MOPSO	0,003048	-	0,004781	0,003462	0,026345
STD	*0,000288*	-	*0,000141*	*0,000380*	*0,014011*
MOOCEM	0,001200	0,001200	0,003900	-	-
STD	-	-	-	-	-

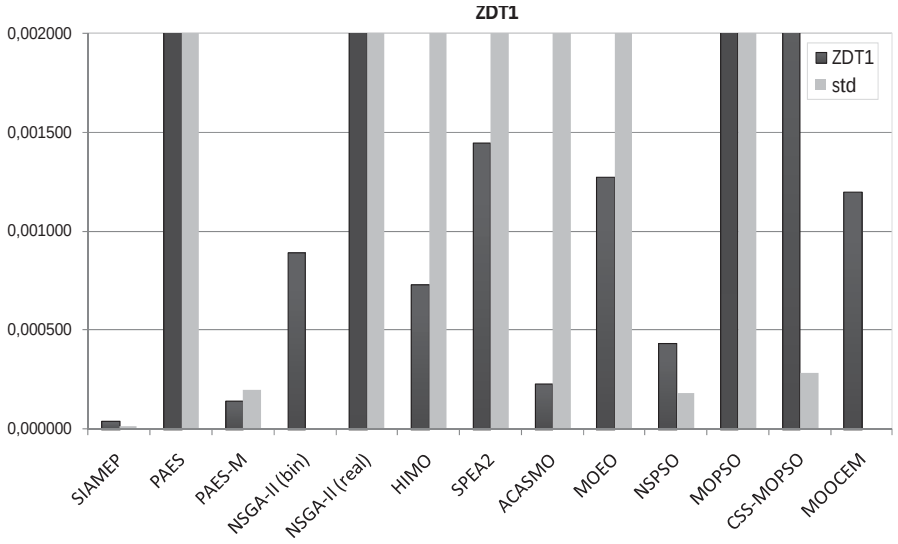


Figura 4.10: Resultado da medida de convergência GD para ZDT1

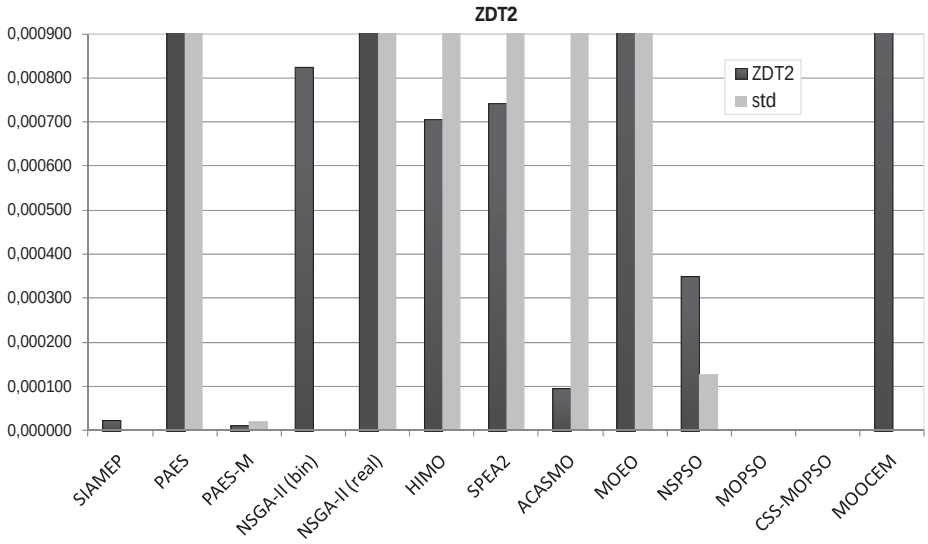


Figura 4.11: Resultado da medida de convergência GD para ZDT2

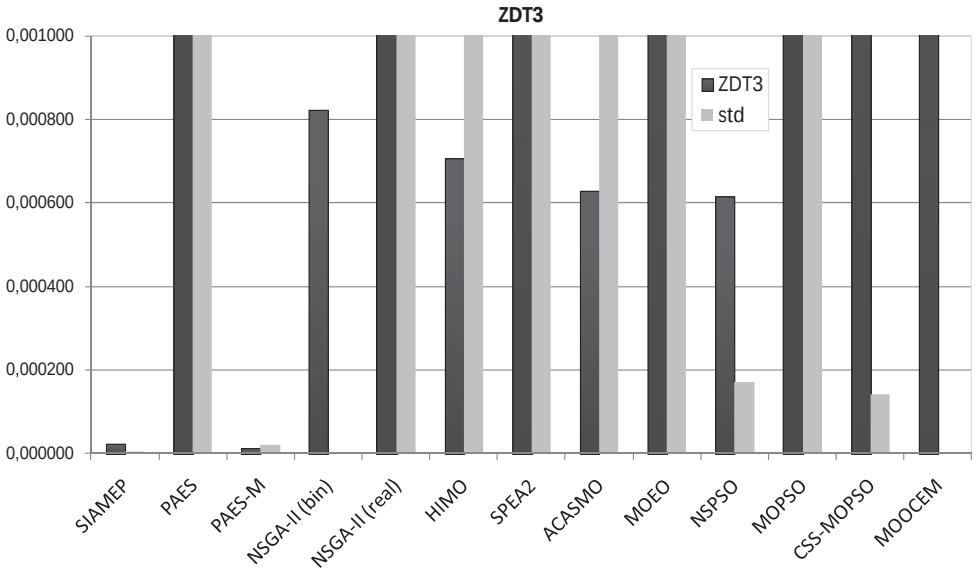


Figura 4.12: Resultado da medida de convergência GD para ZDT3

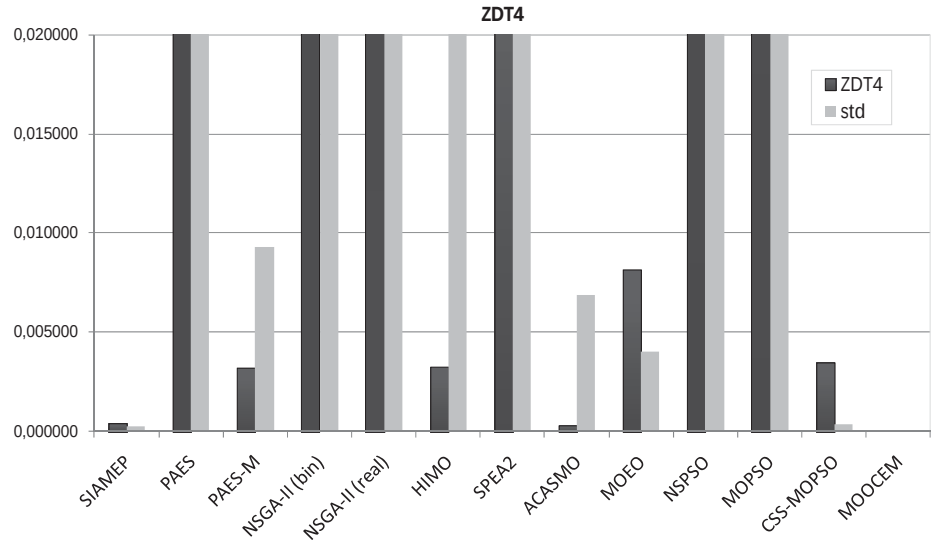


Figura 4.13: Resultado da medida de convergência GD para ZDT4

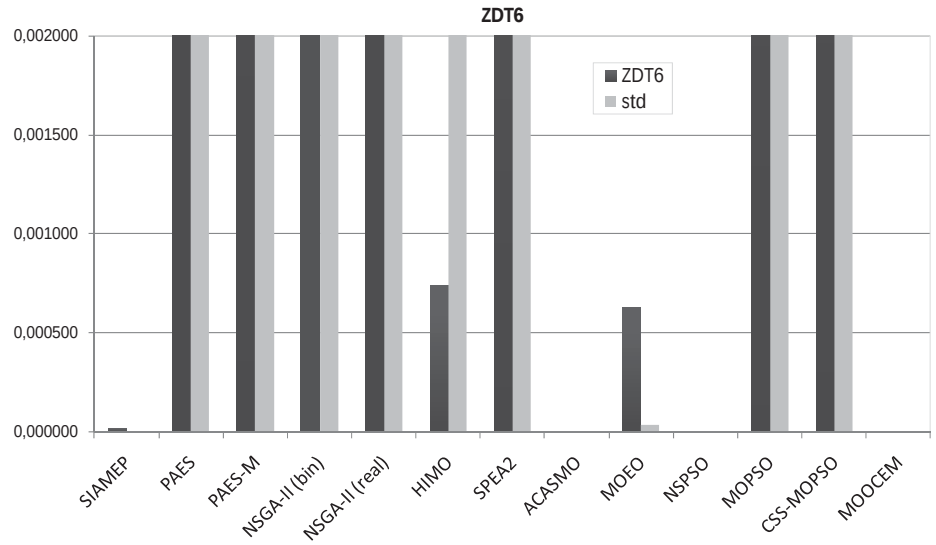


Figura 4.14: Resultado da medida de convergência GD para ZDT6

Tabela 4.10: Resultados de Δ: Média (1ª linha) e desvio padrão (2ª linha)

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
SIAMEP	0,160824	0,165728	0,376682	0,197123	**0,169381**
STD	*0,01866*	*0,017843*	*0,019474*	*0,024809*	*0,017677*
PAES	1,229794	1,165942	0,789920	0,870458	1,153052
STD	*0,069563*	*0,087647*	*0,040657*	*0,318432*	*0,062578*
PAES-M	**0,137407**	**0,137666**	0,157638	**0,139004**	0,319951
STD	*0,020873*	*0,024722*	*0,025408*	*0,023742*	*0,326727*
NSGA-II (Bin)	0,463292	0,435112	0,575606	0,479475	0,644477
STD	*0,204015*	*0,156867*	*0,071260*	*0,099202*	*0,187195*
NSGA-II (real)	0,390307	0,430776	0,575606	0,479475	0,668025
STD	*0,043313*	*0,068710*	*0,071260*	*0,099202*	*0,099614*
HIMO	0,346411	0,348676	0,529629	0,312165	0,444738
STD	*0,176482*	*0,172029*	*0,145598*	*0,178818*	*0,143882*
SPEA-II	0,472254	0,473808	0,606826	0,705629	0,670549
STD	*0,311564*	*0,306431*	*0,437500*	*0,515909*	*0,277505*
ACSAMO	0,306300	0,326200	0,626400	0,295800	-
STD	*0,143527*	*0,109545*	*0,115326*	*0,160312*	*-*
MOEO	0,327140	0,285062	0,965236	0,275664	0,225468
STD	*0,255623*	*0,238701*	*0,216698*	*0,428607*	*0,033884*
NSPSO	0,638000	0,580000	**0,007770**	0,975000	-
STD	*0,051865*	*0,032863*	*0,008485*	*0,017263*	*-*
MOPSO	-	-	-	-	-
STD	-	-	-	-	-
CSS-MOPSO	-	-	-	-	-
STD	-	-	-		-
MOOCEM	-	-	-	-	-
STD	-	-	-	-	-

Em termos da qualidade da distribuição do fronte, ou seja, o quanto o fronte apresentado está bem espaçado, o algoritmo proposto não apresenta os melhores resultados, nem os piores mantendo a mesma ordem de grandeza do resultados obtidos pelos demais algoritmos do estado da arte exceto o PAES-M, que apresenta os melhores resultados. Ainda assim, visualmente, os resultados do SIAMEP mostram um fronte bem distribuído.

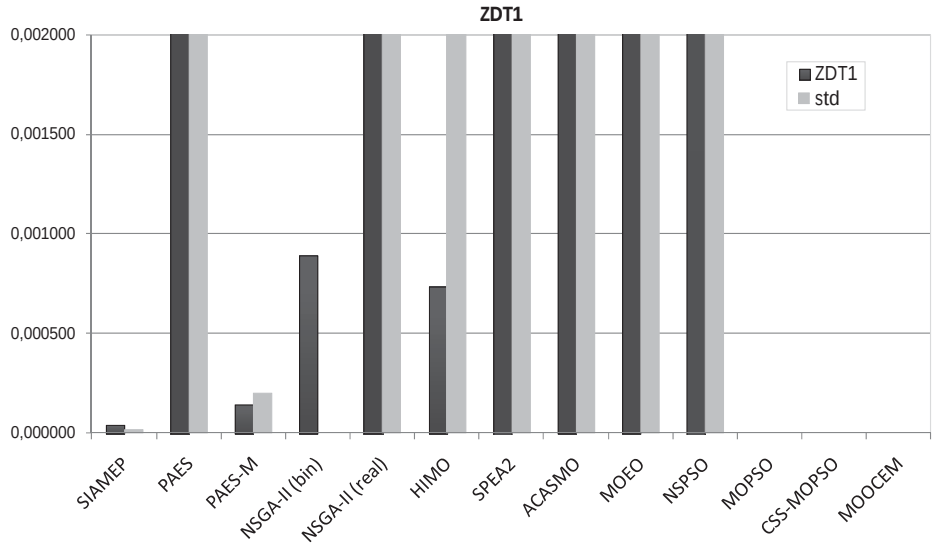


Figura 4.15: Resultado da medida de distribuição Δ para ZDT1.

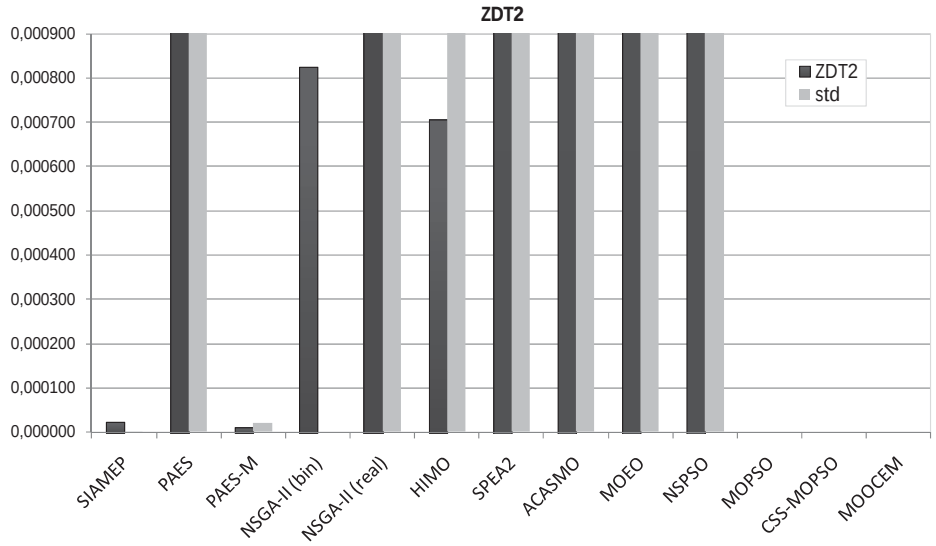


Figura 4.16: Resultado da medida de distribuição Δ para ZDT2.

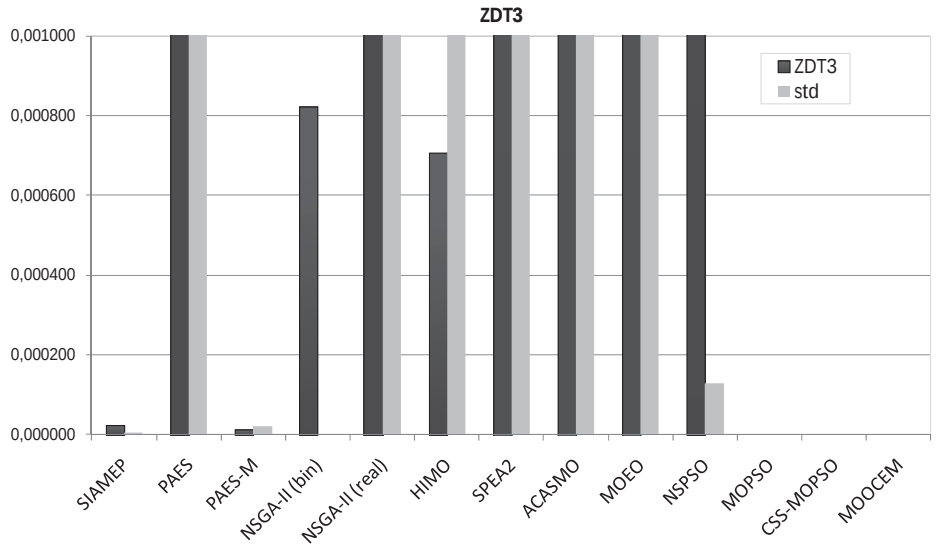


Figura 4.17: Resultado da medida de distribuição Δ para ZDT3.

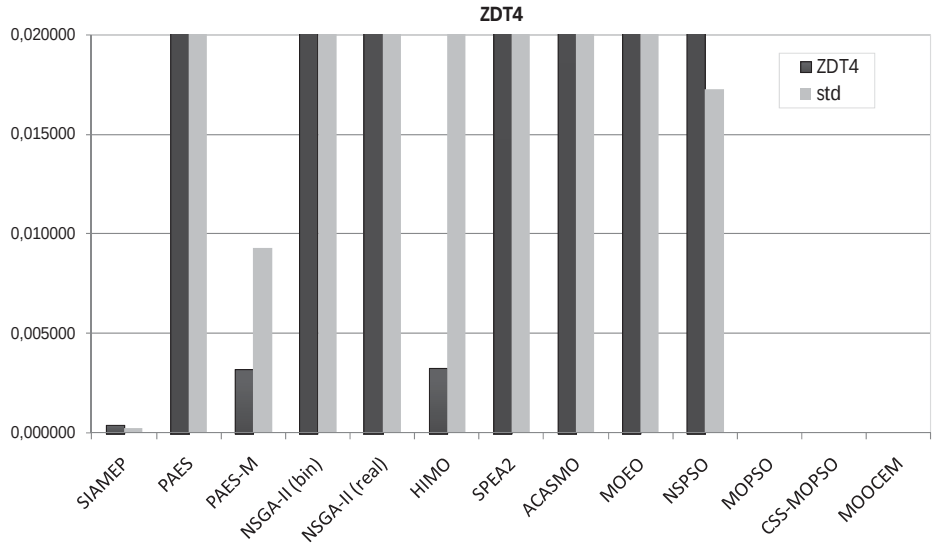


Figura 4.18: Resultado da medida de distribuição Δ para ZDT4.

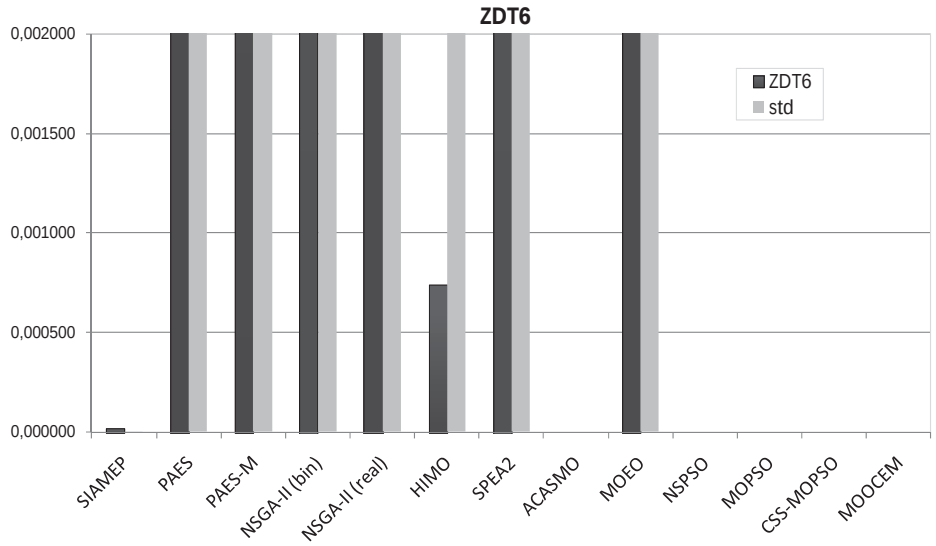


Figura 4.19: Resultado da medida de distribuição Δ para ZDT6.

Conforme mencionado em VIEIRA (2009), aparentemente nenhum algoritmo de otimização bio-inspirado é capaz de resolver bem todos os problemas, geralmente alguma especificidade do problema (função objetivo) faz com que um ou outro algoritmo funcione de forma mais satisfatória em função de seus mecanismos de busca específicos. Com isso, pode-se considerar o resultado do algoritmo proposto como sendo adequado uma vez que este apresenta bons resultados para todas as funções matemáticas utilizadas nos testes.

4.4 PROBLEMAS COM RESTRIÇÃO

Nesta seção serão tratados problemas de otimização multi-objetivos com funções contendo restrições. As funções foram escolhidas da literatura (BARDANACHVILI, 2006; CASTRO, R. E., 2001, DEB, 2001 e outros) para uma comparação de desempenho do

algoritmo. Para estes testes o conjunto de pareto possui 40 indivíduos e o número máximo de gerações foi de 3.000. O critério de parada é o número máximo de avaliações apresentado individualmente para cada um dos testes e a mutação é a híbrida. Os testes CONSTR_EX e TNK foram rodados 30 vezes e os resultados estão apresentados em termos de média e desvio padrão. Os testes em estruturas foram executados apenas uma vez. Os resultados dos melhores frontes gerados em cada um das rodadas foram combinados ao final da seqüência de rodadas para dar origem a somente um único fronte de pareto que será exposto nas figuras apresentadas.

Tabela 4.11: Configurações do algoritmo para alguns dos testes com Restrição

PARÂMETROS	CONSTR_EX	TNK	VIGA BI-APOIADA	TRELIÇA DUAS BARRAS	VIGA SOLDADA
Num. Soluções Fronte Exato	679	152	---	---	---
Num. máximo de avaliações	5.000	10.000	4.000	10.000	20.000
Num. de Anticorpos (N_{ab}) e Clones (N_{clones})	[10 e 6]	[10 e 5]	[10 e 6]	[8 e 6]	[8 e 6]
Tamanho População	*70*	*60*	*70*	*56*	*56*
Par. mutação somática (β_m)	$\beta_1 = 1$ $\beta_2 = 3$	$\beta_1 = 0,5$ $\beta_2 = 0,5$	$\beta_1 = 20, \beta_2 = 50$ $\beta_3 = 2, \beta_4 = 2$	$\beta_1 = 1$ $\beta_2 = 0,005$ $\beta_3 = 0,005$	$\beta_1, \beta_2 = 2,4375$ $\beta_3, \beta_4 = 4,95$
Par. mutação somática ($\beta_{m-final}$) (*)	$\beta_m \cdot 10^{-2}$	$\beta_m \cdot 10^{-3}$	$\beta_m \cdot 10^{-2}$	$\beta_m \cdot 10^{-5}$	$\beta_m \cdot 10^{-4}$
Par. mutação somática (ρ)	[1, 3]	2	[1, 3]	2	2
Escopo mutação somática	Todos os parâmetros			Um parâmetro	
Par. mutação diferencial (*F*)	0,8				
Escopo mutação diferencial	Todos os parâmetros podem sofrer mutação				

(*) $\beta_{m-final}$ representa o valor de β_m após sofrer um decaimento linear com o num. de avaliações ou gerações.

Para estes testes os parâmetros do algoritmo foram configurados para possibilitar os melhores resultados com o menor número de avaliações da função objetivo, ou seja, aqui os parâmetros foram sendo ajustados sucessivamente até darem resultados melhores que os apresentados pelos algoritmos referenciados. O parâmetro β_m foi ajustado como sendo igual a metade do intervalo das variáveis.

4.4.1 CONSTR-EX

CONSTR_EX é o problema Min-Ex, utilizado por DEB (2001), mas com duas restrições. Sua formulação está apresentada na expressão (4.13).

$$CONSTR-EX:\begin{cases} f_1(x) = x_1 \\ f_2(x) = \dfrac{1+x_2}{x_1} \\ sujeito\ a: \\ g_1(x) = -x_2 - 9x_1 + 6 \le 0 \\ g_2(x) = +x_2 - 9x_1 + 1 \le 0 \\ 0{,}1 \le x_1 \le 1\ e\ 0 \le x_2 \le 5 \end{cases} \tag{4.13}$$

O fronte de pareto para o problema Min-Ex sem restrições é dado por uma hipérbole $f_2 = 1/f_1$. Com a restrição 1, há uma interrupção no fronte original, para seguir a fronteira por ela definida.

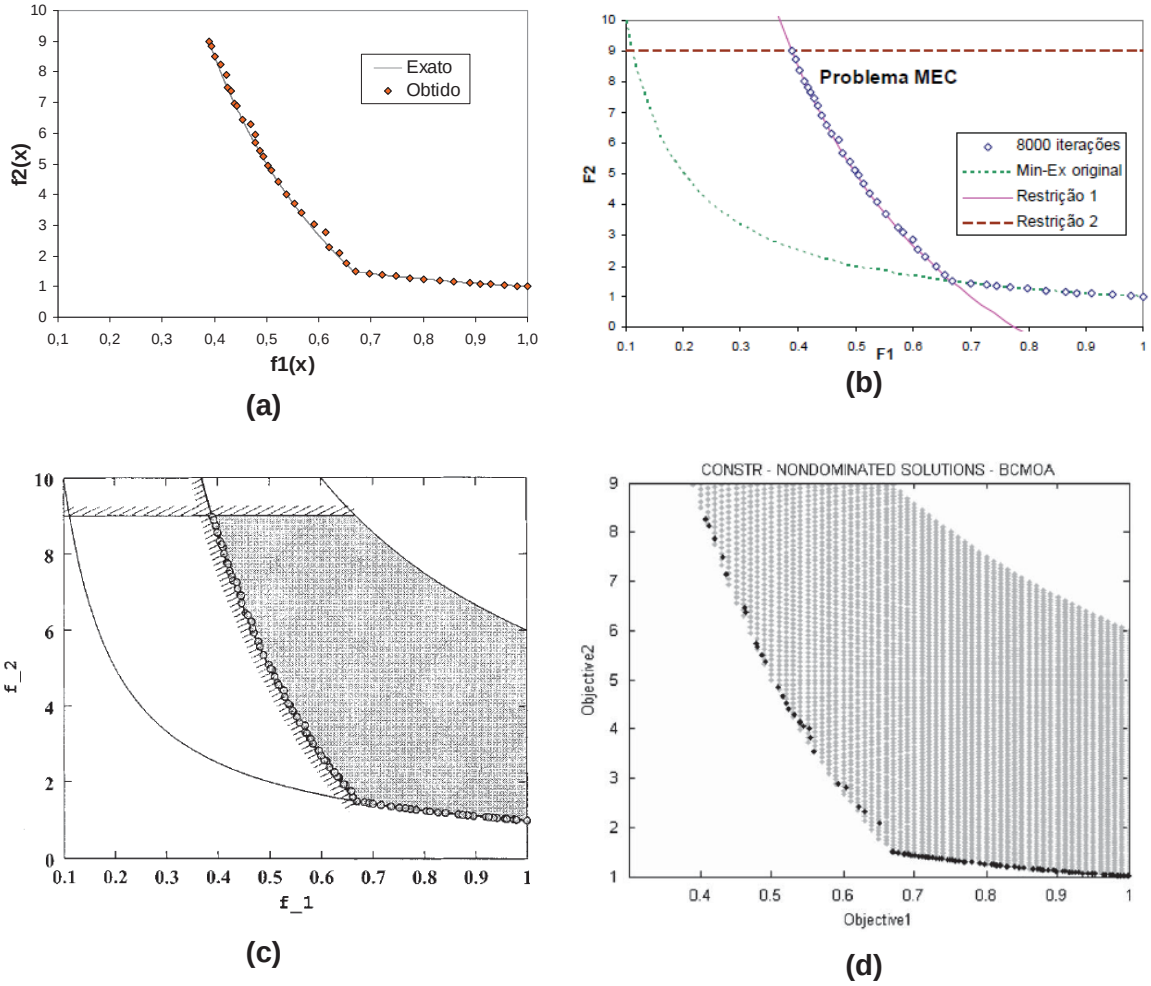


Figura 4.20: Resultados para CONSTR-EX: **(a)** SIAMEP; **(b)** PAES-M; **(c)** NSGA-II (DEB *et al.*, 2002); e **(d)** BCMOA (GUZMÁM *et al.*, 2010)

A Figura 4.20 (a) apresenta o resultado obtido pelo algoritmo proposto para uma execução com 5.000 avaliações da função objetivo. A Figura 4.20 (b) apresenta o resultado obtido em BARDANACHVILI (2006) com 8.000 avaliações. A Figura 4.20 (c)

apresenta o resultado obtido pelo algoritmo NSGA-II (DEB *et al.*, 2002) com 100 indivíduos no conjunto de pareto obtido com 50.000 avaliações. Não foram apresentadas métricas de desempenho para esta função com o NSGA-II. E, a Figura 4.20 (d) apresenta o resultado obtido pelo algoritmo BCMOA (GUZMÁM *et al.*, 2010), no entanto, não foram informados o número de avaliações nem o número de soluções no conjunto de pareto e também não foram apresentadas métricas de desempenho para esta função.

Conforme se pode perceber pela análise da Figura 4.20 e da Tabela 4.12, o algoritmo proposto foi capaz de alcançar um resultado superior ao de PAES-M com um número de avaliações inferior, apresentando também uma boa distribuição do fronte de pareto, que é inferior à de PAES-M em termos da medida de distribuição Δ, mas é visualmente bom.

Tabela 4.12: Resultados comparativos CONSTR-EX

RESULTADO		SIAMEP (5.000)	PAES-M (8.000)	PAES-M (24.000)
GD	Média	**0,001328**	0,464124	0,468321
	STD	*0,000331*	*0,023285*	*0,012583*
SP	Média	0,005488		
	STD	*0,000685*		
Δ	Média	0,605210	0,151443	**0,133778**
	STD	*0,058448*	*0,034012*	*0,015496*

4.4.2 TNK

Proposta por TANAKA *et al.* (1995) esta função possui um fronte de pareto descontínuo. DEB (2002) testou esta função em seu algoritmo NSGA-II. Sua formulação está apresentada na expressão (4.14).

$$TNK: \begin{cases} f_1(x) = x_1; \;\; f_2(x) = x_2 \\ sujeito\ a: \\ g_1(x) = x_1^2 + x_2^2 - 1 - 0{,}1\cos\left[16\arctan\left(\frac{x_1}{x_2}\right)\right] \leq 0 \\ g_2(x) = (x_1 - 0{,}5)^2 + (x_2 - 0{,}5)^2 - 0{,}5 \leq 0 \\ 0 \leq x_1 \leq \pi \; e \; 0 \leq x_2 \leq \pi \end{cases} \tag{4.14}$$

A Figura 4.21(a) apresenta o resultado obtido pelo algoritmo proposto para uma execução com 10.000 avaliações da função objetivo. A Figura 4.21 (b) apresenta o resultado obtido em BARDANACHVILI (2006) com 50.000 avaliações. A Figura 4.21 (c) apresenta o resultado obtido pelo algoritmo VAIS (FRESCHI & REPETTO, 2005) com

12.000 avaliações da função objetivo. E, a Figura 4.21 (d) apresenta o resultado obtido pelo algoritmo NSGA-II (DEB *et al.*, 2002) com 50.000 avaliações. Não foram apresentadas métricas de desempenho para esta função com o NSGA-II.

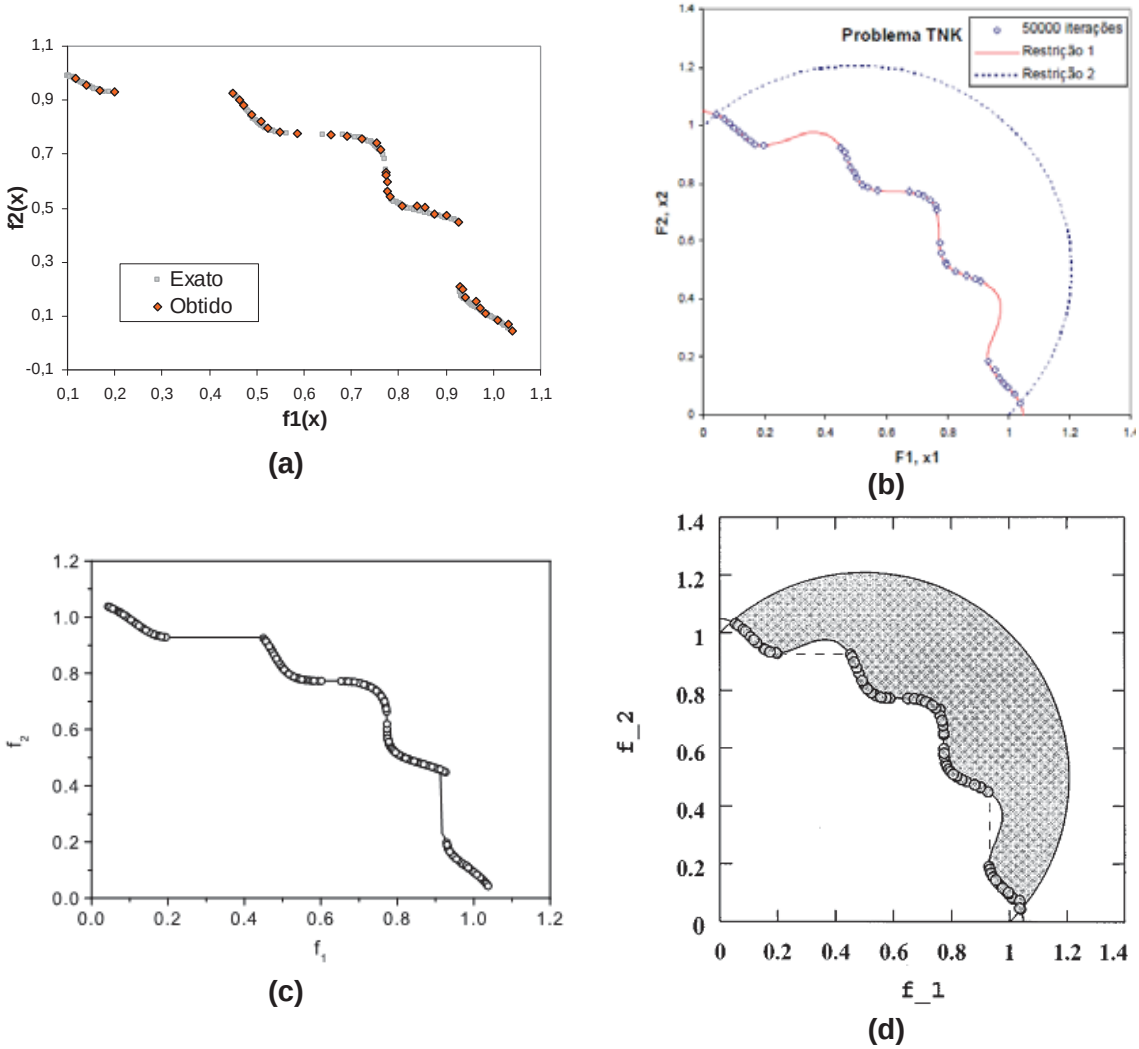


Figura 4.21: Resultados para TNK: **(a)** com SIAMEP; **(b)** com PAES-M; (c) VAIS (FRESCHI & REPETTO, 2005); e **(d)** NSGA-II (DEB *et al.*, 2002);

Conforme se pode perceber pela análise da Figura 4.21 e da Tabela 4.13, o algoritmo proposto foi capaz de alcançar um resultado superior ao de PAES-M em relação a GD, com um número de avaliações bem inferior, apresentando também uma distribuição visualmente aceitável do fronte de pareto. Ainda assim, a distribuição do fronte de pareto é inferior à de PAES-M em termos da medida de distribuição Δ. Cabendo ressaltar que o PAES-M utilizou-se de um número bem maior de avaliações da função objetivo. O algoritmo NSGA-II não apresenta tabela com resultados para os valores das métricas de desempenho medidas. Em relação ao parâmetro SP o algoritmo VAIS foi superior, apresentando um valor ligeiramente inferior ao do SIAMEP, mas com um número maior de avaliações da função objetivo.

Tabela 4.13: Resultados comparativos TNK

RESULTADO		SIAMEP (10.000)	PAES-M (50.000)	PAES-M (200.000)	VAIS (12.000)
GD	Média	**0,001481**	0,111288	0,111523	
	STD	*0,000344*	*0,001573*	*0,002243*	
SP	Média	0,007046			0,002010
	STD	*0,001113*			*0,000277*
Δ	Média	0,546725	**0,134559**	0,161537	
	STD	*0,025904*	*0,021200*	*0,038487*	

4.4.3 VIGA bi-apoiada (perfil em I)

Este problema foi apresentado por COELLO & CHRISTIANSEN (1998) e foi abordado em CASTRO, R. E. (2001). O problema consiste em determinar o conjunto ótimo de pareto para a otimização da viga bi-apoiada em seção I apresentada a seguir.

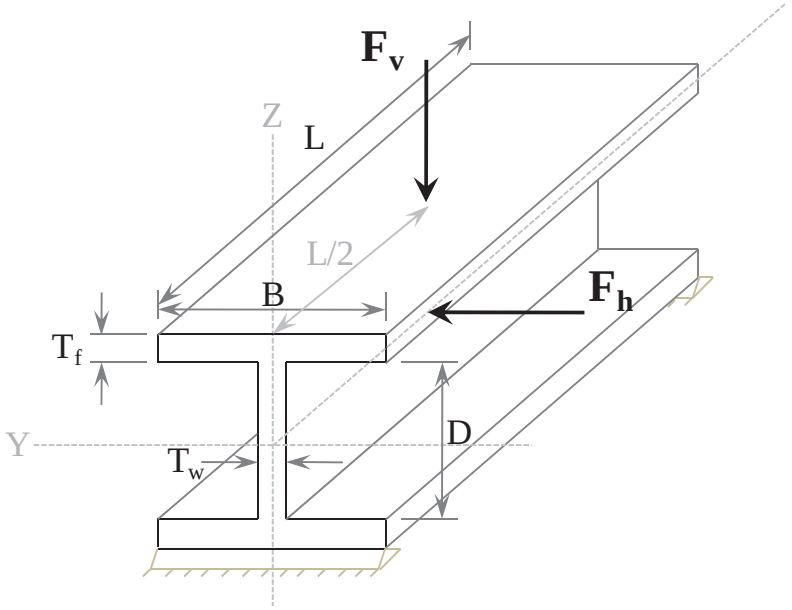


Figura 4.22: Viga Bi-apoiada em seção I

A viga tem como características:

- $E = 20.000$ kN/cm^3 (módulo de Young, ou módulo de elasticidade);
- $\sigma_{max} = 16$ kN/cm^2 (tensão máxima admissível);
- $\delta_{max} = 0,5$ cm (deslocamento máximo admissível);
- $F_v = 600$ kN (carga vertical em *L/2*);
- $F_h = 50$ kN (carga horizontal em *L/2*);

- $L = 200$ cm (comprimento da viga);

Onde *B* é a largura das mesas; *D* é a altura total da viga; T_w é a espessura da viga; T_f é a espessura das mesas; *A* é a área da seção transversal da viga; *Iy* e *Iz* são os momentos de inércia em relação aos eixos y e z da seção, respectivamente; W_y e W_z são os módulos resistentes da seção em relação aos eixos y e z; $\delta(x)$ é o deslocamento estático máximo;

O problema consiste em resolver:

$$\text{Minimizar: } \begin{cases} A = 2 \cdot B \cdot T_f + T_w(D - 2T_f) = \text{área da seção} = f_1 \\ \delta(x) = \dfrac{F_v L^3}{48 E Iy(x)} = \text{deslocamento} = f_2 \end{cases} \tag{4.15}$$

$$\text{Sujeito a restrições: } \begin{cases} g_1 = \dfrac{M_y}{W_y} + \dfrac{M_z}{W_z} - \sigma_{\max} \leq 0 \\ g_2 = \delta(x) - \delta_{\max} \leq 0 \end{cases}$$

Onde as variáveis livres e seus limites de variação são: ***B*** $(10 \leq B \leq 50)$, ***D*** $(10 \leq D \leq 80)$, $\boldsymbol{T_w}$ $(0{,}9 \leq T_w \leq 5)$ e $\boldsymbol{T_f}$ $(0{,}9 \leq T_f \leq 5)$.

Os módulos resistentes podem ser calculados pela equação a seguir:

$$W_y = \frac{T_w(D - 2T_f)^3 + 2BT_f\left[4(T_f)^2 + 3D(D - 2T_f)\right]}{6D}, \text{ e}$$

$$W_z = \frac{(D - 2T_f)T_w^{\,3} + 2T_f B^3}{6B} \tag{4.16}$$

Os momentos nas seções são calculados pela equação a seguir:

$$M_y = \frac{F_v L}{4}, \text{ e } M_z = \frac{F_h L}{4} \tag{4.17}$$

O momento de inércia em torno do eixo y é dado pela equação seguinte:

$$I_y = \frac{T_w(D - 2T_f)^3 + 2BT_f\left[4(T_f)^2 + 3D(D - 2T_f)\right]}{12} \tag{4.18}$$

Foram executados dois testes, um com 40 indivíduos no conjunto de pareto (SIAMEP 40) e outro com 20 indivíduos no conjunto de pareto (SIAMEP 20). Os testes

foram executados com o critério de parada sendo o número máximo de avaliações da função objetivo, pré-estabelecido no valor de 4.000 avaliações. Os resultados obtidos estão apresentados nas Figura 4.23 e Figura 4.24.

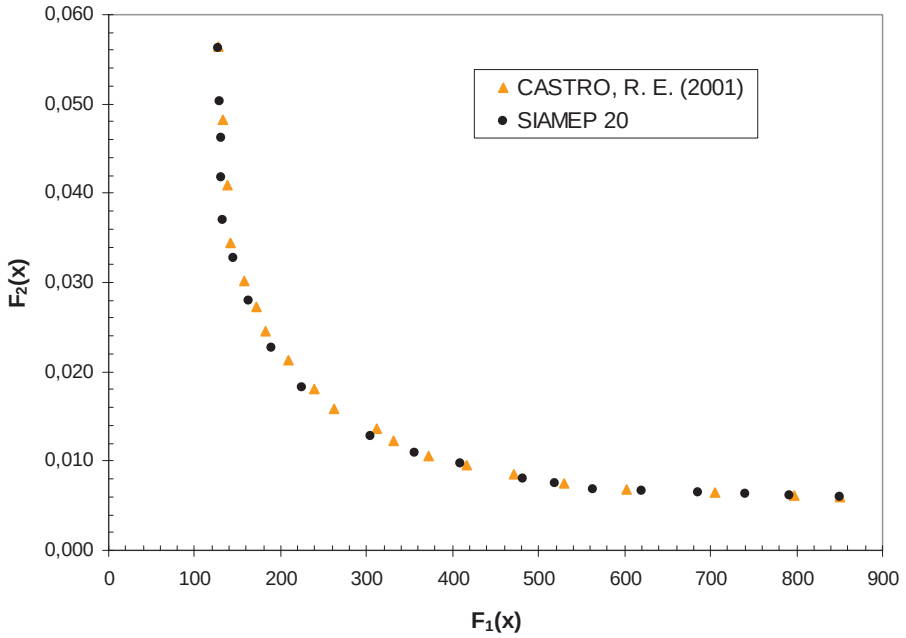


Figura 4.23: Resultado comparativo para o problema da Viga bi-apoiada

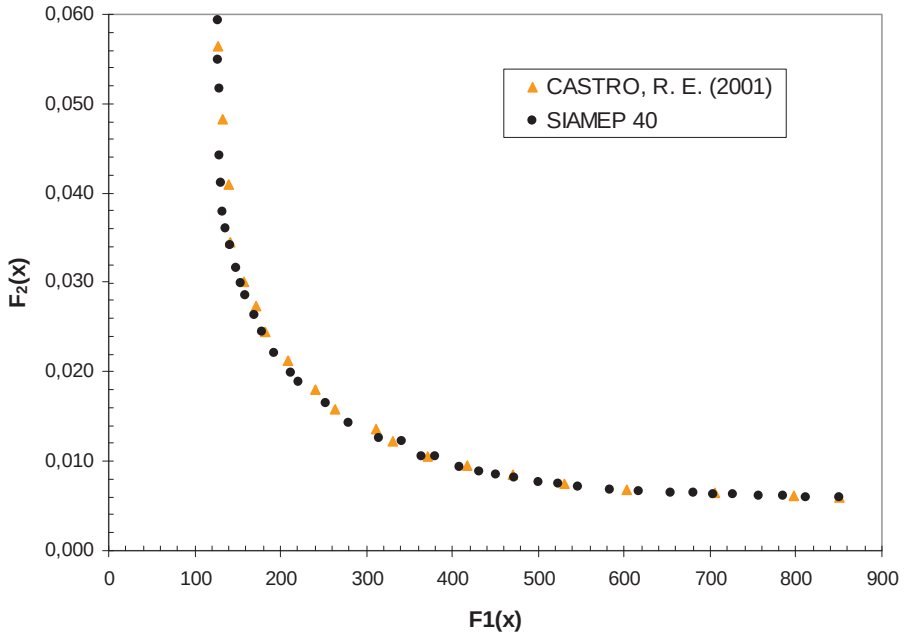


Figura 4.24: Resultado comparativo para o problema da Viga bi-apoiada

Como se pode notar visualmente, o algoritmo desenvolvido foi capaz de alcançar os valores dos melhores resultados obtidos por CASTRO, R. E. (2001) e superá-los em algumas regiões, como é o caso da parte superior da função. Além disso, nota-se visualmente uma boa distribuição dos pontos ao longo do fronte de pareto, tanto com 20 quanto com 40 soluções no fronte.

Em COELLO & CHRISTIANSEN (1998) são apresentados os resultados dos vetores ideais de diferentes algoritmos para o exemplo da Viga bi-apoiada. Os resultados podem ser vistos na Tabela 4.14 a seguir. Em negrito estão marcados os melhores resultados para cada um dos algoritmos.

Tabela 4.14: Resultados da viga bi-apoiada para COELLO & CHRISTIANSEN (1998)

MÉTODO	F_1	F_2	D	B	T_W	T_F
Monte Carlo 1	**188,65**	0,06175	30,84	28,26	3,79	4,06
	55,22	**0,00849**	52,97	44,08	1,99	0,99
Min-Max (OS)	**316,85**	0,01697	74,97	44,97	1,97	1,97
	326,49	**0,01636**	74,97	44,99	1,99	2,06
GA (Binário)	**128,27**	0,05241	66,39	38,63	0,90	0,91
	848,41	**0,00591**	80,00	50,00	4,99	4,99
GA (real)	**127,46**	0,06034	61,14	41,14	0,90	0,90
	850,00	**0,00590**	80,00	50,00	5,00	5,00
Literatura	**128,47**	0,06000	60,70	49,90	0,90	0,90
	850,00	**0,00590**	80,00	50,00	5,00	5,00

Os valores dos parâmetros obtidos para o SIAMEP 40 estão apresentados na Tabela 4.15 a seguir.

Tabela 4.15: Valores obtidos pelo SIAMEP 40.

F_1	F_2	D	B	T_W	T_F
127,44	0,0613	41,4	60,5	0,90	0,90
127,67	0,0593	41,0	61,7	0,90	0,90
128,03	0,0548	39,7	64,6	0,90	0,90
129,69	0,0516	39,1	66,7	0,90	0,91
129,84	0,0443	36,4	73,2	0,90	0,90
131,09	0,0412	35,7	76,0	0,90	0,90
132,93	0,0379	35,1	79,2	0,90	0,90
135,98	0,0361	36,4	80,0	0,90	0,90
141,89	0,0341	39,1	80,0	0,90	0,91
148,68	0,0316	43,5	80,0	0,90	0,90
154,21	0,0299	46,6	80,0	0,90	0,90
159,31	0,0286	49,4	80,0	0,90	0,90
169,61	0,0264	36,2	80,0	1,38	0,90
179,40	0,0244	42,6	80,0	1,29	0,90
193,37	0,0221	50,0	80,0	1,24	0,90
211,42	0,0198	41,4	80,0	1,72	0,90
221,41	0,0188	38,5	80,0	1,99	0,90
252,71	0,0165	47,0	78,6	1,97	0,90
278,68	0,0143	48,3	80,0	2,16	0,92
314,43	0,0127	39,9	80,0	3,08	0,93
342,13	0,0123	48,3	76,8	2,88	0,90
365,12	0,0106	50,0	80,0	2,98	0,90
379,55	0,0105	35,7	80,0	4,42	0,90
408,87	0,0094	48,0	80,0	3,58	0,90
431,87	0,0089	49,4	80,0	3,71	0,90
451,72	0,0085	50,0	80,0	3,84	0,94
471,91	0,0082	46,4	80,0	4,40	0,90
499,80	0,0077	48,7	80,0	4,45	0,93
522,71	0,0074	46,0	80,0	5,00	0,90
546,44	0,0071	48,3	80,0	5,00	0,90

F_1	F_2	D	B	T_W	T_F
582,69	0,0068	50,0	80,0	5,00	1,18
617,36	0,0066	50,0	80,0	5,00	1,68
653,50	0,0065	50,0	80,0	5,00	2,19
680,81	0,0064	50,0	80,0	5,00	2,58
703,85	0,0063	50,0	80,0	5,00	2,91
727,41	0,0063	50,0	80,0	5,00	3,25
756,68	0,0062	50,0	80,0	5,00	3,67
784,88	0,0061	50,0	80,0	5,00	4,07
812,29	0,0060	50,0	80,0	5,00	4,46
850,00	0,0059	50,0	80,0	5,00	5,00

Os valores dos parâmetros obtidos por CASTRO, R. E. (2001) estão apresentados na Tabela 4.16 a seguir.

Tabela 4.16: Valores obtidos por CASTRO, R. E. (2001).

F_1	F_2	D	B	T_W	T_F
127,60	0,0565	63,6	40,0	0,90	0,90
132,20	0,0482	68,1	40,3	0,90	0,90
139,00	0,0409	72,4	41,3	0,91	0,91
141,70	0,0345	79,1	40,1	0,90	0,90
158,10	0,0301	78,3	45,7	0,92	0,96
171,80	0,0273	79,7	23,0	0,96	2,15
183,00	0,0245	78,9	40,1	0,92	1,41
208,70	0,0213	78,9	43,9	1,00	1,51
240,00	0,0181	78,6	34,8	0,96	2,42
262,70	0,0158	79,7	46,6	1,02	1,99
311,50	0,0136	78,0	42,2	1,04	2,80
331,40	0,0123	78,6	45,1	0,93	2,92
372,00	0,0106	79,8	40,9	0,93	3,72
416,70	0,0095	79,9	40,0	1,01	4,30
470,50	0,0085	80,0	47,5	1,29	3,97
529,80	0,0075	79,6	50,0	1,15	4,49
602,30	0,0068	80,0	48,7	1,66	4,99
705,00	0,0064	79,6	50,0	2,95	5,00
796,60	0,0062	79,7	50,0	4,42	4,87
850,00	0,0059	80,0	50,0	5,00	5,00

Pode-se perceber comparando a Tabela 4.14 com a Tabela 4.15 que o algoritmo SIAMEP foi capaz de alcançar os resultados mais extremos (grifados em negrito na Tabela 4.14) dos algoritmos apresentados em COELLO & CHRISTIANSEN (1998), ou seja, SIAMEP alcançou os pontos (850,00; 0,00590) e melhor do que (127,46; 0,06034), chegando a (127,44; 0,0613) para os valores de F_1 e F_2 respectivamente. Estes resultados confirmam mais uma vez o bom funcionamento do algoritmo SIAMEP.

4.4.4 Treliça de duas barras

Este problema foi proposto por DEB *et al.* (2000b) e consiste em minimizar o volume de aço da treliça (o que equivalente a projetar com o menor custo de fabricação) e adicionalmente minimizar os esforços em cada um dos dois membros AC e BC. A treliça (ver Figura 4.25) deve ser capaz de suportar o carregamento sem sofrer falha por

excesso de deformação elástica. Assim, foi construído o seguinte problema de minimização com dois objetivos e três variáveis livres sendo y, a distância vertical entre B e C em metros, a_1, a área da seção transversal do segmento AC em m² e a_2, a área da seção transversal do segmento BC em m² .

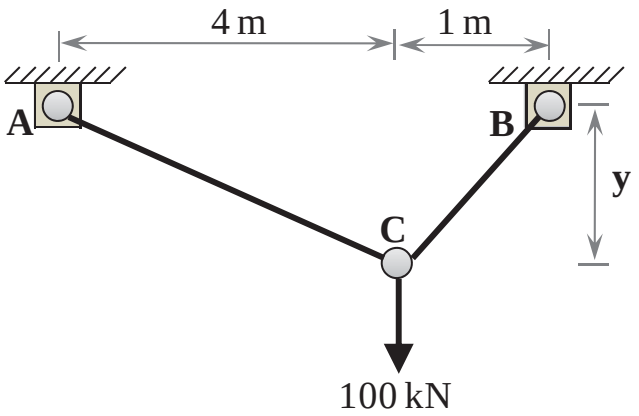


Figura 4.25: Treliça de duas barras

O problema consiste em resolver:

$$\text{Minimizar: } \begin{cases} f_1 = a_1\left(\sqrt{16+y^2}\right)+a_2\left(\sqrt{1+y^2}\right) \\ f_2 = \max\left(\sigma_{AC},\sigma_{BC}\right) \end{cases} \tag{4.19}$$

Sujeito a restrições: $g_1 = \max\left(\sigma_{AC},\sigma_{BC}\right) \le 1(10^5)$

$$1 \le y \le 3;\ a_1 \ge 0\ e\ a_2 \ge 0;$$

As tensões normais nas barras são calculadas como se segue:

$$\sigma_{AC} = \frac{20\sqrt{16+y^2}}{y \cdot a_1}\ \ e\ \ \sigma_{BC} = \frac{80\sqrt{1+y^2}}{y \cdot a_2} \tag{4.20}$$

Onde σ_{AC} e σ_{BC} são as tensões respectivamente nas barras AC e BC em kPa.

Para limitar os resultados aos obtidos em PALLI *et al.* (1999) e compará-los com os obtidos em DEB *et al.* (2000b) foram adicionadas as restrições de máxima tensão sendo menor que $1(10^5)$ usando uma penalidade de 10^3 caso esta restrição seja violada. E ainda a_1 e a_2 foram limitados ao intervalo $0 \le a_1, a_2 \le 0,01$.

Os resultados obtidos por DEB *et al.* (2000b) estão apresentados na Figura 4.26.

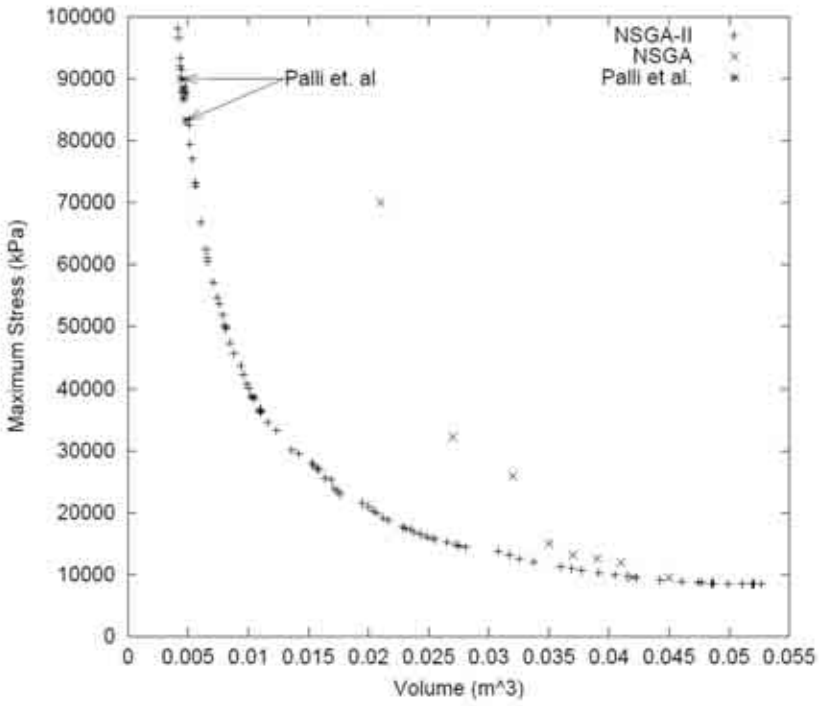


Figura 4.26: Resultados Treliça de duas barras (DEB *et al.*, 2000b)

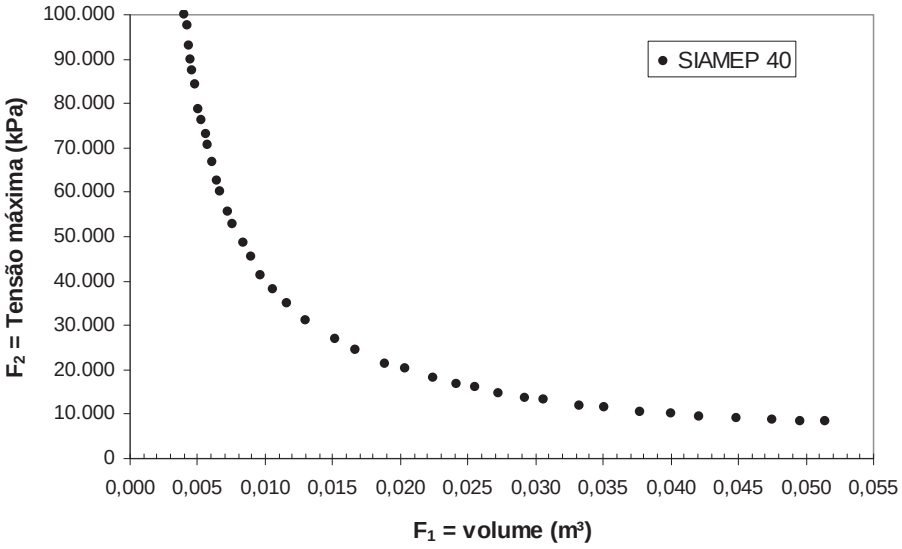


Figura 4.27: Resultados Treliça de duas barras

A partir da análise comparativa entre as Figura 4.26 e Figura 4.27 nota-se que os resultados do algoritmo proposto (SIAMEP) são perfeitamente satisfatórios apresentado excelente distribuição e convergência. A seguir temos a tabela com os resultados expostos no gráfico.

Tabela 4.17: Resultados Treliça de duas Barras – SIAMEP 40

F_1	F_2	y	a_1	a_2
0,05139	8432,7	3,000	0,00395	0,01000
0,04958	8506,8	2,766	0,00415	0,01000
0,04748	8649,8	2,432	0,00453	0,01000
0,04486	8925,6	2,021	0,00498	0,01000

F_1	F_2	y	a_1	a_2
0,04207	9528,6	2,014	0,00467	0,00942
0,04002	10021,6	1,909	0,00463	0,00904
0,03770	10618,9	1,992	0,00423	0,00843
0,03515	11398,2	1,905	0,00408	0,00793
0,03327	12057,3	1,859	0,00394	0,00754
0,03063	13113,2	2,061	0,00333	0,00683
0,02918	13740,3	2,047	0,00320	0,00651
0,02732	14839,3	1,943	0,00317	0,00606
0,02553	15962,6	2,052	0,00285	0,00558
0,02413	16680,3	2,093	0,00259	0,00537
0,02242	18019,5	2,043	0,00244	0,00504
0,02031	20180,6	2,298	0,00199	0,00444
0,01888	21265,2	2,061	0,00207	0,00418
0,01665	24302,1	2,324	0,00164	0,00358
0,01523	27028,1	1,990	0,00166	0,00350
0,01296	30964,8	1,876	0,00152	0,00293
0,01163	34874,9	2,114	0,00123	0,00260
0,01061	37955,9	1,936	0,00121	0,00240
0,00972	41270,5	1,920	0,00112	0,00220
0,00895	45482,6	1,927	0,00105	0,00198
0,00837	48605,8	1,699	0,00105	0,00192
0,00762	52707,6	1,850	0,00091	0,00173
0,00730	55671,9	2,029	0,00082	0,00160
0,00668	60039,3	1,870	0,00079	0,00151
0,00640	62579,9	2,047	0,00070	0,00142
0,00613	66654,0	1,943	0,00071	0,00135
0,00570	70757,5	1,909	0,00066	0,00128
0,00558	72937,3	2,055	0,00062	0,00122
0,00534	76301,1	2,337	0,00052	0,00115
0,00510	78834,3	2,060	0,00055	0,00114
0,00483	84238,9	1,753	0,00060	0,00109
0,00459	87547,9	1,960	0,00052	0,00104
0,00454	89881,3	2,102	0,00048	0,00102
0,00437	93060,1	1,873	0,00052	0,00097
0,00420	97418,7	1,713	0,00052	0,00097
0,00401	99996,5	1,961	0,00045	0,00090

4.4.5 Viga Soldada

Proposto por DEB *et al.* (2000b) o problema consiste em uma viga que precisa ser soldada a outro componente estrutural e deve suportar certo carregamento F_v (ver Figura 4.28). As variáveis livres são: espessura da solda (h)*;* comprimento da solda (l); Altura da viga (t) e largura da viga (b), todas em polegadas (*in*).

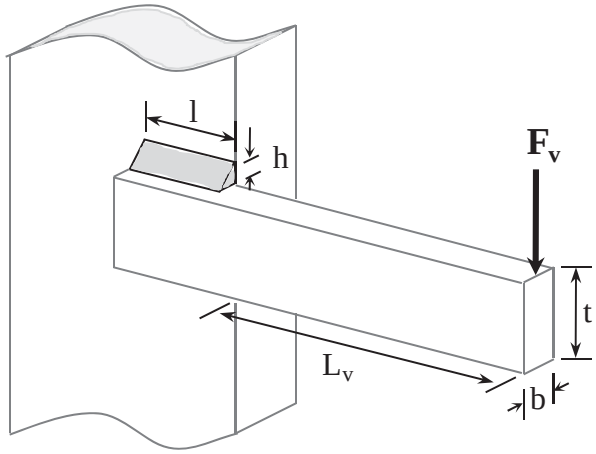


Figura 4.28: Viga soldada

Os objetivos do projeto são minimizar o custo de fabricação (representado pelo volume de material) e minimizar o deslocamento na extremidade livre da viga.

O problema consiste em resolver:

$$\text{Minimizar: } \begin{cases} f_1(\boldsymbol{x}) = 1{,}10471\left(h^2 l\right) + 0{,}04811\left(tb\right)\left(14{,}0 + l\right) \rightarrow custo \\ f_2(\boldsymbol{x}) = \dfrac{2{,}1952}{t^3 b} \rightarrow deslocamento \end{cases}$$

$$\text{Sujeito a restrições: } \begin{cases} g_1(\boldsymbol{x}) \equiv \tau(\boldsymbol{x}) - \tau_{\max} \leq 0 \\ g_2(\boldsymbol{x}) \equiv \sigma(\boldsymbol{x}) - \sigma_{\max} \leq 0 \\ g_3(\boldsymbol{x}) \equiv F_v - P_c(\boldsymbol{x}) \leq 0 \\ g_4(\boldsymbol{x}) \equiv \left(\dfrac{2{,}1952}{t^3 b}\right) - u_{\max} \leq 0 \\ g_5(\boldsymbol{x}) \equiv h - b \leq 0 \end{cases} \qquad (4.21)$$

$$0{,}125 \leq h, b \leq 5{,}0 \;\; e \;\; 0{,}1 \leq l, t \leq 10{,}0$$

Onde $\tau(\boldsymbol{x})$ é a tensão de cisalhamento ao longo do apoio da viga; $\tau_{\max}$ é a máxima tensão admissível de cisalhamento do material; $\sigma(\boldsymbol{x})$ é a tensão normal ao longo do apoio da viga; $\sigma_{\max}$ é a máxima tensão normal admissível do material; e $\delta_{máx}$ é o limite máximo de deslocamento na extremidade da viga.

Existem cinco restrições ao problema. As restrições g_1 e g_2 garantem que a tensão de cisalhamento e a tensão normal desenvolvidas ao longo do apoio da viga sejam respectivamente menores que as tensões admissíveis de cisalhamento e normal do material. A restrição g_3 garante que o esforço resistente (ao longo da direção *t*) da viga seja menor que a carga aplicada F_v. A restrição g_4 estabelece um limite máximo ($\delta_{máx}$)

para o deslocamento na extremidade da viga. E a restrição g_5 garante que a largura da viga não seja menor do que a espessura da solda (CASTRO, R. E., 2001). As tensões são calculadas como se segue.

Tensão cisalhante:

$$\tau(x) = \sqrt{\tau'^2 + \tau''^2 + l(\tau'\tau'')\sqrt{0,25\left[l^2 + \left(h+t\right)^2\right]}}$$

$$\tau' = \frac{6000}{hl\sqrt{2}} \qquad (4.22)$$

$$\tau'' = \frac{6000\left(14+0,5l\right)\sqrt{0,25\left[l^2+\left(h+t\right)^2\right]}}{2\left\{0,707hl\left[\frac{l^2}{12}+0,25\left(h+t\right)^2\right]\right\}}$$

Tensão normal:

$$\sigma(x) = \frac{504000}{t^2 b} \qquad (4.23)$$

Esforço resistente ao longo da direção *t*:

$$P_c(x) = 64746,022\left(1-0,0282346t\right)tb^3 \qquad (4.24)$$

Limites e dados do problema:

$$F = 6000\,lb$$
$$E = 30x10^6\ psi$$
$$G = 12x10^6\ psi$$
$$L_v = 14\ in\ (comp.\ livre)$$

e

$$\tau_{\max} = 13600\ psi$$
$$\sigma_{\max} = 30000\ psi$$
$$\delta_{\max} = 0,25\ in$$

As unidades não estão no Sistema Internacional (SI) para comparação com os resultados de obtidos por DEB *et al.* (2000b) que estão apresentados na Figura 4.29. Os resultados de CASTRO, R. E. (2001) e do o algoritmo SIAMEP estão apresentados na Figura 4.30, Tabela 4.18 e Tabela 4.19.

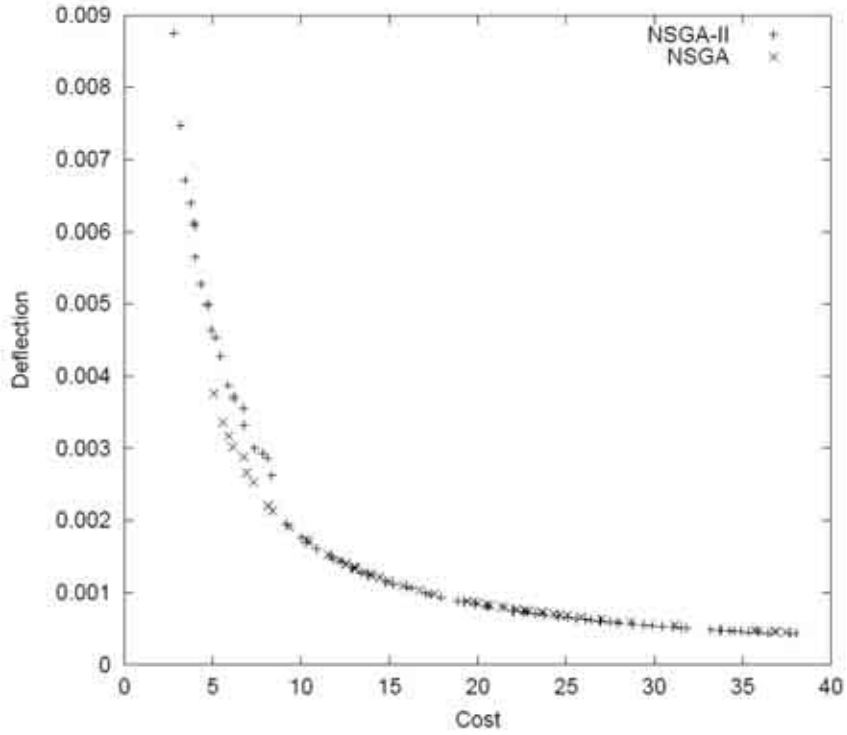


Figura 4.29: Resultados para Viga Soldada (DEB *et al.*, 2000b)

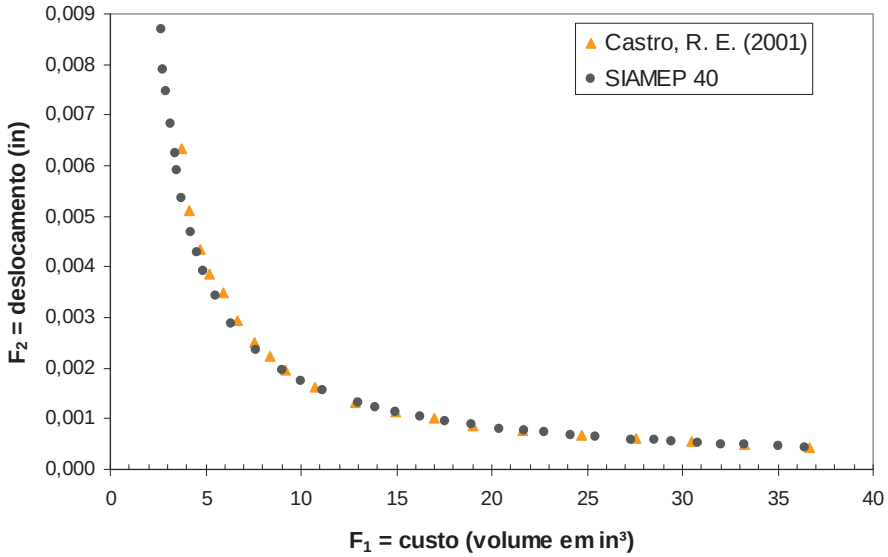


Figura 4.30: Resultados para Viga Soldada

Tabela 4.18: Resultados Viga Soldada para CASTRO, R. E. (2001)

F_1	F_2	h	l	t	b
36,630	0,00044	1,182	0,748	10,000	5,000
33,250	0,00049	1,182	0,748	10,000	4,524
30,470	0,00054	1,334	0,750	10,000	4,086
27,590	0,00060	1,182	0,754	9,923	3,753
24,680	0,00068	1,184	0,787	9,922	3,324
21,650	0,00077	0,870	1,057	10,000	2,867
19,040	0,00087	1,182	0,753	10,000	2,519
17,030	0,00101	1,182	0,903	10,000	2,181
14,930	0,00114	1,184	0,748	9,942	1,953

F_1	F_2	h	l	t	b
12,830	0,00133	1,136	0,782	10,000	1,648
10,770	0,00163	1,182	0,787	10,000	1,343
9,213	0,00195	0,798	1,246	9,966	1,140
8,348	0,00223	0,739	1,498	9,937	1,005
7,600	0,00250	0,454	2,471	9,948	0,893
6,672	0,00293	0,393	2,898	9,926	0,765
5,909	0,00349	0,413	3,004	9,810	0,666
5,209	0,00385	0,393	2,927	9,933	0,582
4,707	0,00434	0,396	3,091	9,981	0,508
4,112	0,00511	0,317	3,662	9,926	0,439
3,742	0,00634	0,393	2,888	9,304	0,430

Tabela 4.19: Resultados Viga Soldada para SIAMEP 40

F_1	F_2	h	l	t	b
36,42885	0,00044	5,000	0,523	10,000	1,609
35,06162	0,00046	4,780	0,611	10,000	1,472
33,25849	0,00048	4,532	0,698	10,000	1,255
32,02604	0,00050	4,370	0,523	10,000	1,609
30,80955	0,00052	4,182	0,727	10,000	1,211
29,46744	0,00055	3,994	0,714	10,000	1,230
28,56445	0,00057	3,864	0,756	10,000	1,166
27,28053	0,00059	3,690	0,533	10,000	1,585
25,41610	0,00064	3,423	0,712	10,000	1,230
24,15962	0,00068	3,245	0,698	10,000	1,254
22,76497	0,00072	3,033	0,904	10,000	1,009
21,70182	0,00076	2,898	0,698	10,000	1,252
20,44348	0,00081	2,720	0,661	10,000	1,312
18,91511	0,00088	2,493	0,609	10,000	1,438
17,52933	0,00095	2,300	0,956	10,000	0,965
16,24527	0,00104	2,131	0,655	9,964	1,329
14,97702	0,00113	1,950	0,856	10,000	1,048
13,87583	0,00123	1,789	1,164	10,000	0,800
12,97179	0,00132	1,664	0,733	10,000	1,205
11,15906	0,00155	1,416	0,892	10,000	1,013
10,02171	0,00176	1,248	0,714	10,000	1,228
9,05623	0,00195	1,127	1,180	10,000	0,795
7,62437	0,00236	0,930	1,150	10,000	0,814
6,35324	0,00289	0,760	1,365	10,000	0,698
5,49804	0,00344	0,639	2,173	10,000	0,468
4,87345	0,00391	0,562	1,813	9,994	0,547
4,54656	0,00428	0,516	2,224	9,980	0,463
4,18996	0,00468	0,469	2,299	10,000	0,450
3,75003	0,00535	0,412	2,532	9,980	0,412
3,50977	0,00590	0,372	3,062	10,000	0,368
3,41766	0,00624	0,368	3,009	9,848	0,367
3,15566	0,00682	0,326	3,809	9,960	0,298
2,92950	0,00746	0,300	3,809	9,936	0,298
2,78787	0,00788	0,278	4,183	10,000	0,276
2,67467	0,00870	0,265	4,542	9,839	0,264
2,63411	0,00952	0,265	4,672	9,547	0,265
2,62824	0,01143	0,276	4,857	8,858	0,276
2,61469	0,01263	0,279	5,068	8,536	0,276

F_1	F_2	h	l	t	b
2,57830	0,01348	0,277	5,189	8,377	0,276
2,56548	0,01396	0,276	5,264	8,286	0,276

Pela análise das figuras pode-se perceber mais uma vez que o algoritmo desenvolvido foi capaz de obter resultados melhores do que os que haviam sido obtidos por CASTRO, R. E. (2001) e DEB *et al.* (2000b). Nota-se um fronte mais homogêneo e com melhor convergência aos ótimos.

4.4.6 Conclusões sobre os resultados das Funções com restrição

Conforme foi possível notar pelos resultados apresentados nos testes, o algoritmo SIAMEP foi capaz de visualmente superar os resultados dos algoritmos utilizados nos estudos para a comparação. Isto indica que o algoritmo se mostra competitivo e com características que permitem sua aplicação em problemas de engenharia, conforme pode ser percebido nos testes da Viga bi-apoiada, treliça de duas barras e viga soldada.

4.5 COMENTÁRIOS

Os algoritmos evolutivos, de uma maneira geral, tem se mostrado bastante capazes de lidar com problemas de otimização multi-objetivos. E foi isso o que se verificou até agora neste trabalho para a resolução de problemas com funções matemáticas com e sem restrição, e com problemas de engenharia. Pode-se destacar a dificuldade encontrada para a resolução da função ZDT4 que possui muitos frontes de pareto sendo ótimos locais. Além disso, destaca-se a dificuldade em se trabalhar com funções com grandes variações de ordem de grandezas nos parâmetros, tornando a busca mais lenta e dificultosa. Outro destaque é que mesmo utilizando-se da abordagem mais simples para o tratamento de restrições, o algoritmo foi capaz de apresentar bons resultados em todos os testes realizados com funções com restrição.

Utilizando o conceito de distinção da solução em função da camada em que se encontra a solução não-dominada, da distribuição por distância de agrupamento, e da força (*strength*) de cada solução, foi possível desenvolver um algoritmo simples, capaz de lidar com a otimização multi-objetivo de forma razoavelmente eficiente. A ferramenta de otimização desenvolvida ainda pode ser melhorada, mas pode-se considerar que no momento encontra-se apta para o desenvolvimento da otimização de rota de dutos submarinos.

Capítulo 5 Otimização de Rota de Dutos Submarinos

Equation Chapter (Next) Section 1Neste capítulo serão tratados os aspectos relacionados à otimização de rota de dutos submarinos, a parametrização da solução e desenvolvimento da aplicação.

5.1 Representação Geométrica

Todo o problema, para ser otimizado por algoritmos, deverá ser representado matematicamente. Para a otimização de rota de dutos não será diferente. Inicialmente consideremos o objeto da otimização: *a rota*. A rota deverá ser executada sobre um fundo (solo marinho) por se tratar de uma rota de duto submarino. A rota deverá ligar um ponto inicial (*A*) a um ponto final (*B*) realizando uma trajetória que seja capaz de desenvolver trechos de reta e trechos de curva para desviar de obstáculos e melhor adaptar-se à batimetria no local.

Pode-se dizer que o traçado de uma rota de duto se assemelha um pouco ao traçado de uma rodovia, é claro, levando-se em consideração outros critérios técnicos, mas a representação pode ser bem parecida.

No projeto geométrico de rodovias temos, por exemplo, o seguinte traçado (ver Figura 5.1) com seus respectivos elementos.

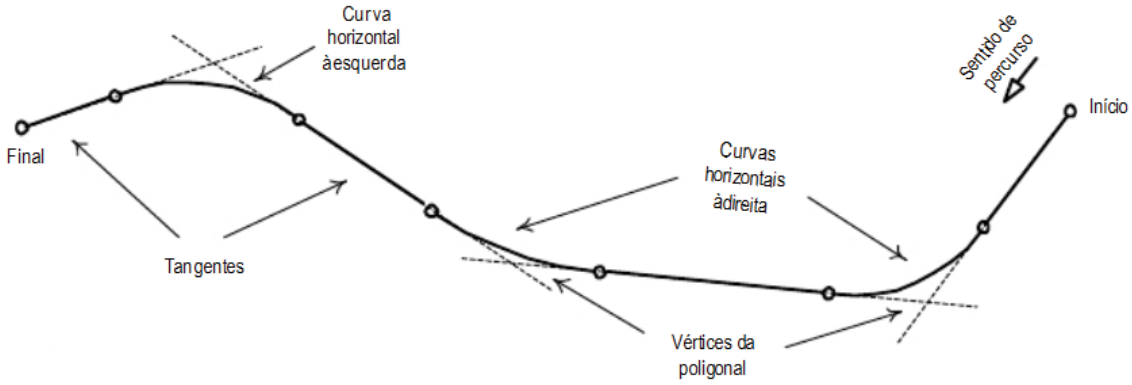


Figura 5.1: Elementos do eixo de uma rodovia (LEE, 2000).

Nota-se que existe uma grande semelhança na representação, bastando substituir o eixo da rodovia pelo eixo de um duto submarino para termos uma rota de duto visualmente traçada. No entanto, para uma otimização por meio de algoritmo bio-inspirado, somente isso não é suficiente, é necessário representar a rota matematicamente. Para representar a rota matematicamente, alguns pontos são destacados na curva (ver Figura 5.2), de forma que algumas características sejam

calculadas e uma curva possa ser representada computacionalmente. Estes elementos são basicamente os mesmo de uma rodovia.

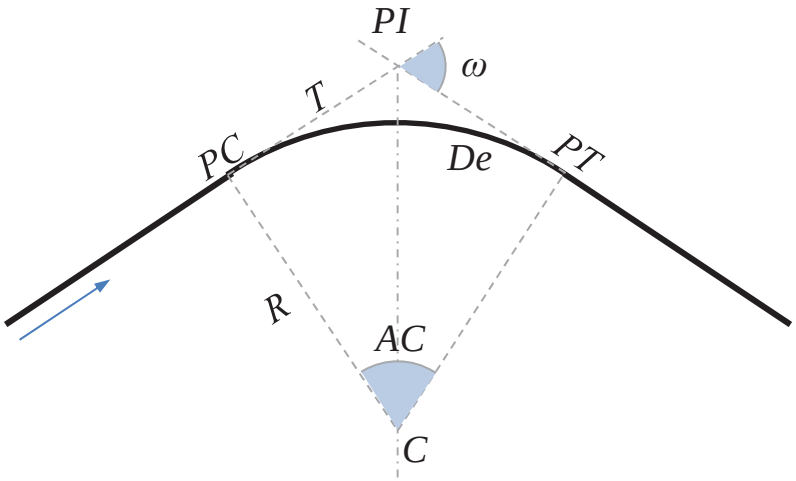


Figura 5.2: Curva circular simples.

A saber:

PI = ponto de interseção;

PC = ponto de curva;

PT = ponto de tangente;

ω = ângulo de deflexão - *não será usado neste trabalho*;

AC = ângulo central;

T = tangente externa ou exterior (distância entre PC e PI, ou PI e PT);

De = desenvolvimento (comprimento do arco desenvolvido entre PC e PT) - *não será usado neste trabalho*;

R = raio da curva;

C = coordenadas do centro da curva;

Com estas informações têm-se a definição de curvas no espaço, sem conexão com o resto do traçado da rota. Assim, é necessário mais uma característica para amarrar os pontos de forma que eles façam uma rota saindo de um ponto inicial (*A*) em direção a um ponto final (*B*) desenvolvendo o menor caminho em trechos de reta e curva. Então será feito o seguinte:

- Ligam-se os pontos *A* e *B* por uma reta;
- Sobre esta reta, e de maneira homogeneamente espaçada, são colocados pontos, chamados de *pontos base,* representados por pb_{ib}, onde ib é um

$ib = 1, 2, ..., N_{pb}$, e N_{pb} é o número de pontos base (ver Figura 5.3);

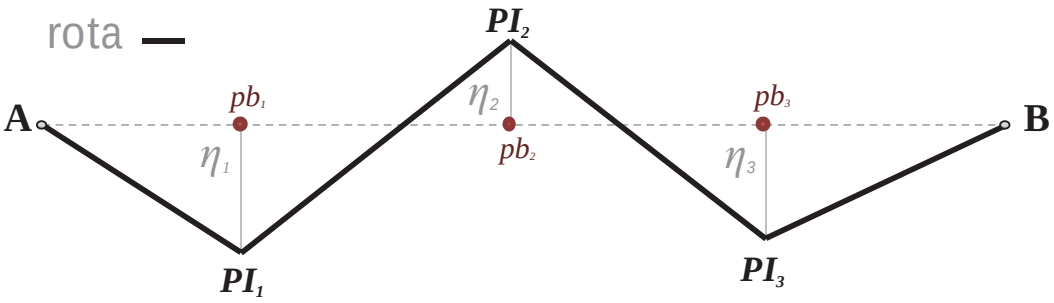


Figura 5.3: Representação da rota base

- Em seguida são gerados afastamentos, representados por η_{ib}, a estes pontos sobre a reta *AB*, fazendo com que a rota fique parecida com um polígono aberto composta por uma série de vértices (ver Figura 5.3);
- Para permitir maior flexibilidade quanto ao posicionamento dos pontos de interseção ($PI's$), estes afastamentos serão utilizados em conjunto com valores gerados de *ângulo com o eixo X* (θ_{ib}) no sentido anti-horário. Isto faz com que os afastamentos tenham diferentes direções (ver Figura 5.4) e os $PI's$ possam ser localizados de forma a contornar os obstáculos na rota gerada;
- Com isso, obtêm-se a coordenada do PI de cada ponto base (pb_{ib});
- A última etapa é ligar os $PI's$ por linhas retas e fazer a inserção de curvas para cada um dos $PI's$ (ligando também os $PI's$ às extremidades *A* e *B*).

Desta forma, a representação poderá ser visualizada na Figura 5.4 a seguir.

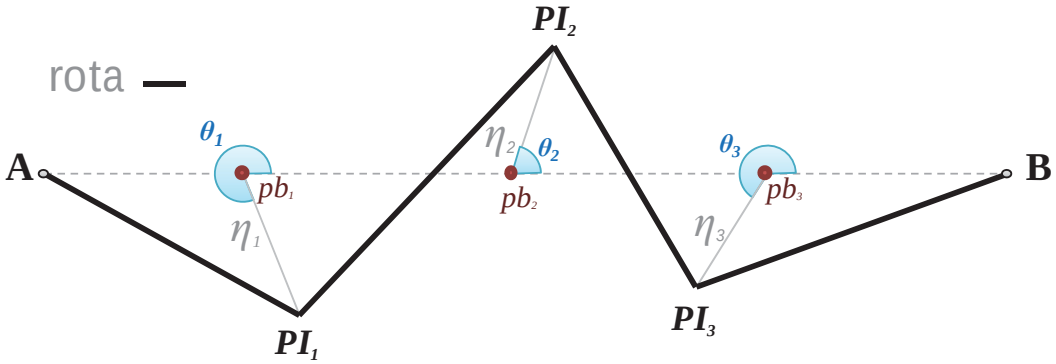


Figura 5.4: Representação de $PI's$ de uma rota, ainda sem curvas.

As curvas deverão ser inseridas em seguida, em cada um dos $PI's$ levando-se em consideração as informações apresentadas no começo deste item. Como ilustrado na Figura 5.5, na representação geométrica estabelecida para a implementação do *OtimRota* as curvas serão definidas como arcos de círculo, e a cada curva estarão associados os seguintes parâmetros:

- Um valor para o raio de curvatura R;
- Coordenadas de um ponto de interseção PI.

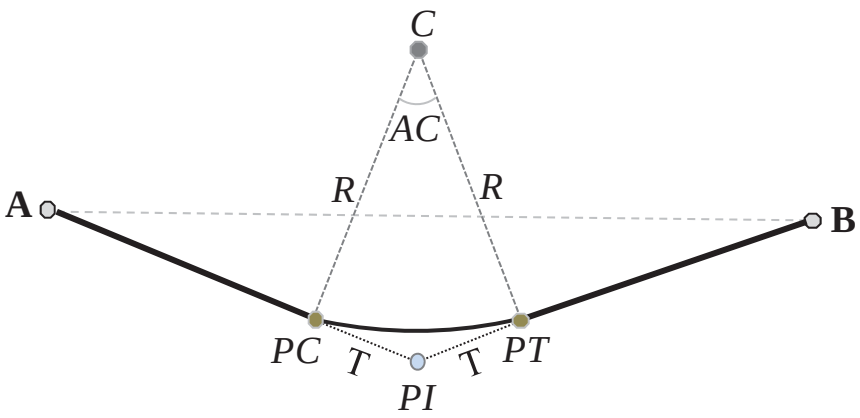


Figura 5.5: Representação de trechos em curva

A seguir será demonstrado que o conjunto de retas e curvas que descrevem a rota entre os pontos *A* e *B* fica completamente caracterizada por esses parâmetros associados a cada PI: suas coordenadas, e o raio de curvatura R do trecho em curva. Mais adiante serão demonstradas as expressões para, em função desses parâmetros principais, calcular os demais parâmetros secundários que caracterizam os trechos retos e curvos da rota, indicados na Figura 5.5, incluindo:

- As coordenadas dos pontos de curva PC e ponto de tangente PT;
- O comprimento da reta tangente T que liga os pontos (PC ou PT) ao PI;
- As coordenadas do centro do setor circular C, e
- O ângulo de abertura AC do setor circular.

Conforme exposto anteriormente, o posicionamento do conjunto de $PI's$ é associado a pontos base distribuídos sobre a reta base. Na implementação atual do *OtimRota* assume-se uma distribuição homogênea para estes pontos base, como indicado na Figura 5.6 respectivamente para 1, 2, 3 e 4 pontos base.

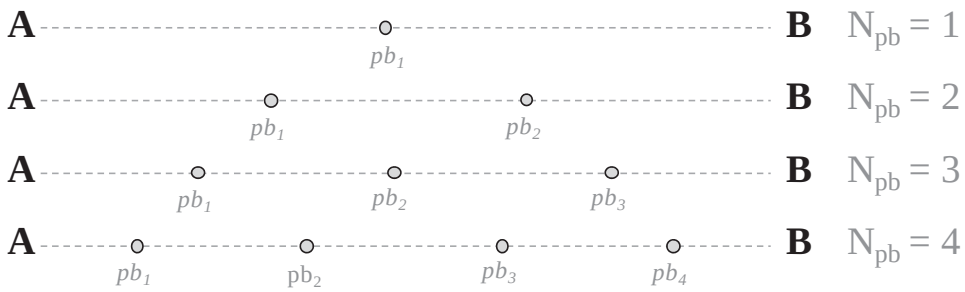


Figura 5.6: Distribuição de pontos base sobre a reta *AB*.

A partir de cada um destes pontos base sobre a reta $\overline{AB}$, a posição de cada ponto PI é definida pelos seguintes parâmetros, indicados na Figura 5.7:

- Afastamento (η_{ib});
- Ângulo com o eixo X no sentido anti-horário (θ_{ib}) ;

Tomando esses valores de afastamento (η_{ib}) e ângulo (θ_{ib}) e as coordenadas cartesianas do ponto base, definem-se as coordenadas cartesianas de cada PI pelas expressões a seguir:

$$\begin{cases} x_{PI_{ib}} = x_{pb_{ib}} + \eta_{ib} \cdot \cos(\theta_{ib}) \\ y_{PI_{ib}} = y_{pb_{ib}} + \eta_{ib} \cdot sen(\theta_{ib}) \end{cases} \tag{5.1}$$

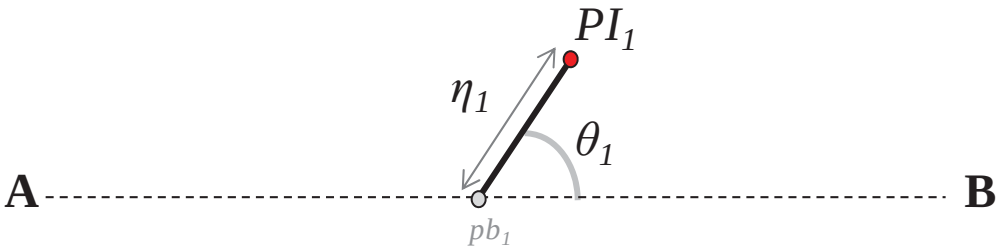


Figura 5.7: Definição de PI

Vamos deduzir agora os procedimentos para o cálculo dos parâmetros secundários que caracterizam os trechos retos e curvos da rota, já relacionados acima e indicados na Figura 5.5. Primeiramente é calculado o ângulo central (AC) parametrizando-se os dois segmentos consecutivos $\left(\overline{PI_{(ib-1)}PI_{(ib)}}\right.$ e $\left.\overline{PI_{(ib)}PI_{(ib+1)}}\right)$ indicados na Figura 5.8, com comprimentos L1 e L2 respectivamente, e empregando a seguinte expressão (5.2):

$$AC = a\cos\left[\frac{u_1 \cdot u_2 + q_1 \cdot q_2}{\sqrt{\left(u_1^2 + q_1^2\right) \cdot \left(u_2^2 + q_2^2\right)}}\right] \tag{5.2}$$

Onde:

$$\begin{cases} u_1 = x_{ib} - x_{ib-1} \\ u_2 = x_{ib+1} - x_{ib} \end{cases} \text{ e } \begin{cases} q_1 = y_{ib} - y_{ib-1} \\ q_2 = y_{ib+1} - y_{ib} \end{cases} \tag{5.3}$$

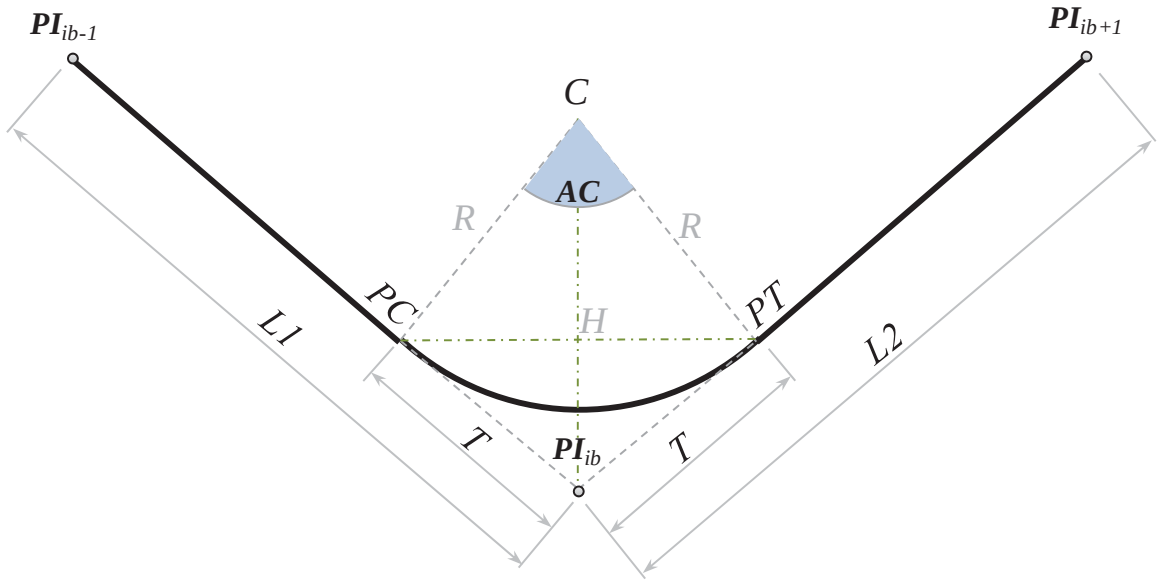


Figura 5.8: Cálculo dos pontos sobre a curva

Com o ângulo central (AC) calculado, é possível calcular o comprimento da tangente (T):

$$T = R \cdot \tan\left(\frac{AC}{2}\right) \tag{5.4}$$

Assim, é possível entrar nas equações paramétricas dos dois segmentos de reta e calcular as coordenadas *x* e *y* dos pontos *PC* e *PT*.

$$PC \begin{cases} x = x_{ib-1} + u_1 \cdot \left[1 - \left(T \diagup L1\right)\right] \\ y = y_{ib-1} + q_1 \cdot \left[1 - \left(T \diagup L1\right)\right] \end{cases} \tag{5.5}$$

e,

$$PT\begin{cases} x = x_{ib} + u_2 \cdot \left[\left(T/L2\right)\right] \\ y = y_{ib} + q_2 \cdot \left[\left(T/L2\right)\right] \end{cases} \tag{5.6}$$

Para o cálculo do centro C é necessário definir mais um ponto. Na Figura 5.8, seja o ponto *H* um ponto que divide a reta $\overline{PC\ PT}$ em dois segmentos de igual tamanho, de modo que *H* é calculado como:

$$H\begin{cases} x = \left(x_{PC_{ib}} + x_{PT_{ib}}\right) \cdot 0,5 \\ y = \left(y_{PC_{ib}} + y_{PT_{ib}}\right) \cdot 0,5 \end{cases} \tag{5.7}$$

Em seguida são calculados os comprimentos $\overline{CPI}$ e $\overline{HPI}$:

$$\overline{CPI} = \sqrt{T^2 + R^2} \tag{5.8}$$

Com o ponto H calculado e PI, parametriza-se a reta $\overline{HPI}$ e calcula-se o comprimento $\overline{HPI}$. Tomando:

$$\begin{cases} u = (x_H - x_{PI}) \\ q = (y_H - y_{PI}) \end{cases} \tag{5.9}$$

Obtém-se:

$$\overline{HPI} = \sqrt{u^2 + q^2} \tag{5.10}$$

Definindo-se agora:

$$s = \frac{\overline{CPI}}{\overline{HPI}} \tag{5.11}$$

Pode-se finalmente determinar as coordenadas do centro do círculo (C) pelas seguintes expressões:

$$C\begin{cases} x_C = x_{PI} + s \cdot u \\ y_C = y_{PI} + s \cdot q \end{cases} \tag{5.12}$$

Com isso, todos os dados necessários para a constituição da representação da curva estão devidamente calculados.

5.2 CODIFICAÇÃO NO ALGORITMO DE OTIMIZAÇÃO

Em virtude dos testes apresentados inicialmente nesta tese com as funções matemáticas, o algoritmo multi-objetivo utilizado será o SIAMEP. Cada anticorpo representa uma solução do problema, ou seja, representa uma rota completa. Cada indivíduo terá como variáveis livres:

- PI ativo ou não;
- Afastamento (η);
- Ângulo com eixo *x* (θ);
- Raio (R).

O número de $PI's$ poderá variar ao longo do processo de otimização, partindo de um número máximo pré-definido. Com isso, dependendo do grau de complexidade do problema, a representação geométrica da rota pode ser simplificada pelo algoritmo, com a desativação de alguns $PI's$, ou seja, considerando que no trecho correspondente a rota é reta.

Com isso cada anticorpo poderá ter mais de um PI, sendo que cada PI terá quatro variáveis associadas a ele.

5.3 MODELAGEM DO PROBLEMA

Conforme dito anteriormente, em BAIOCO (2011), a otimização de rota de dutos foi tratada de forma uni-objetivo, e agora nesta tese o problema será tratado como multi-objetivo. A seguir serão apresentadas as duas diferentes maneiras de tratamento ao problema.

5.3.1 Tratamento Uni-objetivo

O primeiro passo (e um dos mais importantes) na construção de uma ferramenta de otimização deve ser a construção da função objetivo, ou função de avaliação da aptidão de cada solução apresentada. A função objetivo deve refletir a "qualidade" da rota analisada e, para isso, deve levar em consideração todos os aspectos relevantes que dizem respeito à escolha de uma rota.

Para a geração da rota, inicialmente estão sendo considerados aspectos geográficos/topográficos relacionados à batimetria de fundo e aos obstáculos/interferências.

O primeiro e mais importante fator envolvido na avaliação é o comprimento total da rota, pois implica em comprimento de duto. Caso nenhum outro fator fosse levado em consideração, a melhor rota seria a rota mais curta ligando o ponto *A* ao ponto *B*. Isto levaria à solução trivial, a reta $\overline{AB}$. Pode-se então definir como primeiro parâmetro de avaliação da rota a relação entre o seu comprimento e o comprimento da reta $\overline{AB}$:

$$Aptidão = \frac{dist_{AB}}{L_{rota}} \tag{5.13}$$

onde $dist_{AB}$ é o valor do comprimento da reta $\overline{AB}$ e L_{rota} o valor do comprimento da rota gerada no processo de otimização. Assim, uma rota ideal teria *Aptidão* igual a um, e qualquer outra rota real teria valor de *Aptidão* entre zero e um, com isso, minimizando-se a rota tem-se os maiores valores de *Aptidão*.

Em vista de atender as restrições físicas, geométricas e estruturais da trajetória, alguns fatores devem ser levados em consideração, tais como a presença de obstáculos e a declividade do terreno. Estes e outros fatores são tratados através da introdução de penalizações à função objetivo. Uma penalização é aplicada à rota sempre que uma restrição for violada.

Para levar em consideração as penalizações no cálculo da *Aptidão*, considera-se a seguinte expressão:

$$Aptidão = 100 \cdot \frac{dist_{AB}}{L_{Rota} + dist_{AB} \sum_{j=1}^{N_{re}} \left(mk_j \cdot penal_j \right)} \tag{5.14}$$

onde N_{re} representa o número de penalidades, ou restrições; $penal_j$ representa o valor de cada penalidade $j = 1, 2, ..., N_{re}$; mk_j representa uma ponderação relativa atribuída a cada penalidade; $dist_{AB}$ é o valor do comprimento da reta $\overline{AB}$ e L_{rota} o valor do comprimento da rota gerada no processo de otimização. A constante 100 funciona como fator de escala para transformar a faixa de variação de [0,1] para [0,100].

Através do fator de ponderação mk_j (peso) para as diversas penalizações, será possível controlar a influência que cada penalização terá sobre o resultado final, aumentando ou diminuindo a sua importância. É claro que este fator de ponderação representa certa dificuldade uma vez que a sua escolha não é trivial, sendo necessário um especialista.

Evidentemente, para rotas penalizadas o valor da *Aptidão* será sempre menor do que o valor para rota não penalizada. Observa-se também que a equação (5.14) iguala-se à equação (5.13) no caso de uma rota não sofrer nenhuma penalização, quando se tem:

$$\sum_{j=1}^{N_{re}} \left(mk_j \cdot penal_j \right) = 0$$

Além desta formulação, foi desenvolvida outra formulação que permite tratar o problema de otimização do peso de lastro mínimo requerido para manter o duto estável, a qual incorpora uma nova parcela (peso requerido de lastro), e onde o objetivo final passa a ser o custo do duto e não mais diretamente o comprimento do duto. A formulação é a seguinte:

$$Aptidão = 100 \cdot \frac{k_{duto} dist_{AB}}{k_{duto} L_{Rota} + k_{lastro} W_{reqTotal} + dist_{AB} \sum_{j=1}^{N_{re}} \left(mk_j \cdot penal_j \right)} \tag{5.15}$$

onde, k_{duto} é a ponderação relativa ao comprimento do duto, em unidade de custo por unidade de comprimento (\$/m); k_{lastro} é a ponderação relativa ao lastro, em unidade de custo por unidade de peso (\$/N); $W_{reqTotal}$ representa o peso total de lastro da rota analisada, que é calculado com a seguinte equação:

$$W_{reqTotal} = \sum_{iel=1}^{N_{elem}} \left(W_{req}(iel) \cdot L_{elem}(iel) \right) \tag{5.16}$$

onde, N_{elem} é o número de elementos que compõe a rota; $W_{req}(iel)$ é o peso requerido para estabilizar determinado elemento iel da rota; $L_{elem}(iel)$ é o comprimento de cada elemento iel da rota analisada.

Esta formulação foi usada em BAIOCO (2011) e será o objeto das comparações realizadas com a formulação multi-objetivo.

5.3.2 Tratamento Multi-Objetivo

O tratamento multi-objetivo deverá comportar duas funções objetivos independentes: uma para computar o comprimento da rota gerada, e outra para computar o peso total de lastro utilizado para que a rota se torne estável segundo os critérios de estabilidade apresentados e definidos em BAIOCO (2011).

Assim, as funções objetivo serão dadas pelas equações abaixo:

$$\begin{aligned} f_1 &= L_{rota} \\ f_2 &= W_{reqTotal} \end{aligned} \tag{5.17}$$

onde L_{rota} é o comprimento da rota gerada, calculado pela simples soma dos segmentos de reta e curva que compõem a rota; e $W_{reqTotal}$ é o peso de lastro requerido para o duto proposto ficar estável na trajetória analisada, sendo calculado por um processo iterativo proposto por BAIOCO (2011); sendo o comprimento expresso em *m* (metros) e o peso requerido expresso em *N* (*newtons*).

O processo iterativo do cálculo do peso de lastro requerido para o duto ficar estável pode ser apresentado de forma resumida como se segue:

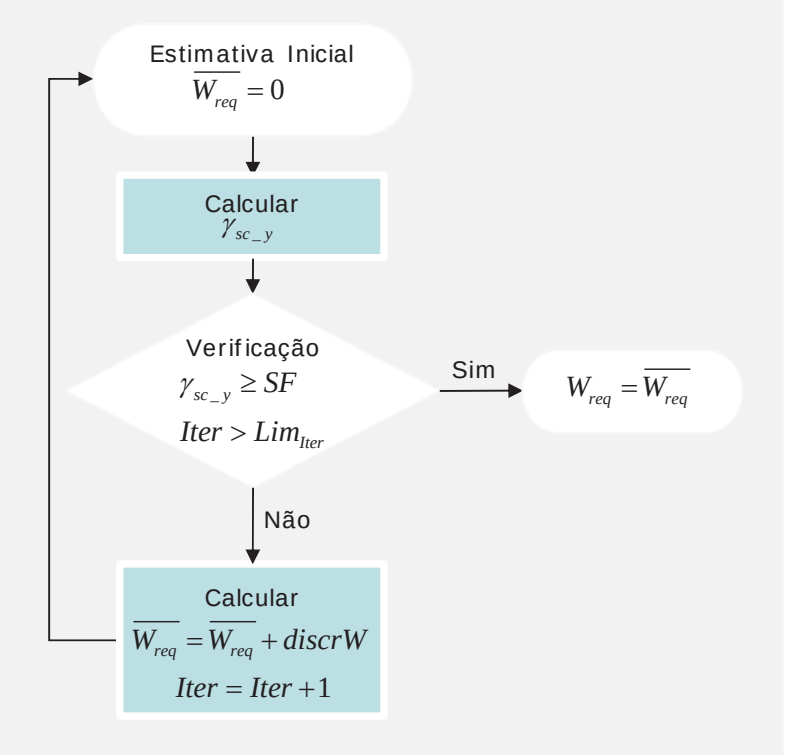


Figura 5.9: Cálculo do peso de lastro requerido para o duto ficar estável

onde $\overline{W_{req}}$ é o peso de lastro do duto; γ_{sc_y} é um fator de segurança calculado para o duto; SF é o fator de segurança mínimo para o duto (neste trabalho, igual a 1,1); Lim_{Iter} é o limite de iterações para o cálculo do peso requerido (é um valor pré-definido); $Iter$ é o número da iteração atual; $discrW$ é o incremento aplicado ao peso de lastro no duto para a conseguinte verificação de estabilidade (um valor pré-definido).

Este procedimento aumenta gradativamente o peso de lastro do duto ($\overline{W_{req}}$) até que este possua um fator de segurança (γ_{sc_y}) que atenda ao fator de segurança mínimo estipulado (SF). O cálculo do fator de segurança é descrito em sua totalidade em BAIOCO (2011).

De acordo com a norma DNV-RP-F109 (2007), três metodologias podem ser consideradas na verificação de estabilidade de dutos (BAIOCO, 2011):

1. "*Absolute Lateral Static Stability*" ou Estabilidade Estática Absoluta - tem por objetivo garantir a Estabilidade Estática Absoluta, ou seja, garantir que as cargas hidrodinâmicas atuando no duto sejam menores do que a resistência do solo;
2. "*Generalized Lateral Stability Method*" ou Estabilidade Generalizada - objetiva garantir o "*No Break-Out*", para um duto praticamente estável,

3. "*Dynamic Lateral Stability Analysis*" ou Análise Dinâmica de Estabilidade - permite deslocamentos acumulados, de modo que o duto pode sair e entrar em sua "cavidade", e a resistência do solo é dependente do histórico de deslocamentos do duto.

O cálculo de peso total requerido de lastro desta tese atende ao critério de estabilidade absoluto definido em DNV-RP-F109 (2007) de forma a garantir que as cargas hidrodinâmicas atuando no duto sejam menores do que a resistência do solo. O método de establidade absoluto está baseado no equilíbrio de forças (ver Figura 5.10). Adiconalmente o duto deverá atender a algumas premissas: **(i)** não são tolerados deslocamentos do duto; **(ii)** a resultante horizontal das cargas hidrodinâmicas atuantes no dute deverá ser menor do que a resistência do solo, assim como a carga vertical de sustentação deverá ser menor do que o peso submerso do duto.

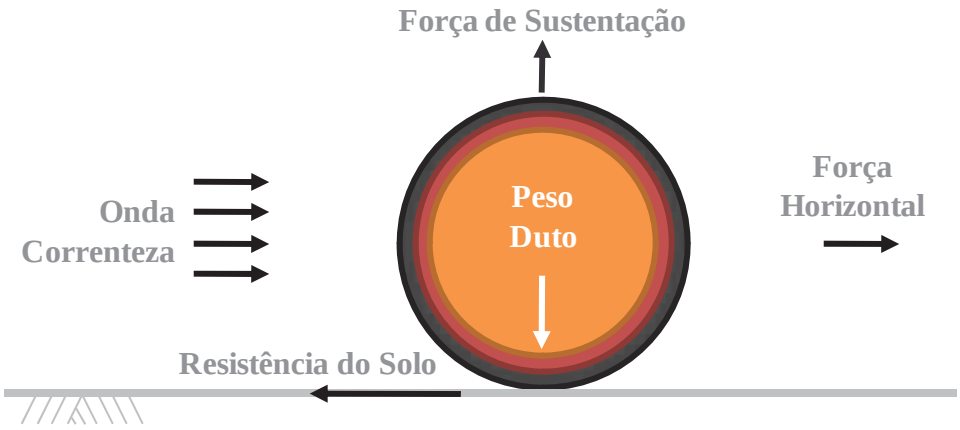


Figura 5.10: Forças atuantes no duto, modificado de BAIOCO (2011)

Segundo a DNV-RP-F109 (2007), é necessário que duas condições sejam atendidas para que um duto seja considerado estável. Estas condições estabelecem um fator de segurança γ_{sc} que garantem a estabilidade lateral γ_{sc_y} e vertical γ_{sc_z}.

A primeira condição está relacionada ao equilíbrio estático lateral, medido por um fator de segurança γ_{sc_y}, onde se verifica o equilíbrio entre as componentes horizontais de forças hidrodinâmicas e a resistência do solo.

A segunda condição está relacionada ao equilíbrio estático vertical, medido por um fator de segurança γ_{sc_z}, onde as cargas do peso submerso do duto e a força de sustentação são analizadas.

5.4 RESTRIÇÕES

Os itens a seguir descrevem as penalizações consideradas no processo de otimização.

5.4.1 Restrição 1 - Cruzamento sobre si mesmo

Durante o processo de otimização, quando as variáveis livres associadas à representação geométrica podem assumir diferentes valores dentro da faixa pré-definida, e dependendo do número máximo de pontos de Interseção escolhido, dos afastamentos e dos ângulos gerados, pode ocorrer que a trajetória do duto cruze sobre si mesmo.

Sendo assim, estabeleceu-se uma estratégia de penalizar soluções que possuam auto-cruzamento. A verificação de auto-cruzamento é feita por um algoritmo de verificação de cruzamento de segmentos de retas, percorrendo cada segmento de duto contra todos os demais, excluindo os que já foram verificados no início do ciclo. O número de interseções é contado e a trajetória é penalizada pela equação (5.18) abaixo.

$$penal_1 = \exp(ncruz_1) - 1 \tag{5.18}$$

onde $ncruz_1$ representa o número encontrado de interseções sobre si mesmo. Assim, apesar de qualquer numero de cruzamento gerar uma penalização, um duto com menos cruzamentos será menos penalizado do que um com mais cruzamentos.

Graficamente a função de restrição é como apresentada na Figura 5.11.

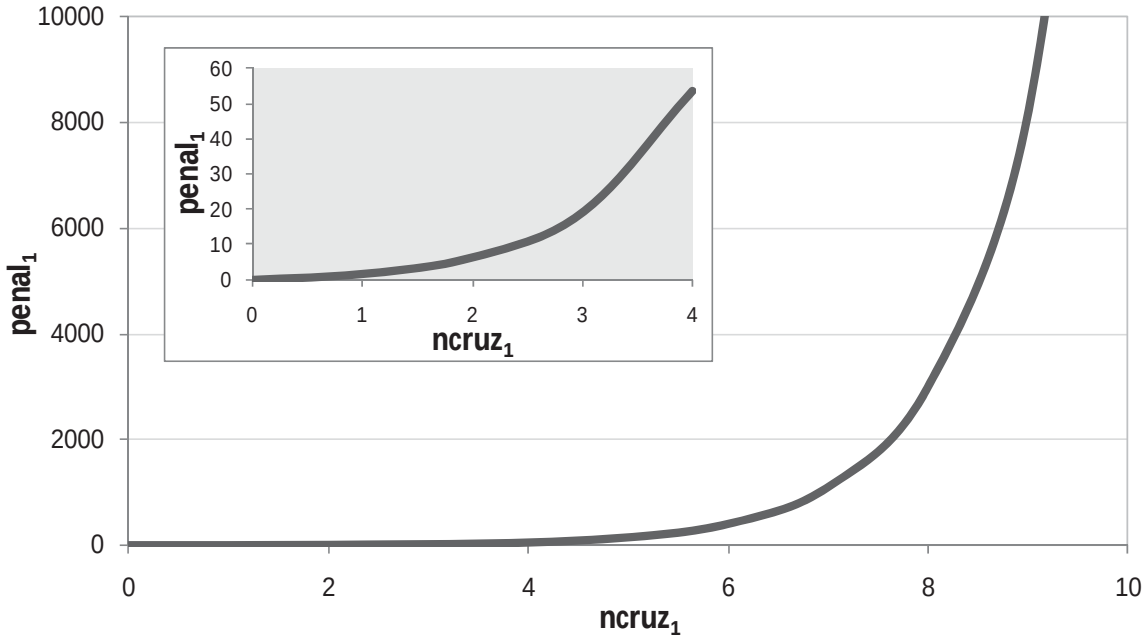


Figura 5.11: Restrição de cruzamento sobre si mesmo

5.4.2 Restrição 2 - Cruzamento de obstáculos

Quando uma rota é gerada, esta poderá estar passando sobre obstáculos pré-definidos na batimetria. Estes obstáculos podem ser: regiões de corais, outros elementos de infra-estrutura, regiões de cânions, etc. Os obstáculos considerados na modelagem podem ser dos quatro tipos apresentados a seguir:

i. Linha – definida por nós com coordenadas x, y e z , e segmentos conectando sucessivamente os nós;

ii. Polígono – região fechada definida por vértices com coordenadas x, y e z, e segmentos que ao final conectam ao primeiro vértice;

iii. Ponto – na verdade uma região esférica definida por um centro com coordenadas x, y e z e um raio;

iv. Cilindro – região definida por um círculo com centro em coordenadas x, y e z, raio e altura, orientado sempre na direção do eixo Z global.

Na geração de uma rota, a superposição do duto com obstáculos é identificada realizando verificações de cruzamento entre os segmentos de reta que compõem o duto contra todos os segmentos que definem os obstáculos.

A cada obstáculo poderá ser atribuído um grau de importância de cruzamento:

- **Nível 0** – Ilustrativo: Caso o obstáculo seja apenas ilustrativo, podendo ser superposto sem nenhuma restrição (representado pela cor cinza) $< imp = 0,0 >$;
- **Nível 1** – Fácil: Admite-se haver superposição entre duto e obstáculo, e esta é relativamente simples (representado pela cor verde) $< imp = 0,2 >$.
- **Nível 2** – Normal: Superposição entre duto e obstáculo condicionada (representado pela cor amarelo) $< imp = 0,5 >$.
- **Nível 3** – Difícil: Superposição requer intervenção cara e difícil (representado pela cor vermelho) $< imp = 1,0 >$.
- **Nível 4** – Impossível: Não se admite superposição (representado pela cor preta) $< imp = 100,0 >$.

A cada um destes níveis de dificuldades foi atribuído um peso diferente para a respectiva penalidade (imp), representando a importância deste obstáculo, fazendo com que o algoritmo aceite soluções que cruzam os obstáculos menos penalizados. O número de interseções é contado para cada tipo de obstáculo e a penalidade é obtida pela equação (5.19).

$$penal_2 = \exp\left(\sum_{ic=0}^{ncruz_2} (imp) \right) - 1 \tag{5.19}$$

onde $ncruz_2$ representa o número encontrado de interseções com obstáculos, imp é o valor da importância do obstáculo; e ic é o contador do número de interseções. O valor de imp deverá ser pré-definido por um especialista atribuindo uma quantidade numérica para cada nível de dificuldade.

Como descrito anteriormente, o cruzamento de obstáculos é feito por um algoritmo de verificação de cruzamento de segmentos de retas, percorrendo cada segmento da rota contra cada segmento de cada obstáculo, excluindo os que já foram verificados no início do ciclo. Entretanto, esta verificação torna-se computacionalmente onerosa em casos com elevado número de obstáculos. Com o intuito de minimizar este problema foi implementado o sistema de *Bounding Box* para os obstáculos do tipo linha.

O *Bounding Box* é um termo da computação gráfica usado para definir uma região de influência. Neste caso, consiste em um retângulo que envolve os obstáculos do tipo linha nos modelos a serem otimizados. O objetivo é simplificar a verificação de cruzamento da rota com estes tipos de obstáculos e assim diminuir o tempo de análise no processo de otimização (BAIOCO, 2011). A formulação do modelo *Bounding Box* e os estudos de ganho computacional estão apresentados em JACOB *et. al* (2011).

5.4.3 Restrição 3 - Comprimento mínimo entre duas curvas

Para atender a requerimentos da operação de lançamento de duto, é necessário estabelecer uma distância mínima entre duas curvas consecutivas, para que não haja mudança de sentido sem haver espaço para a manobra da embarcação de lançamento do duto.

Por isso, foi estabelecida uma restrição em termos da distância mínima entre curvas. Quando este valor é violado a trajetória é penalizada, pela expressão (5.20) indicada a seguir.

$$penal_3 = \begin{cases} \left(\dfrac{L_{\min Curvas} - L_{medidoCurvas}}{L_{\min Curvas}} \right), \; se \; L_{medidoCurvas} < L_{\min Curvas} \\ 0, \; se \; L_{medidoCurvas} \geq L_{\min Curvas} \end{cases} \tag{5.20}$$

onde $L_{\min Curvas}$ o comprimento mínimo entre curvas (valor pré-definido); e $L_{medidoCurvas}$ é o comprimento medido entre duas curvas consecutivas.

Nota-se que quanto menor a distância entre curvas, mais penalizada será a solução; quando este valor superar o mínimo comprimento estabelecido ($L_{\min Curvas}$), a solução não será penalizada.

5.4.4 Restrição 4 - Comprimento reto mínimo de saída (ponto A)

Ainda para atender a requisitos da operação de lançamento de duto, o duto precisa sair em linha reta do ponto de origem (*A*). Sendo assim, foi estabelecida uma restrição para favorecer configurações de rota contendo um trecho reto no início. Se o comprimento do trecho reto de início for menor do que o valor mínimo estabelecido, a solução será penalizada pela seguinte expressão:

$$penal_4 = \begin{cases} \left(\dfrac{L_{\min A_cur} - L_{medidoA_cur}}{L_{\min A_cur}} \right), \; se \; L_{medidoA_cur} < L_{\min A_cur} \\ 0, \quad se \; L_{medidoA_cur} \geq L_{\min A_cur} \end{cases} \tag{5.21}$$

onde $L_{\min A_cur}$ representa o comprimento mínimo (pré-definido) entre o ponto *A* e o início da primeira curva; e, $L_{medidoA_cur}$ representa a distância medida entre o ponto *A* e o início da primeira curva.

5.4.5 Restrição 5 - Comprimento reto mínimo de chegada (ponto B)

De modo semelhante, estabelece-se a seguinte restrição para favorecer configurações de rota contendo um trecho reto no final:

$$penal_5 = \begin{cases} \left(\dfrac{L_{\min Cur_B} - L_{medidoCur_B}}{L_{\min Cur_B}} \right), \; se \; L_{medidoCur_B} < L_{\min Cur_B} \\ 0, \; se \; L_{medidoCur_B} \geq L_{\min Cur_B} \end{cases} \tag{5.22}$$

onde $L_{\min Cur_B}$ representa o comprimento mínimo (pré-definido) entre o final da última curva e o ponto *B*; e, $L_{medidoCur_B}$ representa a distância medida entre o final da última curva e o ponto *B*.

5.4.6 Restrição 6 - Limites da batimetria

As rotas geradas poderiam, dependendo dos pontos escolhidos como *A* e *B*, e mesmo dependendo dos limites de cada variável livre, extrapolar a região do limite da batimetria, ou seja, cair fora da região onde há dados definidos para fundo. As rotas

Procedimento 3: Cálculo da *penalidade para o limite da batimetria*

01: $penal_6 = 0$ [inicializando o valor da variável]

02: *ObstaculoBordaBatimetria* [subrotina para criar obstáculo nos limites da batimetria]

03: **Para** cada nó ino na malha do duto, **faça**

04: **Se** *Segmento* intercepta o *Obstáculo* criado, **então** [*Segmento* entre 2 pontos consecutivos da malha do duto]

05: $penal_6 += 1$ [acumulando valor na variável]

06: **Fim** do **Se**

07: **Fim** do **Para**

onde ino representa o contador do número de nós na malha do duto. Desta forma obtém-se o número de vezes em que a rota gerada cruzou os limites da batimetria e este valor será o utilizado para a penalização juntamente com a ponderação mk_6.

5.4.7 Restrição 7 - Declividade longitudinal;

As restrições definidas nos itens anteriores tratavam de aspectos geométricos, que dependiam somente da geometria da rota e dos obstáculos eventualmente presentes. Uma vez dispondo-se de informações de batimetria, pode-se também levar em consideração a topografia do terreno sobre o qual a rota está sendo definida, e com isso introduzir uma restrição em termos da declividade do terreno.

Para isto, o *OtimRota* foi implementado de forma a poder importar e utilizar os dados de batimetria utilizados pelo programa SITUA/PROSIM. Estes dados descrevem a batimetria de uma região de duas formas: Através de linhas isobatimétricas e através de uma superfície interpolada sobre estas linhas.

Para a verificação dos critérios anteriores de cruzamento sobre si mesmo e com obstáculos, a descrição da rota através das retas e curvas descrita no item 5.1 é adequada. No entanto, para a avaliação da batimetria é necessário uma definição mais detalha da rota, pois a variação da superfície do fundo pode ser bastante grande entre os pontos de definição da rota. Para isto foi gerada uma malha mais refinada sobre a rota com a interpolação da cota Z em cada ponto da malha. O grau de refinamento, ou seja, a distância entre os elementos de malha é pré-definido.

Esta interpolação foi feita inicialmente utilizando as informações das curvas isobatimétricas. No entanto, a quantidade de informações disponíveis é muito grande, o que leva a um grande gasto de tempo computacional para o cálculo da cota Z de cada

ponto da malha. Por isso, optou-se pela interpolação sobre a malha quadrangular da superfície do fundo, fornecida pelo programa SITUA. Esta malha possui uma menor densidade de informações, levando a tempos computacionais bem menores.

Para verificar esta opção de uso de malha de superfície foram feitos diversos testes comparativos; verificou-se que o ganho de precisão com o uso de curvas batimétricas ao invés da malha não é tão significativo a ponto de justificar o alto custo computacional.

É desejável que o duto seja traçado limitando a declividade máxima. O processo de penalização em função da declividade se dá da seguinte maneira:

Primeiro é estabelecido um valor limite para a maior declividade longitudinal aceita sem penalização, $decliv_{\max}$, em seguida é executado o procedimento descrito a seguir, onde: $Decliv_i$ representa a declividade longitudinal calculada para dois pontos consecutivos na malha do duto; $N_{nós}$ representa o número de nós da malha de duto; $penal_7$ é o valor da penalidade por declividade; e ino o contador do número de nós na malha do duto.

Procedimento 4: Cálculo da *penalidade por declividade longitudinal*

01: $penal_7 = 0$ [inicializando o valor da variável]

02: **Para** cada nó ino na malha do duto, **faça**

03: **Se** $Decliv_{ino} > decliv_{\max}$, **então** [Declividade longitudinal de *Segmento* entre 2 pontos consecutivos da malha do duto]

04: $penal_7 += \exp(Decliv_{ino} - decliv_{max})$ [acumulando valor na variável]

05: **Fim** do **Se**

06: **Fim** do **Para**

07: $penal_7 += penal_7 / N_{nós}$

Percorre-se a malha do duto verificando se cada segmento excedeu ou não o limite pré-estabelecido de declividade; se excedeu o limite, o valor da penalização dado pela Eq. (5.23) é acumulado na variável ($penal_7$). Ao final, divide-se o total acumulado em $penal_7$ pelo número de nós da malha do duto (N_{nos}). Isto é feito para se levar em consideração o seguinte: seja um duto representado por 100 segmentos; um duto que tenha dois segmentos apenas que não atendam à declividade limite é melhor do que um que não atende em 25 segmentos o mesmo limite de declividade. Então como em ambos o valor será dividido pela mesma quantia, o total da penalidade no primeiro caso será

menor do que no segundo caso. E o valor estará sendo medido por uma equação exponencial que nos permite penalizar mais intensamente valores que mais se afastam do limite pré-estabelecido para a declividade.

$$penal_7 += \exp\left(Decliv_{ino} - decliv_{\max}\right) \tag{5.23}$$

5.4.8 Restrição 8 - Raio de curvatura mínimo

Por limitações físicas do duto, e também por questões relativas à operação de instalação, o duto deverá respeitar um raio de curvatura mínimo, definido conforme se segue (FERNANDES *et al*., 2008):

O raio mínimo de curvatura requerido $R_{\min}$ é dado pela seguinte expressão:

$$R_{\min} = \frac{F_{Hor}}{\mu_{solo} \cdot W_{sub}} \tag{5.24}$$

onde F_{Hor} representa a tensão residual no fundo – valor pré-definido igual à tração de lançamento do duto; μ_{solo} o coeficiente de fricção (ou atrito) do solo marinho – valor pré-definido função do solo em que será instalado o duto; e W_{sub} o peso submerso do duto – calculado pela Eq. (5.25) abaixo.

$$W_{sub} = W_T - W_e \tag{5.25}$$

onde W_T representa o peso total do duto – obtido a partir das áreas e pesos de cada material do duto (ver Figura 5.12); W_e é o empuxo do duto – calculado pela área da seção transversal do duto A_T multiplicado pelo peso específico da água μ_{ag}, ver Eq. (5.26) abaixo:

$$W_{sub} = A_T \cdot \mu_{ag} \tag{5.26}$$

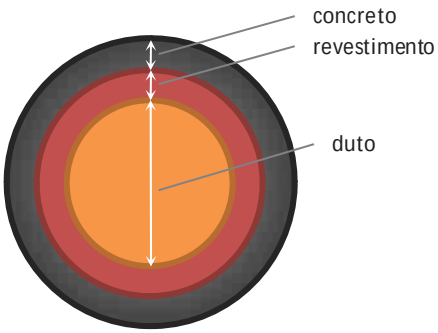


Figura 5.12: Seção de Duto Esquemática

A penalidade será calculada da seguinte maneira:

Procedimento 5: Cálculo da *penalidade para Raio mínimo*

01: $penal_8 = 0$ [inicializando o valor da variável]

02: **Calcular** *Área* e *Peso* dos materiais {*Aço, Revestimento, Concreto*}: [função da espessura e peso específico]

03: **Calcular** *Área* e *Peso* Total do duto

04: **Calcular** *Empuxo* e *Peso Submerso* do duto

05: **Obter** *Raio mínimo* pela expressão (5.24)

06: **Para** cada nó ino na malha do duto, **faça**

07: **Se** $R_{ino} < R_{\min}$, **então** [R_{ino} = raio de curvatura de um nó *ino*]

08: $$penal_8 += \left(\frac{R_{\min} - R_{ino}}{R_{\min}} \right)$$ [acumulando valor na variável]

09: **Fim** do **Se**

10: **Fim** do **Para**

---

onde ino representa o contador do número de nós na malha do duto; e R_{ino} é o raio de curvatura calculado para o nó ino. Assim, penalizam-se as curvas geradas que não atendem ao raio mínimo pré-definido.

Capítulo 6 ESTUDO DE CASOS

6.1 CENÁRIOS E PARÂMETROS DO ALGORITMO

De modo a permitir comparações com resultados anteriores apresentados em BAIOCO (2011) (onde o problema de otimização era tratado como uni-objetivo), o estudo dos casos apresentados a seguir foram baseados nos cenários apresentados naquela referência.

6.1.1 Cenários

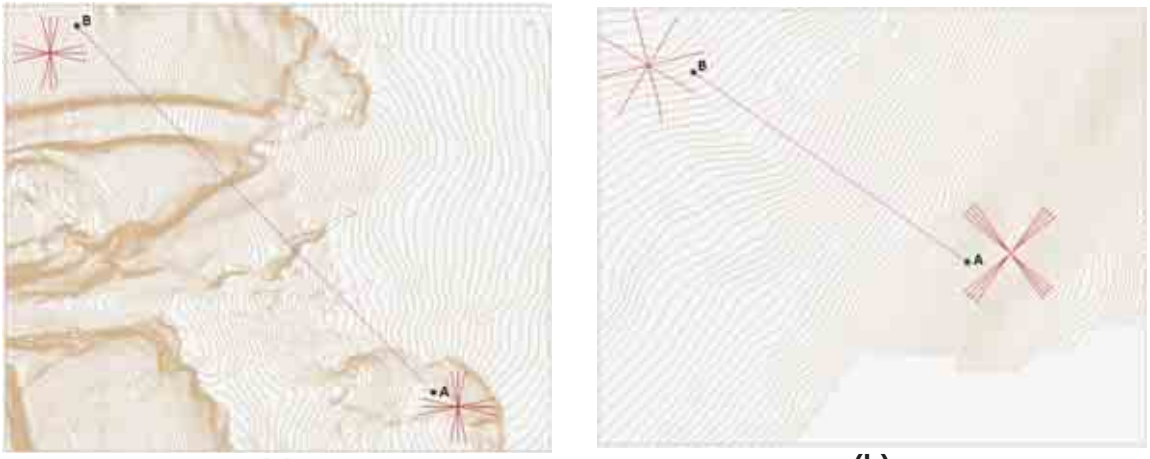
Foram analisadas configurações de duto em dois tipos de batimetria: *acidentada* (ver Figura 6.1(a)) e *suave* (ver Figura 6.1(b)). Além disso, o estudo levou em consideração também, diferentes lâminas d'água: águas rasas e intermediárias (ver Tabela 6.1). Com isso foi possível a realização de três grupos de testes:

T01 – Batimetria suave em águas rasas

T03 – Batimetria suave em águas intermediárias

T05 – Batimetria acidentada em águas intermediárias

Esta variação na lâmina d'água teve o intuito de avaliar a influência das cargas hidrodinâmicas. Em águas rasas avaliou-se a influência da onda e da corrente, já no cenário em águas intermediárias a corrente tem papel mais significativo do que a onda.



(a) **(b)**

Figura 6.1: Batimetria **(a)** Acidentada, **(b)** Suave

Em águas rasas e intermediárias foi empregado um solo arenoso, ver Tabela 6.2 para obter as propriedades físicas do solo.

Tabela 6.1: Profundidade das extremidades A e B da rota

	Batimetria	
	Acidentada	**Suave**
Águas rasas	---	A: 529 m B: 14m
Águas intermediárias	A: 927 m B: 94 m	A: 639 m B: 124 m

Tabela 6.2: Propriedades físicas do solo

Propriedades	**Águas Rasas e Intermediárias**
Coeficiente de atrito do solo μ_{solo}	0,7
Peso seco do solo γ_s	18.000 N/m³
Peso molhado do solo $\gamma_s^{'}$	13.500 N/m³
Resistência não-drenada do solo s_u	10.000 N/m³

6.1.2 Características Físicas e Geométricas do Duto

O duto utilizado para o processo de otimização foi função da batimetria empregada, sendo as propriedades físicas e geométricas apresentadas na Tabela 6.3, a seguir.

Tabela 6.3: Propriedades físicas e geométricas do duto

Propriedades	**Batimetria**	
	Acidentada	**Suave**
Diâmetro externo do duto OD	0,76200 m ≅ 30 in	0,32385 m ≅ 12 ¾ in
Espessura parede de aço t_s	0,01270 m ≅ 1/2 in	0,01905 m ≅ 3/4 in
Espessura do revestimento t_p	0,00397 m ≅ 1/6 in	0,07600 m ≅ 2 in
Espessura da camada de concreto t_c	---	---
Peso específico aço ρ_s	77.000 N/m³	77.000 N/m³
Peso específico do revestimento ρ_p	18.100 N/m³	8.826 N/m³
Peso específico concreto ρ_c	29.800 N/m³	---

6.1.3 Carregamentos Ambientais

As avaliações da estabilidade das rotas candidatas foram efetuadas considerando um conjunto de casos de carregamento (multi-carregamento).

Os carregamentos ambientais de correnteza foram adotados segundo a profundidade da lâmina d'água. Desta forma, para os casos em águas rasas (T01) e intermediárias (T03 e T05) foram adotadas as velocidades de corrente referentes à Tabela 6.4 (melhor visualizada na Figura 6.2), considerando a velocidade da corrente no fundo para oito direções (N, NE, E, SE, S, SW, W e NW) e dois períodos de retorno (10 e 100 anos). Vale destacar que a velocidade da corrente no fundo sofre uma redução devido ao efeito de camada limite inferior e da direcionalidade (BAIOCO, 2011).

Tabela 6.4: Velocidade da corrente próxima ao leito marinho (m/s)

Direção	Período de Retorno (anos) Águas Rasas e Intermediárias	
	10	100
N	0,54	0,65
NE	0,75	0,96
E	0,62	0,74
SE	0,65	0,77
S	0,96	1,18
SW	1,09	1,41
W	0,63	0,75
NW	0,55	0,61

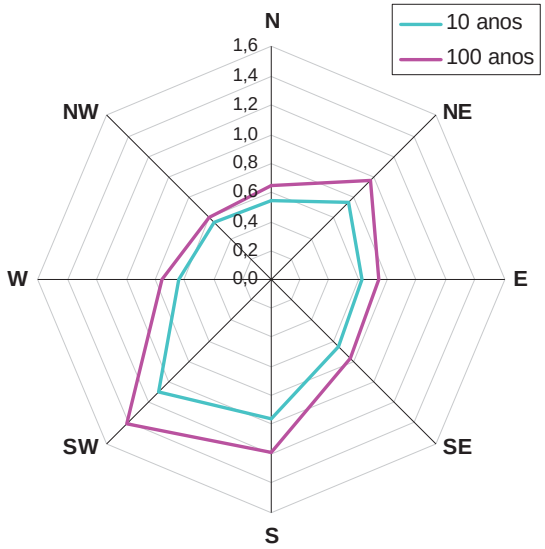


Figura 6.2: Velocidade da corrente próxima ao leito marinho (m/s)

O carregamento de onda utiliza os dados de altura significativa da Tabela 6.6 (Figura 6.3) e período de pico da Tabela 6.5. Vale destacar que a influência da onda é mais expressiva em águas rasas (T01), sendo desconsiderada em águas sucessivamente mais profundas (T03 e T05).

Tabela 6.5: Período de pico da onda T_P

Direção	Período de Retorno (anos)	
	10	100
N	9,20	9,56
NE	9,47	9,77
E	9,90	10,40
SE	10,28	11,63
S	13,54	14,35
SW	14,78	15,55
W	8,22	8,51
NW	8,22	8,51

Tabela 6.6: Altura significativa onda H_s

Direção	Período de Retorno (anos)	
	10	100
N	4,74	5,01
NE	4,88	5,17
E	4,34	4,87
SE	5,72	6,53
S	6,19	7,10
SW	7,16	7,84
W	3,57	3,88
NW	3,57	3,88

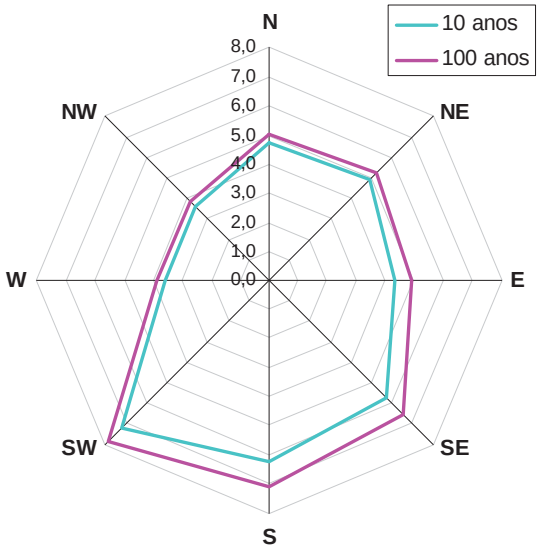


Figura 6.3: Altura significativa onda H_s

6.1.4 Parâmetros do Processo de Otimização

De início foram realizados três grupos de testes (T01, T03, T05) com os seguintes parâmetros de otimização e modelagem apresentados na Tabela 6.7, Tabela 6.8 e Tabela 6.9;

Os parâmetros da Tabela 6.7 foram assim escolhidos em virtude do estudo realizado com as funções matemáticas no Capítulo 4 Validação do Algoritmo. Esclarece-se que os valores de β_m foram adotados como sendo a metade do valor da subtração entre limite superior com o limite inferior de cada variável e estes grupos de valores apresentados na tabela serão repetidos para cada um dos pontos de controle existentes na rota.

Tabela 6.7: Parâmetros do algoritmo SIAMEP

PARÂMETRO	GRUPO DE TESTES		
	T01	**T03**	**T05**
Num. Anticorpos (N_{ab})	10	10	10
Num. Clones (N_{clones})	3	3	3
Tamanho população	*40*	*40*	*40*
Indiv conjunto pareto	40	40	40
Opção de Mutação	Híbrida	Híbrida	Híbrida
Par. Mutação Somática (ρ)	2	2	2
Par. mutação Somática β_m (*)	*0,5* *3118* *180* *2250*	*0,5* *3118* *180* *2250*	*0,5* *6449* *180* *2000*
Par. mutação Somática ($\beta_{m-final}$) (**)	$\beta_m \cdot 10^{-3}$	$\beta_m \cdot 10^{-3}$	$\beta_m \cdot 10^{-3}$
Escopo da mutação somática	2 parâmetros por vez		
Par. mutação diferencial (F)	0,8	0,8	0,8
Escopo da mutação diferencial	Todos os parâmetros podem ser mutados		

(*) β_m definido como sendo o *limite superior subtraído do inferior,* multiplicado por 0,5 para cada parâmetro. Os valores repetem-se para cada PI (na tabela seguinte) existentes na modelagem;
(**) $\beta_{m-final}$ representa o valor de β_m após sofrer um decaimento linear com o num. de avaliações ou gerações.

Na Tabela 6.8 o *Num máx pontos (PI)* representa o número máximo de pontos de controle existentes na rota; *fator de ativação* representa a probabilidade de ativação de cada ponto; e o *Tam elemento* é o tamanho do elemento da modelagem da rota;

Tabela 6.8: Parâmetros da modelagem

PARÂMETRO ROTA BASE	GRUPO DE TESTES		
	T01	**T03**	**T05**
Num máx pontos (N_{pb})	4	4	3
Fator de ativação	0,5	0,5	0,5
Tam elemento (L_{elem})	2% $dist_{AB}$	2% $dist_{AB}$	2% $dist_{AB}$
$dist_{AB}$ (metros)	*12.473,99*	*12.473,99*	*25.794,25*

LIMITE PARÂMETROS	[mín; máx]	[mín; máx]	[mín; máx]
Ativação ponto	[0;1]	[0;1]	[0;1]
Afastamento (η)	[0; $0,5 \cdot dist_{AB}$]	[0; $0,5 \cdot dist_{AB}$]	[0; $0,5 \cdot dist_{AB}$]
Ângulo (θ)	[0; 360]	[0; 360]	[0; 360]
Raio	[500; 5.000]	[500; 5.000]	[1.000; 5.000]

Na Tabela 6.9 estão apresentados os pesos e valores limites das penalizações utilizadas na otimização de rota de dutos. Os valores foram os mesmo adotados por BAIOCO (2011).

Tabela 6.9: Penalizações utilizadas para todos os testes

PENALIZAÇÃO	PESO (mk_j)	VALOR LIMITE
Cruzamento sobre si mesmo ($penal_1$)	1	---
Cruzamento de obstáculo ($penal_2$)	1	---
Comprimento mínimo entre duas curvas ($penal_3$)	1	100 m
Comprimento reto mínimo de saída ($penal_4$)	1	50 m
Comprimento reto mínimo de chegada ($penal_5$)	1	50 m
Limites da Batimetria ($penal_6$)	1	---
Declividade longitudinal ($penal_7$)	5	5 graus
Raio de curvatura mínimo ($penal_8$)	1	500 kN *tração residual*

Os testes foram realizados com um total de 10 rodadas cada, com 500 gerações cada como sendo o critério de parada.

6.2 RESULTADOS

Os resultados apresentados em forma de fronte de pareto representam as 10 rodadas do algoritmo apresentadas em um único fronte. Para cada rodada é gerado um fronte de pareto individual. Ao final do limite de rodadas estes frontes individuais são unidos e são realizadas as operações de pareto (dominância e supressão) neste conjunto, resultando no conjunto mais representativo das múltiplas rodadas. A seguir são apresentados resultados dos testes realizados.

$\mathbf{F_1}$ representa o comprimento da rota gerada (em *metros*) e $\mathbf{F_2}$, o peso total requerido de lastro para torná-la estável (em *Newtons*).

6.2.1 T01 – Batimetria suave em águas rasas

Na Figura 6.4 é apresentado o fronte de pareto para o grupo de teste T01. Na Tabela 6.10 estão expostos os resultados ilustrados na Figura 6.4. Em vermelho, estão apresentados os resultados que compõem o fronte com 5 soluções. O número entre chaves "{ }" denota o número da referida solução na Figura 6.4. Nota-se a existência de um fronte bastante vasto, com soluções com valores de comprimento de duto da ordem de 12.500 metros a 19.000 metros. O fronte está bem distribuído e com espaçamento relativamente homogêneo, com o problema de apresentar um número menor de soluções na parte inferior a direita.

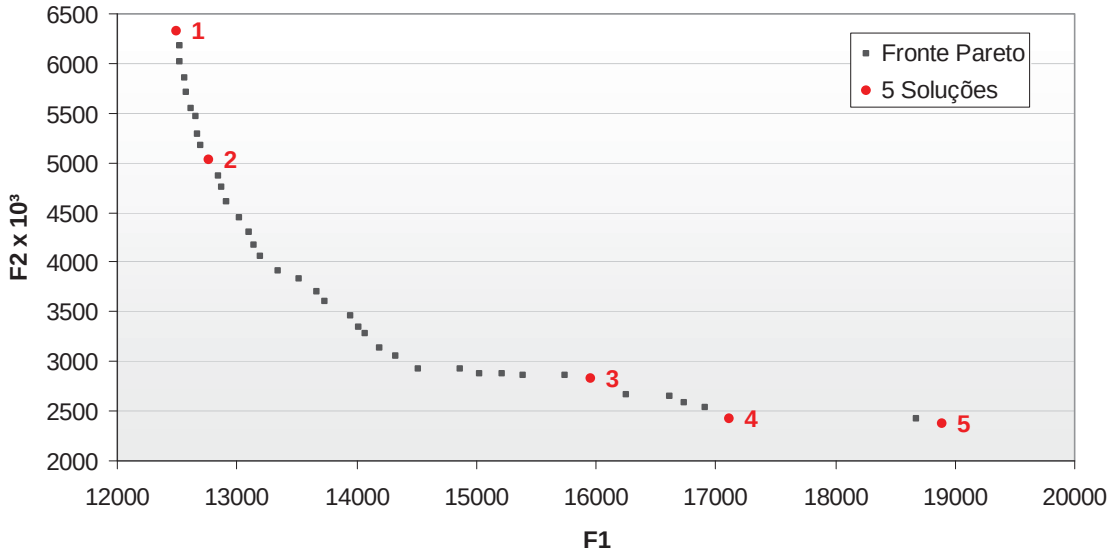


Figura 6.4: Comprimento ($\mathbf{F_1}$) x Peso total requerido ($\mathbf{F_2}$) para T01

Tabela 6.10: Resultado para T01

F_1	$F_2 x 10^3$	F_1	$F_2 x 10^3$
{5} 18900	**{5} 2379**	13666	3693
18684	2419	13522	3831
{4} 17114	**{4} 2420**	13342	3913
16912	2531	13196	4057
16744	2586	13149	4165
16621	2650	13108	4295
16256	2657	13025	4439
{3} 15961	**{3} 2830**	12910	4607
15739	2851	12879	4758
15398	2865	12851	4864
15215	2871	**{2} 12766**	**{2} 5021**
15024	2872	12696	5174
14863	2921	12675	5288
14517	2921	12663	5458
14336	3046	12619	5549
14196	3138	12580	5703
14077	3278	12562	5852
14020	3348	12529	6020
13951	3457	12526	6173
13742	3604	**{1} 12498**	**{1} 6322**

Tabela 6.11: Resultados de BAIOCO (2011) para T01

Rota	Comprimento = F_1	Peso Total = F_2
Critério Absoluto $[k_{duto} = 01; k_{lastro} = 0,01]$	14.822 m	2.979 kN
Critério Absoluto $[k_{duto} = 10; k_{lastro} = 0,01]$	14.217 m	3.156 kN
Critério Absoluto $[k_{duto} = 50; k_{lastro} = 0,01]$	12.580 m	5.674 kN

A Figura 6.5 representa todas as 40 rotas apresentadas nos pontos do fronte de pareto expostos na Tabela 6.10, e, a Figura 6.6 representa a seleção de um fronte com apenas 5 soluções geradas automaticamente pelo algoritmo e grifadas na Tabela 6.10 com os números de {1} a {5}.

Nota-se de maneira geral que quanto menor o comprimento da rota, maior o peso requerido para torná-la estável. E também que as rotas que apresentam maior comprimento tendem a evitar o carregamento direto nas direções NE-SO e N-S tentando alinhar o trecho partindo de B até o ponto de interseção com essas direções. Em virtude da baixa profundidade, há maior peso requerido de lastro na região em torno do ponto B (14m de profundidade), que se encontra numa profundidade mais suscetível aos carregamentos de onda. Tais resultados estão de acordo com o descrito em BAIOCO (2011) e apresentado na Tabela 6.11.

Na Figura 6.6 e na Figura 6.7 a escala de cor indica: quando vermelho, maior peso é requerido, e quando verde menor peso é requerido, sendo amarelo um peso requerido intermediário.

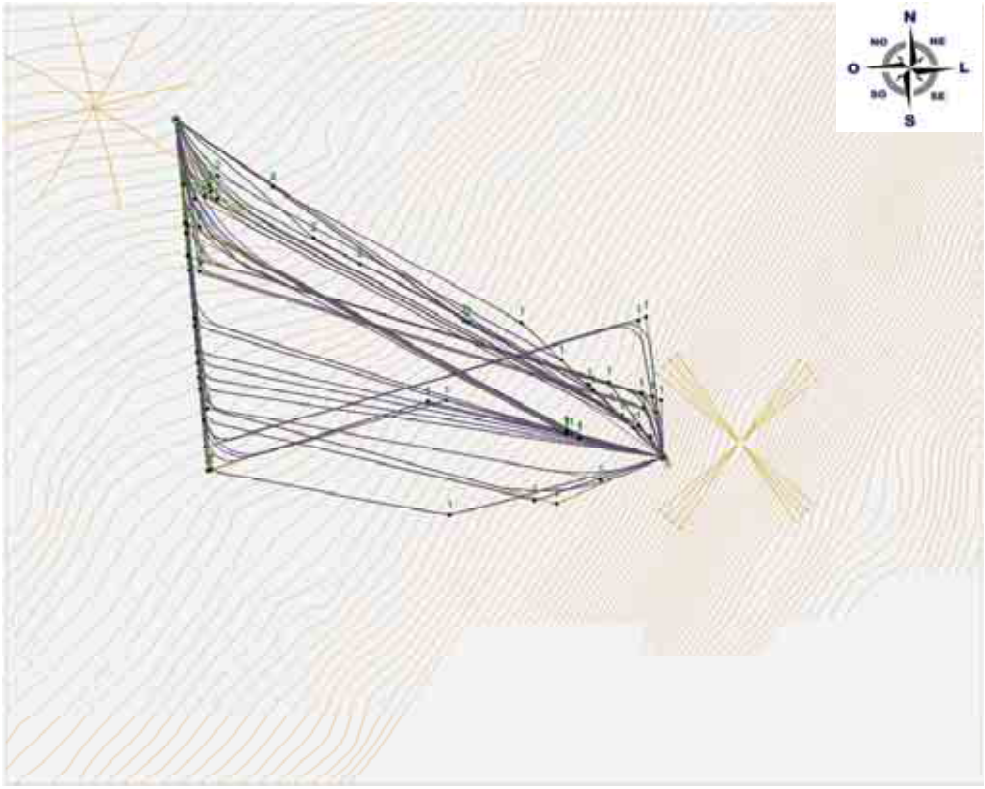


Figura 6.5: Rotas geradas T01 – todas

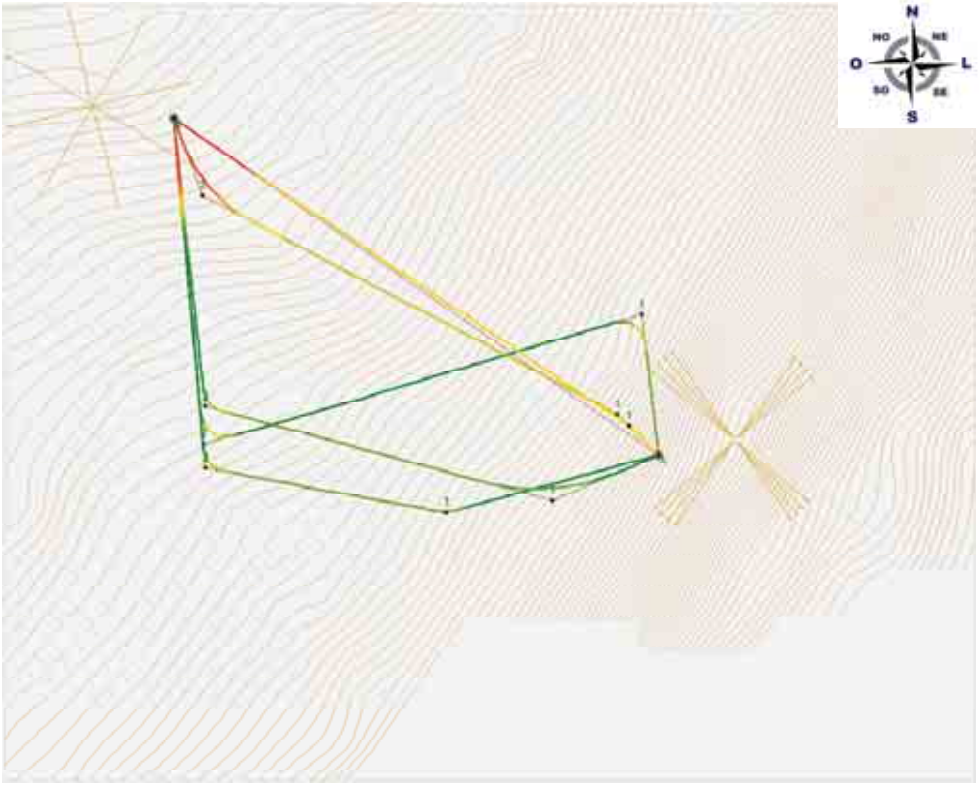


Figura 6.6: T01 – grupo com 5 soluções (visualizando peso requerido)

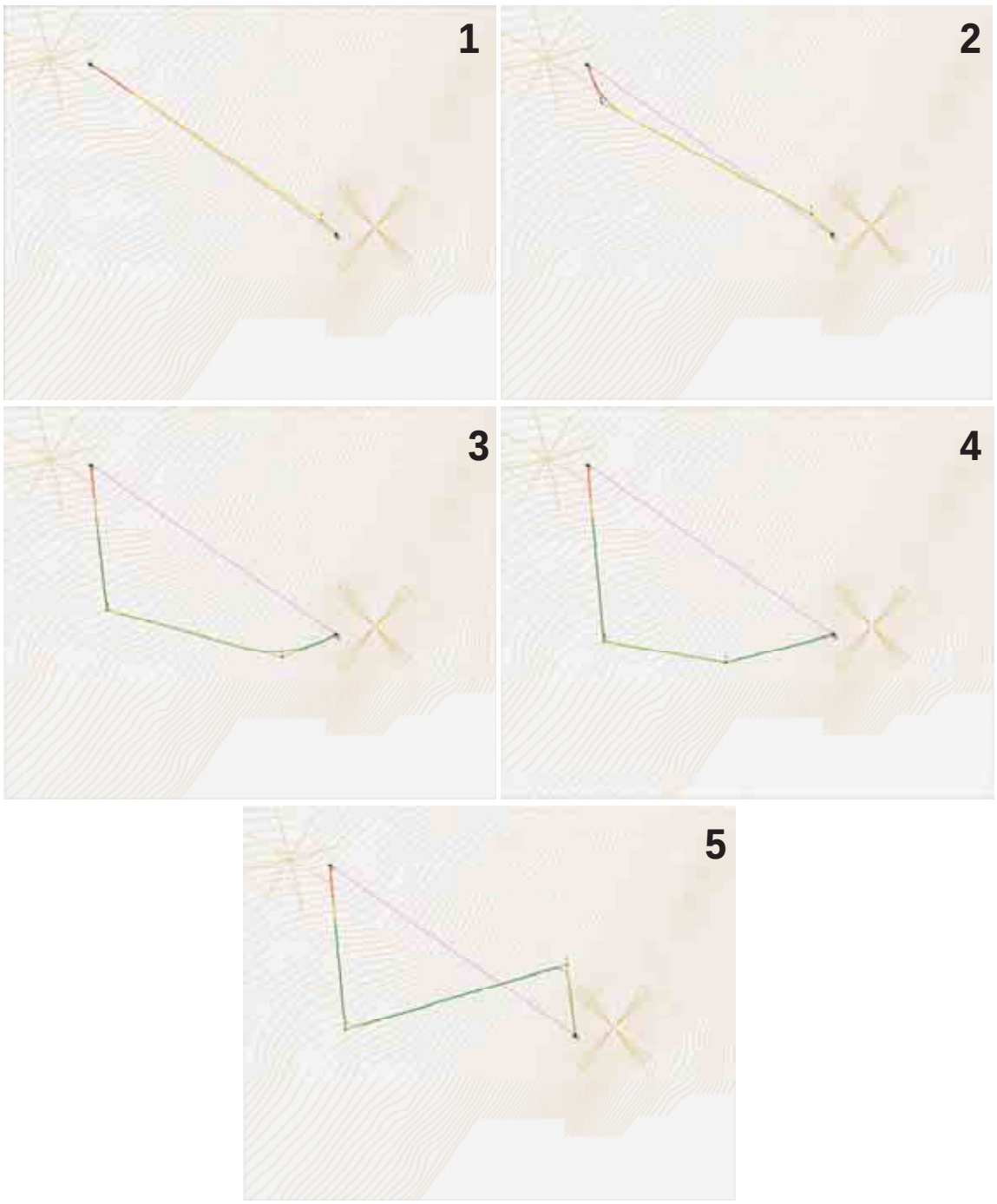


Figura 6.7: Conjunto com as cinco soluções T01, uma a uma

Em BAIOCO (2011) temos resultados que estão expressos na Figura 6.8 apresentada a seguir:

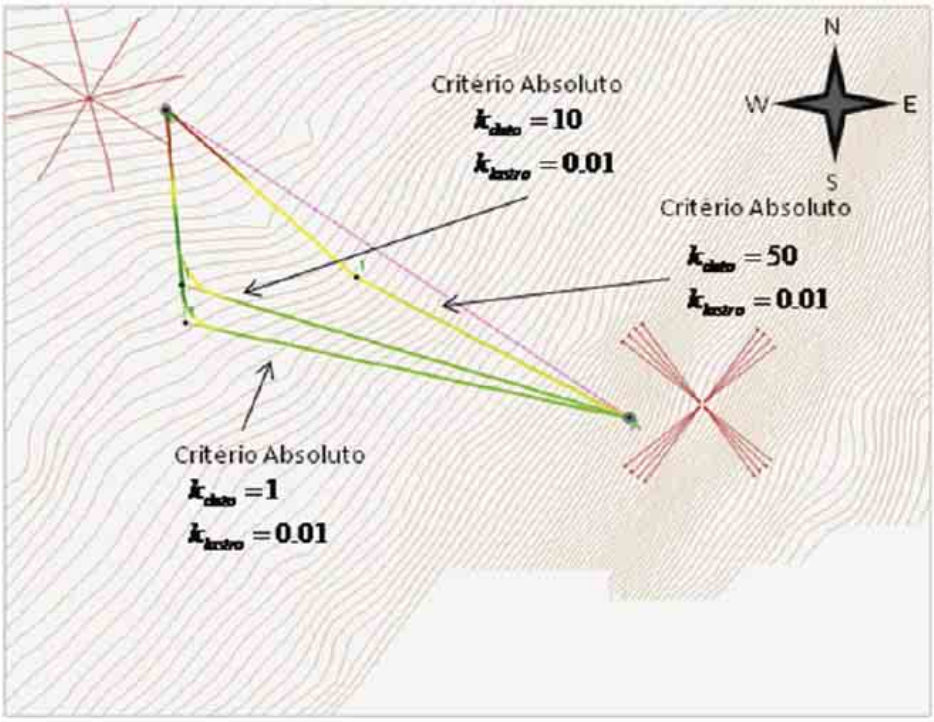


Figura 6.8: Rota gerada T01 em BAIOCO (2011) pelo peso requerido

A partir da análise da Tabela 6.11 e Figura 6.8 (uni-objetivo) e comparando com a Tabela 6.10 e Figura 6.5, Figura 6.6 e Figura 6.7 (multi-objetivo), é possível perceber que os resultados do algoritmo multi-objetivo estão de acordo com a expectativa de soluções visualizadas em BAIOCO (2011). Nota-se que as soluções, por exemplo, (2) e (3) na Tabela 6.10, situando-se na mesma faixa de comprimento e peso de lastro das soluções na Tabela 6.11, apresentam o mesmo comportamento, ou seja, alinham-se na direção do caregamento principal. Sendo este o comportamento esperado. O algoritmo multi-objetivo, no entanto, apresenta um conjunto de soluções amplo, que dá ao tomador de decisão, maior margem de escolha. Pode-se escolher desde uma solução que apresenta o menor comprimento, praticamente alinhada com a reta $\overline{AB}$ (solução {1}), mas com grande peso de lastro; a uma solução com grande comprimento, mas que possui pequeno peso de lastro (solução {5}). Todas as soluções apresentadas atendem ao critério de estabilidade e não possuem violação em nenhuma das restrições, indicando que todas são viáveis em relação aos aspectos técnicos. E isso feito sem a necessidade de modelar uma função objetivo contendo pesos para ponderar cada uma dos objetivos, conforme descrito em BAIOCO (2011).

6.2.2 *T03 – Batimetria suave em águas intermediárias*

Na Figura 6.9 é apresentado o fronte de pareto para o grupo de teste T03. Na Tabela 6.12 estão expostos os resultados ilustrados na Figura 6.9. Em vermelho, são apresentados os resultados que compõem o fronte com 5 soluções. O número entre chaves "{ }" denota o número da referida solução na Figura 6.9. Nota-se a existência de um fronte bastante vasto, com soluções com valores de comprimento da ordem de 12.500 metros a 22.000 metros. O fronte está bem distribuído e com espaçamento relativamente homogêneo, com o problema de apresentar poucas soluções na parte inferior a direita, aparentando "buracos" no fronte de pareto.

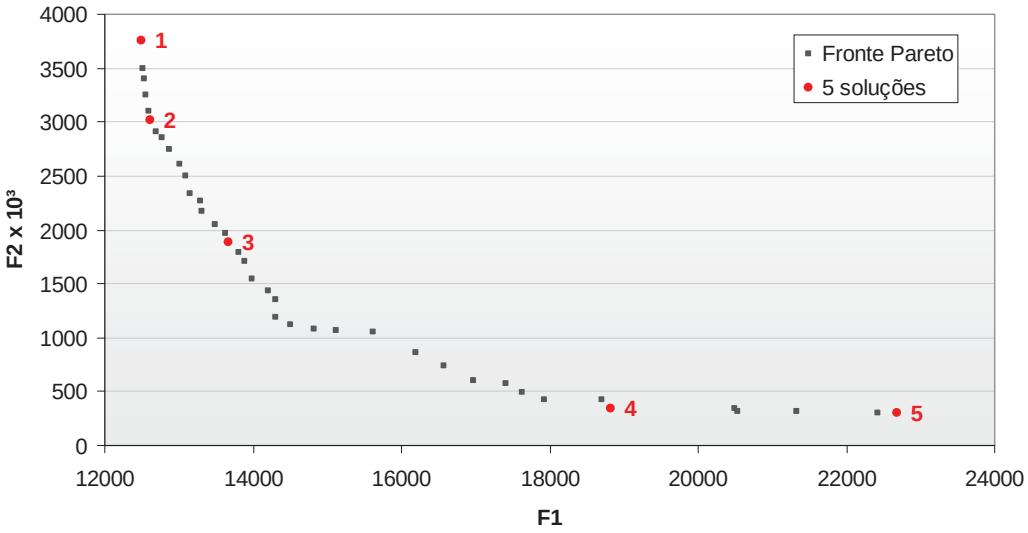


Figura 6.9: Comprimento (F_1) x Peso total requerido (F_2) para T03

Tabela 6.12: Resultado para T03

F_1	$F_2 x 10^3$	F_1	$F_2 x 10^3$
{5} 22692	**{5} 296**	13987	1548
22424	302	13881	1702
21330	309	13815	1791
20546	312	**{3} 13669**	**{3} 1882**
20506	344	13623	1961
{4} 18820	**{4} 344**	13491	2045
18707	418	13323	2172
17927	426	13299	2270
17623	493	13161	2338
17408	578	13086	2504
16966	596	13022	2609
16586	736	12882	2744
16193	863	12781	2848
15619	1056	12706	2913
15121	1071	**{2} 12612**	**{2} 3012**
14829	1080	12592	3099
14506	1118	12558	3256
14304	1190	12534	3397
14302	1355	12518	3492
14209	1439	**{1} 12503**	**{1} 3751**

Tabela 6.13: Resultados de BAIOCO (2011) para T03

Rota	Comprimento = F_1	Peso Total = F_2
Critério Absoluto $[k_{duto} = 01; k_{lastro} = 0,01]$	14.588 m	1.328 kN
Critério Absoluto $[k_{duto} = 10; k_{lastro} = 0,01]$	13.461 m	2.127 kN
Critério Absoluto $[k_{duto} = 50; k_{lastro} = 0,01]$	12.619 m	3.054 kN

A Figura 6.10 representa todas as 40 rotas apresentadas nos pontos do fronte de pareto da Tabela 6.12, e, a Figura 6.11 representa a seleção de um fronte com apenas 5 soluções geradas automaticamente pelo algoritmo e grifadas na Tabela 6.12 com os números de {1} a {5}.

Para este grupo de resultados também nota-se que quanto menor o comprimento da rota, maior o peso requerido para torná-la estável. Alem disso, percebe-se que as rotas que apresentam maior comprimento tendem a evitar o carregamento direto nas direções NE-SO e N-S tentando alinhar o trecho partindo de B até o ponto de interseção com essas direções ou do ponto de interseção 1 até o ponto de interseção 2.

Nota-se também que, em torno de B o peso requerido é pequeno em virtude da baixa influência dos carregamentos de onda na profundidade de B (124m), diferente do que ocorreu em T01, onde a profundidade menor exigiu maior peso de lastro. Comparando-se o caso T03 com o T01, nota-se que os pesos requeridos foram de maneira geral menores em T03, em virtude da não incidência do carregamento de onda em uma profundidade mais elevada. Tais resultados estão de acordo com o descrito em BAIOCO (2011).

Percebe-se que as soluções de maior comprimento, por exemplo, soluções {3}, {4} e {5} tendem a apresentar maior peso de lastro nas regiões de curva, onde estariam mais sujeitas à apresentar dificuldades relacionadas à estabilidade por causa da mudança na direção e carregamento de correnteza. Soluções com menor comprimento, apresentando consequentemente o maior peso de lastro, por exemplo, a solução {2}, concentra o lastro na região mais próxima do ponto *A* provavelmente por causa da incidência perpendicular do carregamento de correnteza nesta região e da alta declividade.

Na Figura 6.11 e na Figura 6.12 a escala de cor indica: quando vermelho, maior peso é requerido, e quando verde menor peso é requerido, sendo amarelo um peso requerido intermediário.

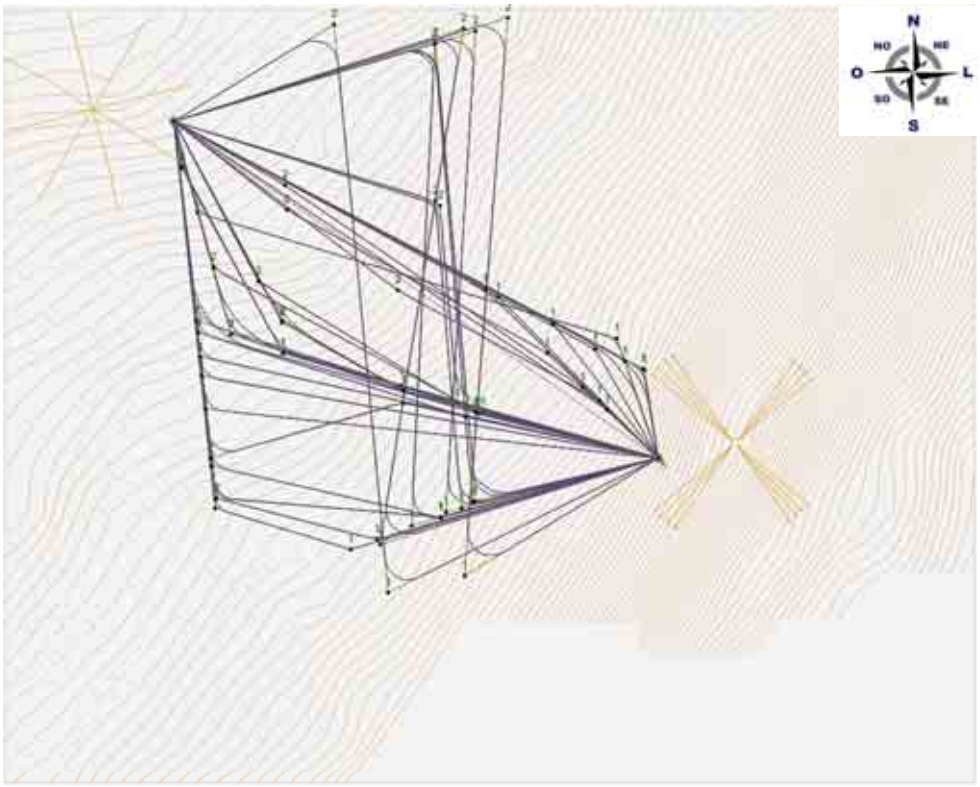


Figura 6.10: Rotas geradas T03 – todas

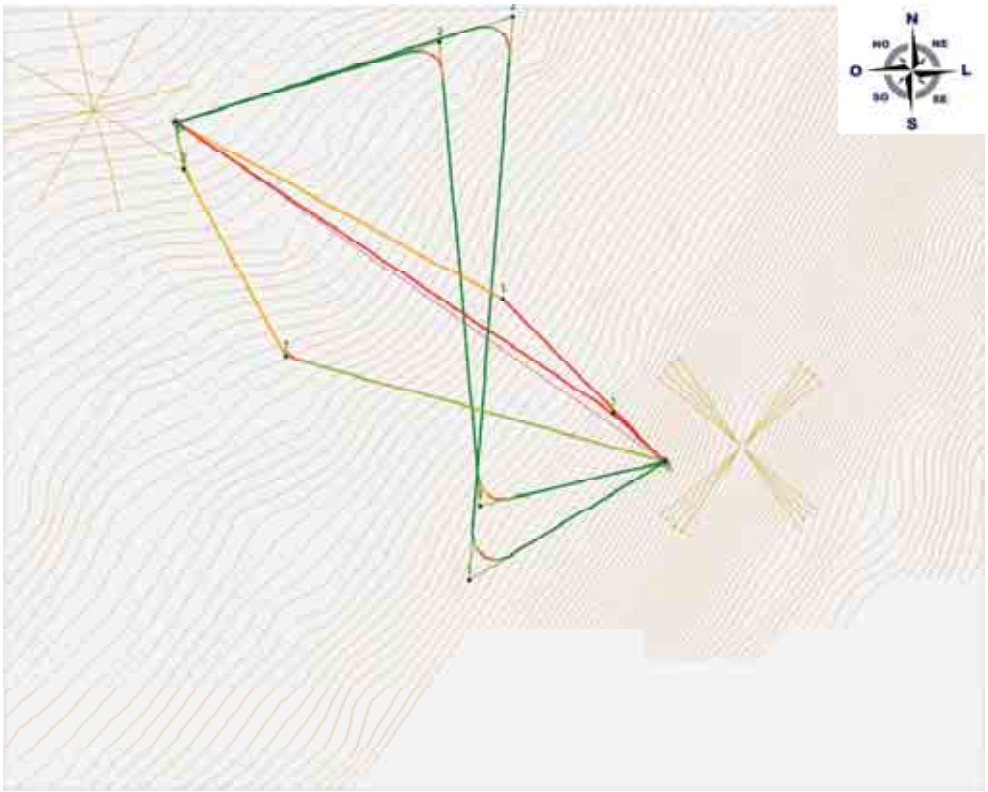


Figura 6.11: T03 – grupo com 5 soluções (visualizando peso requerido)

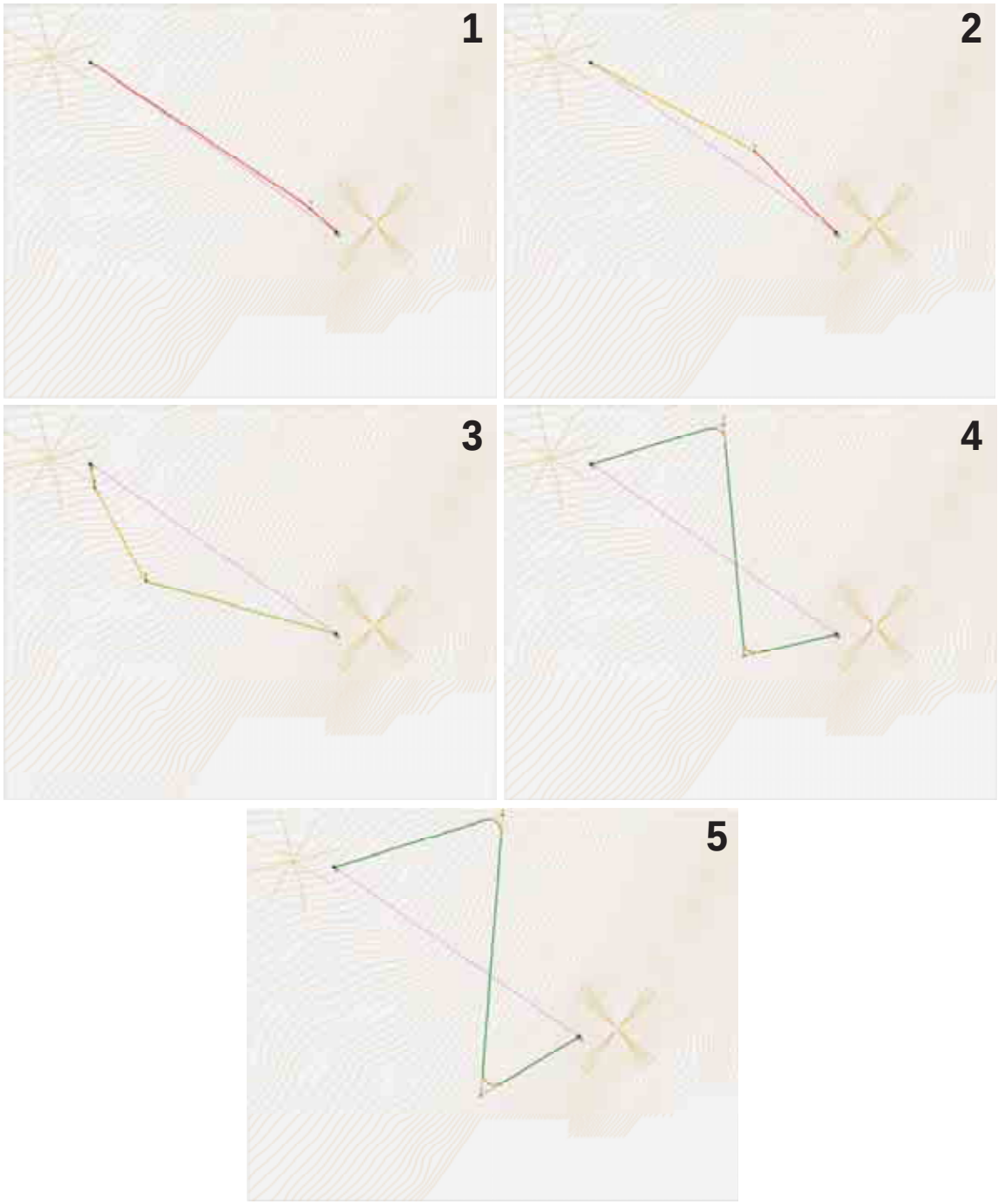


Figura 6.12: Conjunto com as cinco soluções T03, uma a uma

Em BAIOCO (2011) temos resultados que estão expressos na Figura 6.13 apresentada a seguir:

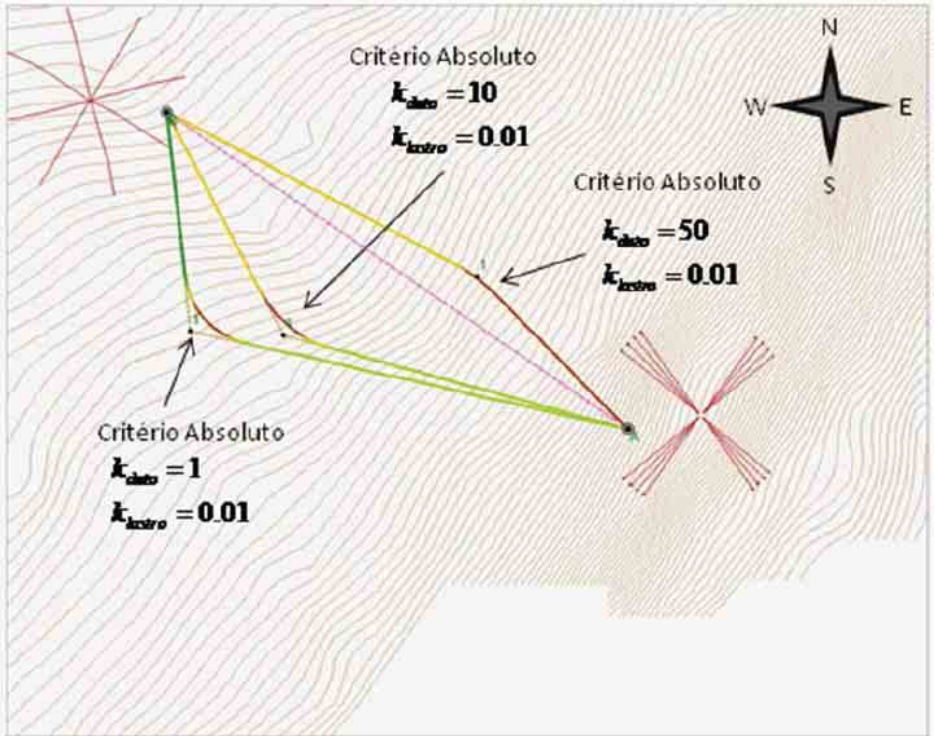


Figura 6.13: Rota gerada T03 em BAIOCO (2011) pelo peso requerido

A partir da análise da Tabela 6.13 e da Figura 6.13 (uni-objetivo) e comparando com a Tabela 6.12 e Figura 6.10, Figura 6.11 e Figura 6.12 (multi-objetivo) é possível perceber para os comprimentos gerados na Tabela 6.13 temos pesos requeridos semelhantes na Tabela 6.12. O que indica que os resultados do algoritmo multi-objetivo estão de acordo com a expectativa de soluções visualizadas em BAIOCO (2011). No entanto, o algoritmo multi-objetivo permite avaliar uma faixa maior de soluções como opção a serem escolhidas pelos tomadores de decisão, ao contrário do tratamento uni-objetivo que inicialmente possui somente uma solução como resposta. Este fato é importante porque ao tomador de decisão é interessante poder visualizar diferentes opções de solução que satisfazem aos requisitos do problema. Pode-se escolher desde uma solução que apresenta o menor comprimento, praticamente alinhada com a reta $\overline{AB}$ (solução {1}), mas com grande peso de lastro; a uma solução com grande comprimento, mas que possui pequeno peso de lastro (solução {5}). Todas as soluções atendendo aos critérios de estabilidade e não apresentando violação das restrições, indicando que todas são tecnicamente viáveis.

6.2.3 T05 – Batimetria acidentada em águas intermediárias

Na Figura 6.14 é apresentado o fronte de pareto para o grupo de teste T05. Na Tabela 6.14 estão expostos os resultados ilustrados na Figura 6.14. Em vermelho, são apresentados os resultados que compõem o fronte com 5 soluções. O número entre chaves "{ }" denota o número da referida solução no gráfico da Figura 6.14. Neste caso, nota-se a existência de um fronte pouco distribuído e mal formado, ou seja, com poucas soluções e soluções extremamente espaçadas. Inicialmente, tem-se a impressão de que o algoritmo não foi capaz de encontrar a solução para o problema.

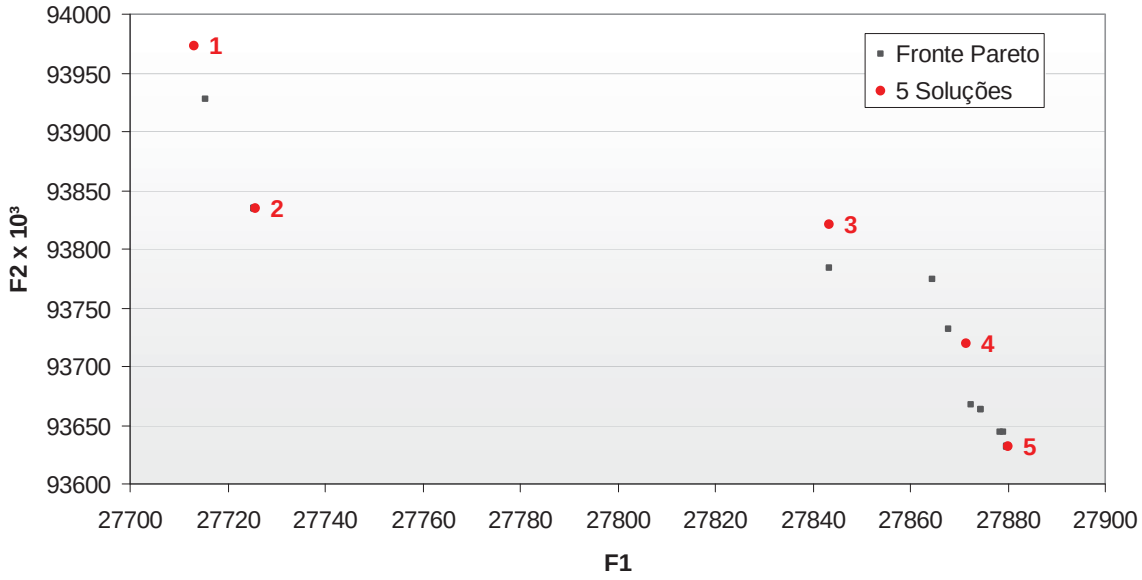


Figura 6.14: Comprimento (F_1) x Peso total requerido (F_2) para T05

Tabela 6.14: Resultado para T05

F_1	F_2 x10³	F_1	F_2 x10³
{5} 27880,27	**{5} 93630,97**	27872,46	93667,69
27880,27	93631,02	**{4} 27871,49**	**{4} 93719,05**
27880,25	93631,03	27867,88	93731,80
27880,24	93631,10	27864,71	93773,43
27880,24	93631,13	27843,47	93783,83
27879,86	93631,39	**{3} 27843,44**	**{3} 93821,10**
27879,86	93631,41	**{2} 27725,93**	**{2} 93833,62**
27879,85	93631,45	27725,57	93833,76
27879,84	93631,47	27725,54	93833,83
27879,13	93643,21	27725,48	93834,30
27878,58	93643,79	27725,40	93834,58
27874,42	93663,29	27715,70	93927,79
27874,42	93663,39	**{1} 27713,12**	**{1} 93972,66**

Tabela 6.15: Resultados de BAIOCO (2011) para T05

Rota	Comprimento = F_1	Peso Total = F_2
Critério Absoluto	27.789 m	94 x 10^3 kN

Nota-se pela Figura 6.14 que a variação de pesos requerido e comprimento é bem menor do que nos casos anteriores, com uma variação da ordem de 200m apenas entre as soluções de maior e de menor comprimento. Isto indica que as soluções estão bem próximas, o que pode ser visualmente percebido na Figura 6.15 e na Figura 6.14 se atentar para a escala no gráfico.

Este comportamento deve-se ao fato do carregamento incidente de correnteza e onda ser bem menos significativo para águas intermediárias do que para águas rasas, fazendo com que não haja grande variação no peso requerido de lastro para tornar a rota estável, e, desta forma, o conjunto de soluções pareto-ótimas seja um grupo muito próximo de soluções, decidido basicamente pelo comprimento da rota. Neste caso, um limitador importante será a declividade longitudinal que determinará basicamente onde a rota poderá existir.

Nota-se também que o grupo de soluções obtidas está muito similar ao resultado uni-objetivo apresentado por BAIOCO (2011). Entende-se que isto acontece porque para este teste o algoritmo funcionou basicamente como uni-objetivo uma vez que o peso requerido de lastro não sofreu grandes variações devido ao carregamento de onda ser desconsiderado para esta profundidade e devido à influência da correnteza de fundo ser relativamente pequena. Para complementar os resultados são apresentadas também as visualizações por peso requerido (Figura 6.16) e por declividade longitudinal (Figura 6.17).

Percebe-se pela Figura 6.16 e Figura 6.17 que nas regiões de maior declividade e de curva é onde estão localizados os maiores pesos de lastro para tornar a rota estável.

Na Figura 6.16 e na Figura 6.18 a escala de cor indica: quando vermelho, maior peso é requerido, e quando verde menor peso é requerido, sendo amarelo um peso requerido intermediário. Nota-se que nas regiões de maior declividade longitudinal, ou seja, regiões mais inclinadas, o peso requerido para tornar o duto estável foi maior do que em regiões com declividade mais suave, menos inclinadas, o que era esperado.

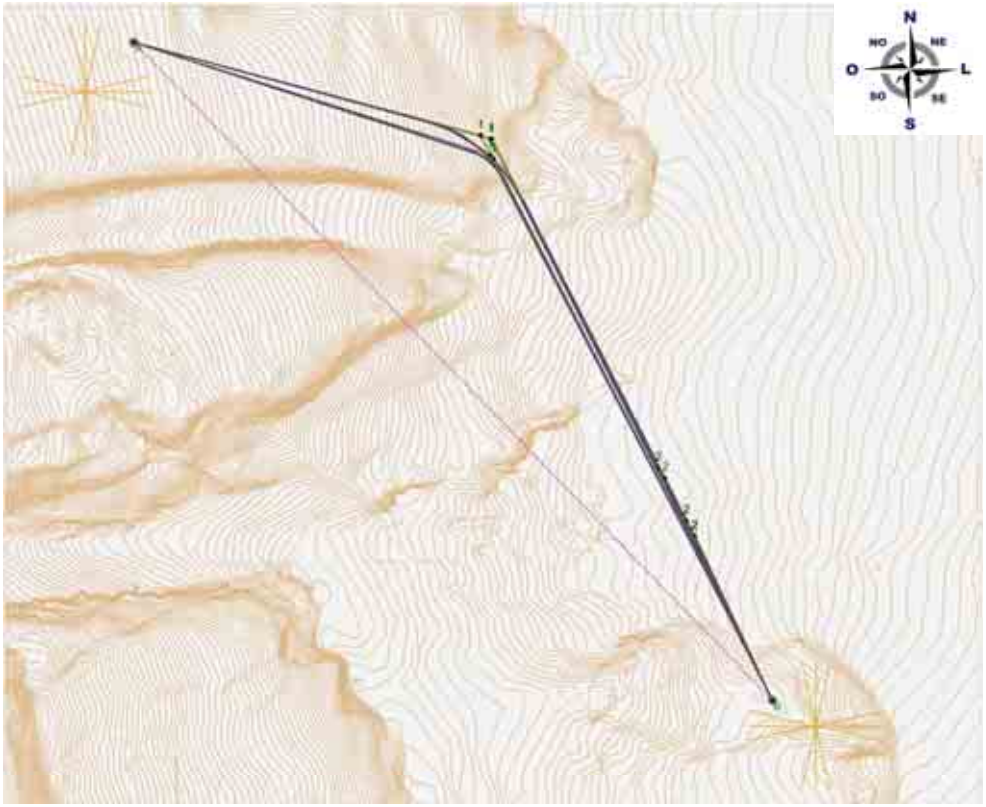


Figura 6.15: Rotas geradas T05 – todas

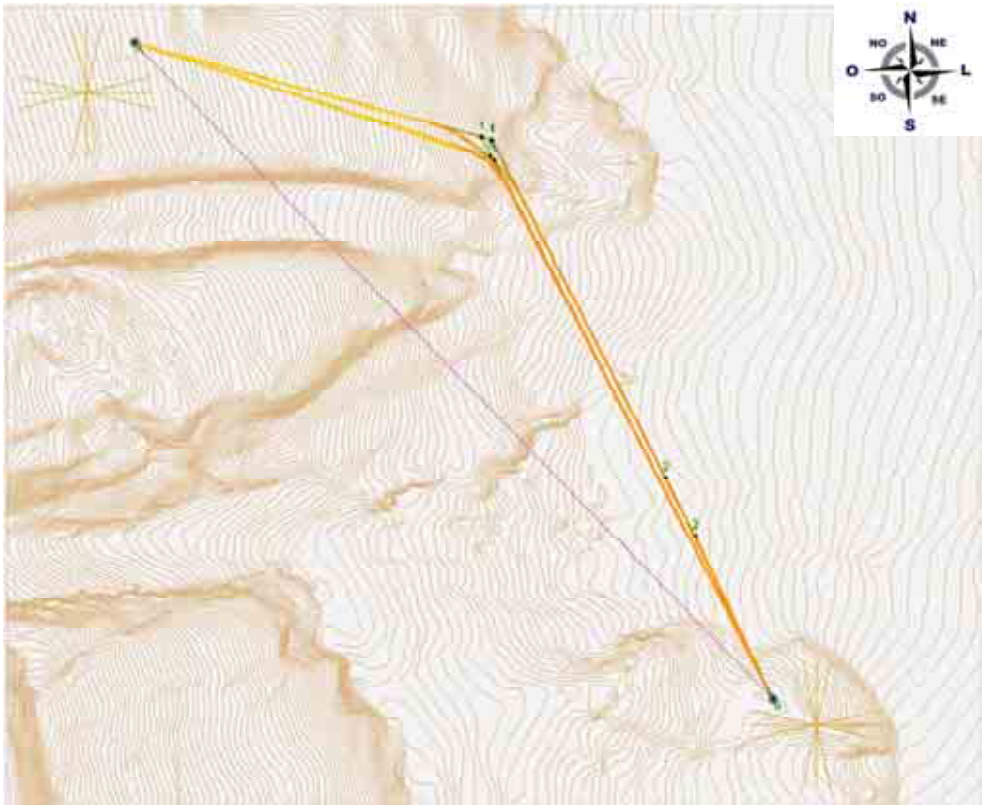


Figura 6.16: T05 – grupo com 5 soluções (visualizando peso requerido)

Na Figura 6.17 a escala de cor indica: quando vermelho escuro, maior é a declividade longitudinal, e quando verde menor é a declividade longitudinal, sendo amarela, uma declividade intermediária.

Figura 6.17: T05 – grupo com 5 soluções (visualizando declividade longitudinal)

A partir da análise da

Tabela 6.15 e Figura 6.19 (uni-objetivo) e comparando com a Tabela 6.14 e Figura 6.15, Figura 6.16, Figura 6.17 e Figura 6.18 (multi-objetivo) pode-se notar que os resultados encontrados pelo algoritmo multi-objetivo estão bem próximos dos obtidos pelo uni-objetivo utilizando a formulação de peso requerido de lastro incorporada na função de aptidão, tanto com relação ao comprimento total e peso de lastro requerido, como visualmente. Na situação particular, em que pela profundidade elevada o carregamento de onda pode ser desprezado e a correnteza de fundo apresenta pouca influência sobre a estabilidade do duto, o algoritmo multi-objetivo tende a apresentar pequena variação na parcela de F_2, desta forma, o critério F_1 passa a ser preponderante na escolha do grupo de soluções pareto-ótimas. Como a batimetria limitava as opções de rota pela restrição de 5º (ver Tabela 6.9) o algoritmo multi-objetivo conseguiu apenas contemplar soluções próximas umas das outras com pequenas variações de comprimento. Ainda assim, o que se tem são um conjunto de soluções satisfazendo aos critérios de estabilidade e nenhuma violação das restrições.

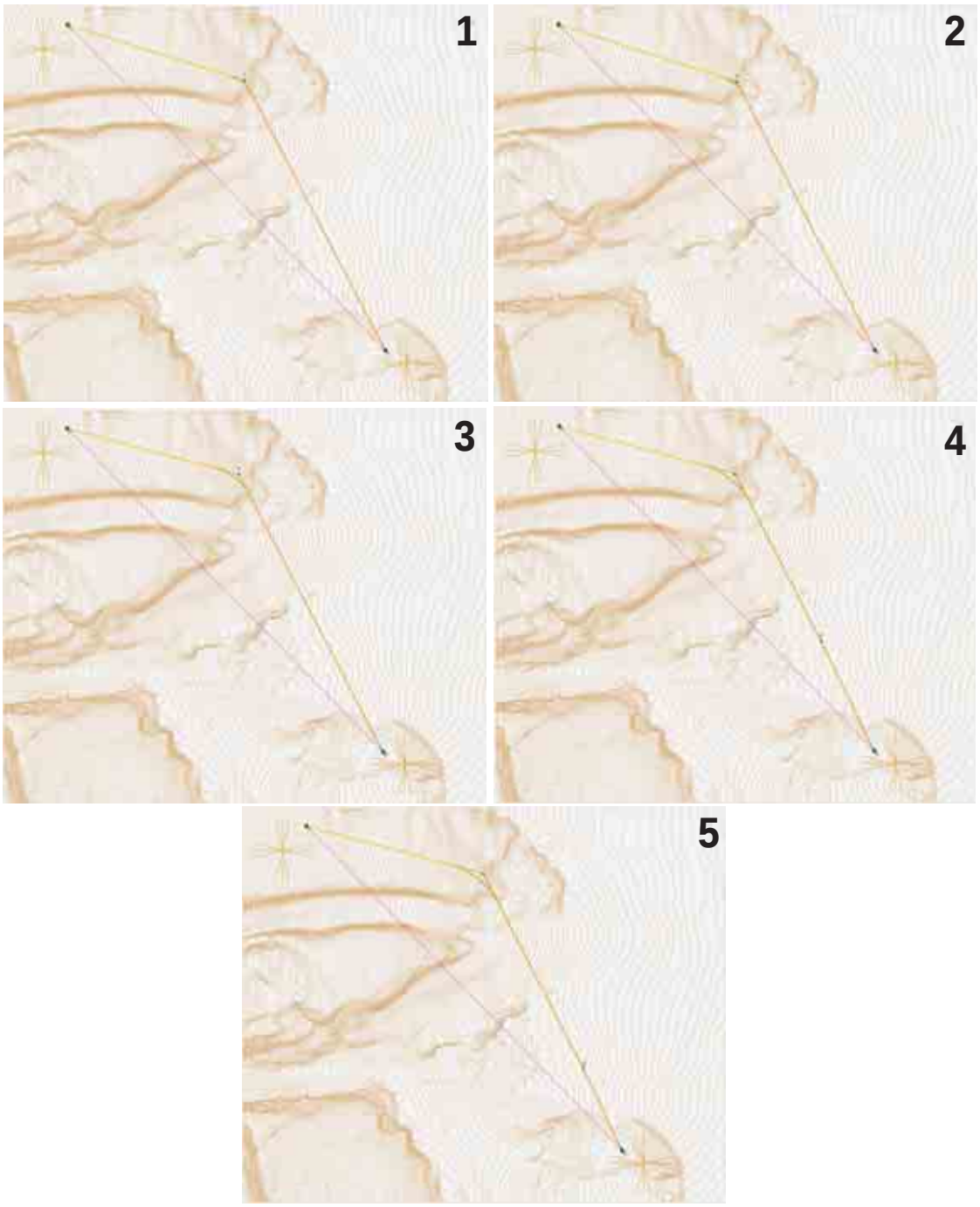


Figura 6.18: Conjunto com as cinco soluções T05, uma a uma

Em BAIOCO (2011) temos resultados que estão expressos na Figura 6.19 apresentada a seguir.

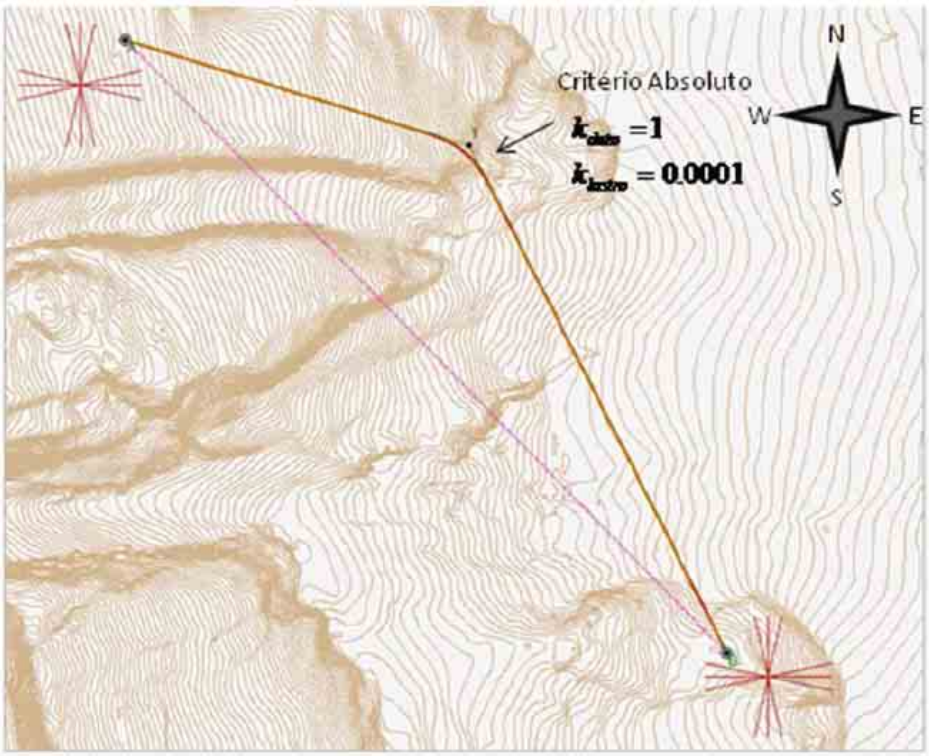


Figura 6.19: Rota gerada T05 em BAIOCO (2011) pelo peso requerido

6.2.4 *Estudo adicional em T05*

No intuito de verificar as questões relacionadas ao caso T05 relativo à declividade, um estudo extra foi realizado, repetindo o caso T05 só que com um limite máximo de declividade maior, igual a **8°**, mantendo-se todos as demais configurações e parêmetros de T05. Desta forma será possível avaliar a limitação ou não da batimetria no caso estudado.

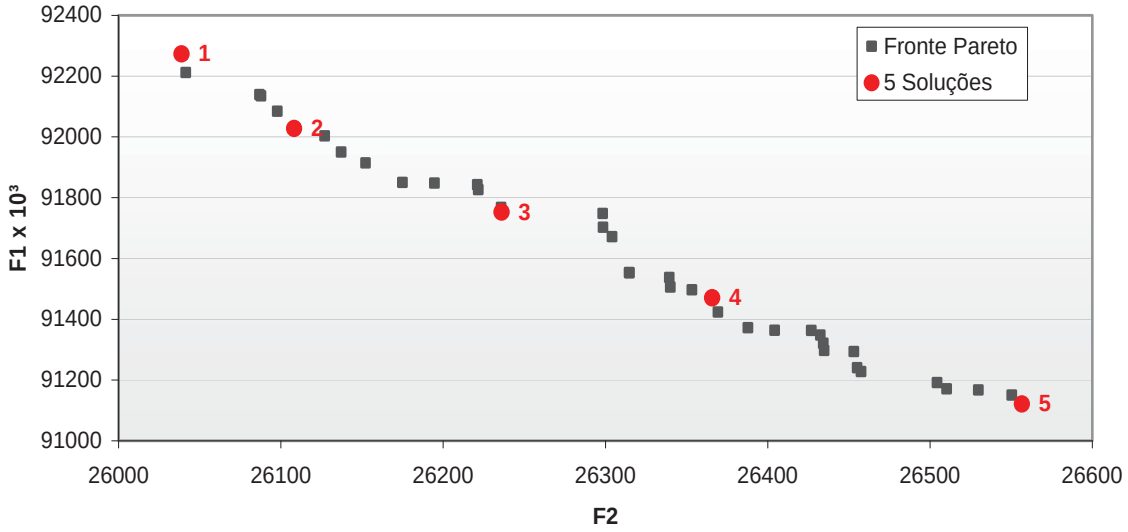


Figura 6.20: Comprimento (F_1) x Peso total requerido (F_2) para T05 adicional

Tabela 6.16: Resultado para T05 adicional

F_1	F_2 x10³	F_1	F_2 x10³
{5} 26556,58	**{5} 91121,44**	26314,70	91553,64
26550,43	91150,50	26304,08	91671,15
26529,80	91166,97	26298,54	91702,87
26510,25	91170,92	26298,32	91747,95
26504,36	91191,70	**{3} 26236,08**	**{3} 91752,77**
26457,56	91227,81	26235,71	91767,62
26455,14	91239,99	26221,72	91826,06
26453,07	91293,76	26221,11	91842,36
26434,81	91296,98	26194,64	91847,62
26434,29	91320,76	26175,01	91849,81
26432,49	91347,31	26152,21	91914,10
26426,86	91363,10	26137,20	91949,91
26404,26	91363,72	26127,05	92003,37
26387,85	91371,97	**{2} 26108,25**	**{2} 92027,44**
26369,32	91423,68	26097,89	92084,65
{4} 26365,81	**{4} 91470,38**	26087,77	92134,83
26353,25	91496,90	26086,86	92138,97
26339,97	91505,95	26041,48	92211,22
26339,39	91537,07	26039,43	92270,74
26314,83	91552,31	**{1} 26038,80**	**{1} 92272,94**

Nota-se pela Figura 6.20 e Tabela 6.16 um fronte com soluções mais bem distribuídas do que em T05. Ao contrário de T05, foi conseguido um conjunto com 40 soluções de pareto, o que era desejado. A variação de pesos requeridos de lastro e

comprimentos de duto está maior do que T05, observando-se uma variação da ordem de 600m entre as soluções de maior e de menor comprimento. Isto indica que as soluções estão próximas, o que pode ser visualmente percebido pela Figura 6.20 e Figura 6.21.

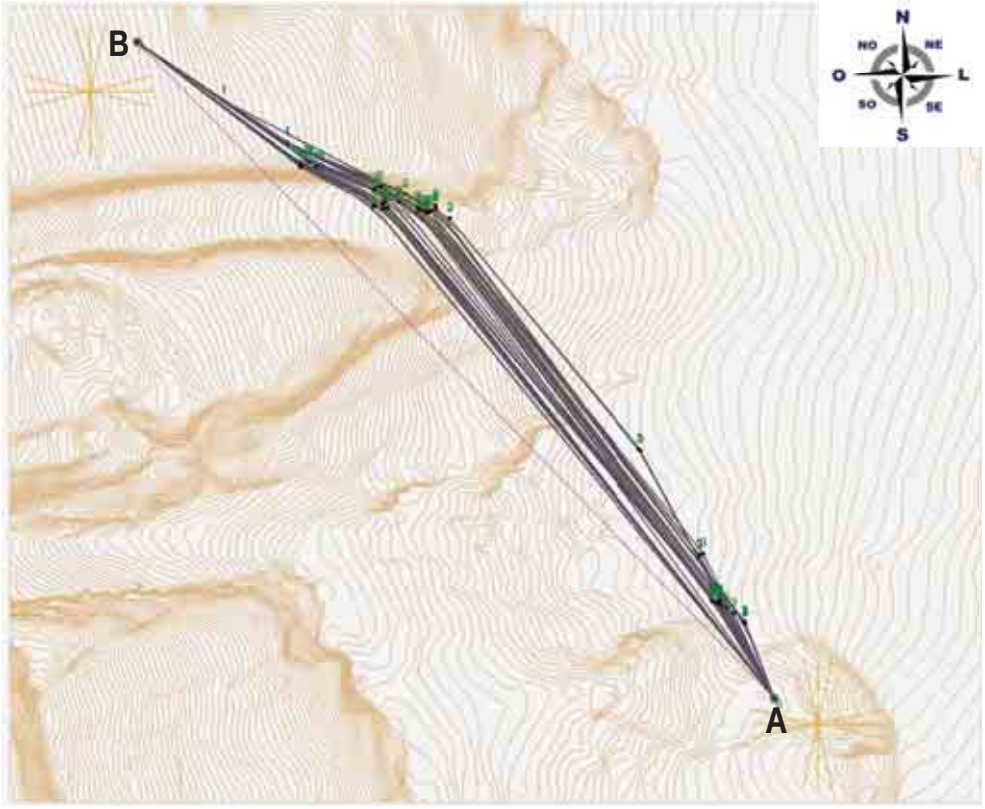


Figura 6.21: T05 adicional, com todas as 40 rotas geradas

Na Figura 6.21 observa-se um conjunto com 40 soluções todas sem violação nas restrições de comportamento. Devido ao limite máximo de declividade ter sido aumentado para 8°, as soluções mudaram de posição em relação a T05 e passaram a poder passar por uma região com declividade mais acentuada, conforme percebe-se na Figura 6.23.

Em relação ao peso de lastro, percebe-se que nas regiões em que o carregamento de correnteza atua de maneira perpendicular à trajetória do duto, o peso requerido de lastro foi maior do que em regiões onde a incidência não era perpendicular, em particular entre o ponto B e o ponto 2 – região mais amarelada na Figura 6.22. Muito próximo de A o peso requerido apresenta-se na cor verde, indicando o menor peso requerido.

Pode-se concluir então que a batimetria (declividade) é sim um fator limitante para a geração das rotas; lembra-se ainda que isto ocorre devido à configuração da equação de restrição de declividade longitudinal, escolhida como exponencial, e por isso a importância na escolha desta equação.

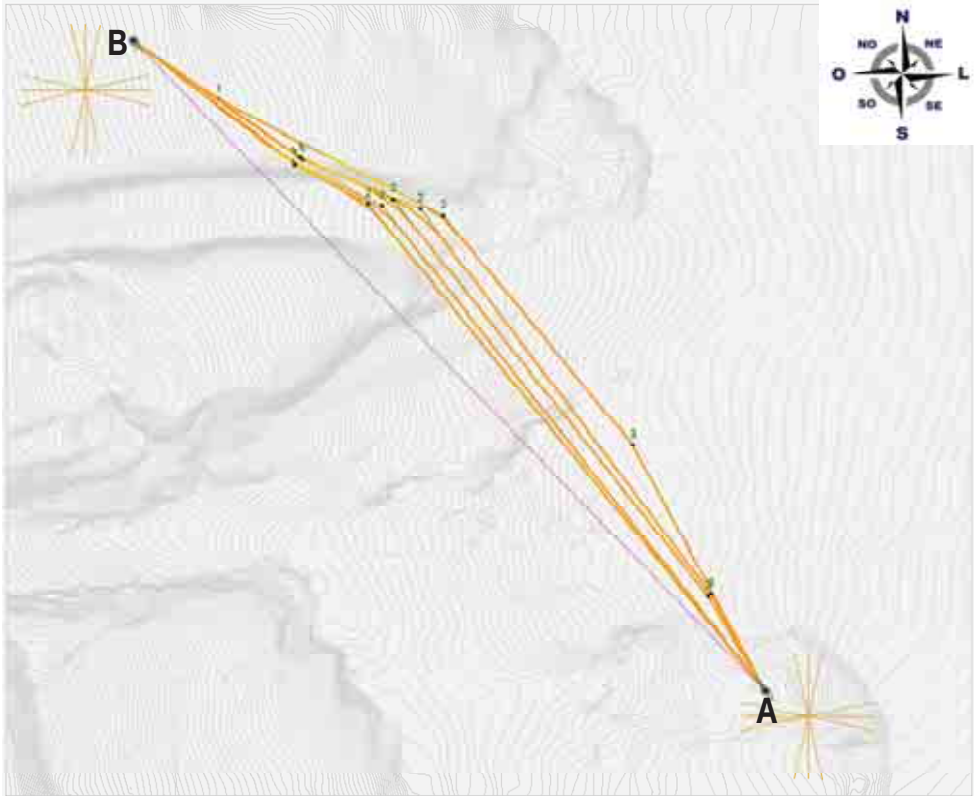


Figura 6.22: T05 adicional, grupo com 5 soluções (visualizando peso requerido)

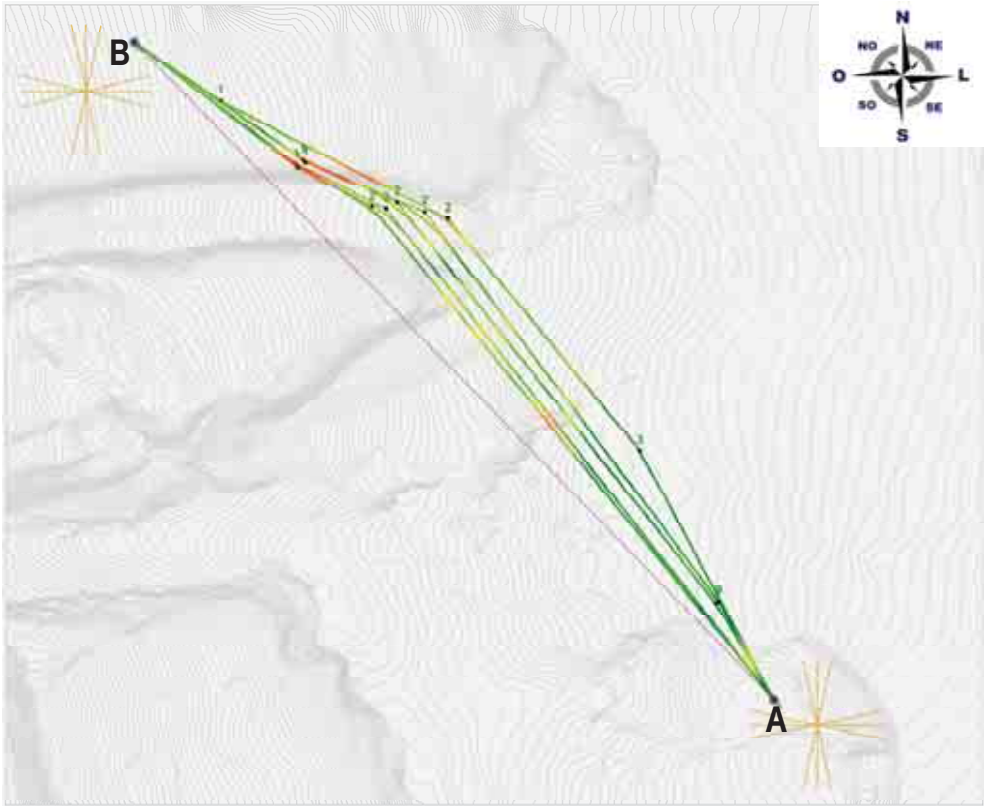


Figura 6.23: T05 adicional, grupo com 5 soluções (visualizando declividade longitudinal)

6.3 COMENTÁRIOS

Através do estudo de casos foi possível perceber que a ferramenta computacional é capaz de gerar rotas que apresentam resultados compatíveis com o esperado. O algoritmo mostrou-se eficaz na resolução do problema proposto de otimização de rotas de duto, gerando múltiplas soluções que poderão ser levadas em consideração na escolha da melhor rota por um especialista.

Um aspecto de destaque pode ser levantado com relação à comparação entre o comportamento uni-objetivo, com uso de pesos para a ponderação na função objetivo com peso de lastro requerido, e o comportamento das soluções com função objetivo em multi-objetivo. Pode-se destacar que a não necessidade de desenvolvimento de uma função com pesos para ponderar cada um dos objetivos é algo de grande interesse que pode facilitar o desenvolvimento do projeto e agilizar a obtenção de resultados. A abordagem multi-objetivo gera ao final do processo de otimização um conjunto de soluções ótimas que atendem aos critérios de estabilidade e não possuem violação em nenhuma das restrições. Tal conjunto de soluções engloba soluções não pecebidas pelo algoritmo uni-objetivo.

Nota-se também, que na situação em que os carregamentos de onda e correnteza possuem pequena importância na influência nos esforços atuantes no duto (águas intermediárias), F_1 passa a ser preponderante sobre F_2 dando a impressão de que o algoritmo multi-objetivo funciona como uni-objetivo. Este comportamento particular é muito interessante, pois indica que o algoritmo saberia encontrar de maneira adequada as soluções pareto-ótimas, mesmo quando as soluções podem ser decididas por apenas um critério. Então pode-se concluir que para águas intermediárias (ou profundas), onde a influencia dos carregamentos ambientais é pequena, o algoritmo multi-objetivo responderá de maneira mais forte aos critérios de comprimento do duto e suas restrições de comportamento.

Capítulo 7 CONCLUSÃO

7.1 CONSIDERAÇÕES FINAIS

Para alcançar os objetivos propostos nesta Tese (desenvolver um algoritmo bio-inspirado para otimização multi-objetivo com e sem restrições, e implementá-lo em associação a uma modelagem do problema de otimização de rotas que combina objetivos antagônicos), foram estudados diversos algoritmos bio-inspirados disponíveis no estado da arte, em particular os algoritmos multi-objetivo que utilizam o Sistema Imunológico como inspiração para o seu desenvolvimento. Foram vistas algumas métricas de desempenho para a comparação entre algoritmos quando testados em funções conhecidas.

O Algoritmo desenvolvido (SIAMEP) foi apresentado e validado em testes *benchmark* com funções matemáticas (com e sem restrição) e problemas clássicos de engenharia (com restrição) conhecidos e disponíveis na bibliografia apresentada. Pode-se dizer que o correto funcionamento dos algoritmos evolutivos está na dependência de seus operadores, em especial o operador de seleção e o operador de exploração do espaço de busca (por exemplo, a mutação). A ferramenta desenvolvida contempla estas características de maneira a obter convergência e espalhamento das soluções no fronte de pareto gerado. Em particular, a mutação híbrida, desenvolvida para esta tese, apresentou resultados promissores com uma capacidade de busca e distribuição das soluções ao longo da fronteira de pareto.

Conforme percebido dos resultados mostrados nas tabelas e gráficos, o algoritmo SIAMEP apresenta bom desempenho quando comparado aos melhores algoritmos do estado da arte no caso das funções matemáticas (com e sem restrição). Em problemas de engenharia disponíveis em artigos (Viga bi-apoiada, treliça de duas barras e viga soldada) os resultados do SIAMEP também foram extremamente satisfatórios. É importante destacar aqui a dificuldade na obtenção de tabelas com resultados de métricas de desempenho para a comparação dos algoritmos, principalmente para as funções com restrições. Imagina-se que neste caso, a dificuldade seja também em virtude da obtenção das soluções exatas para o fronte. Muitos artigos apresentam somente o resultado por meio de figuras mostrando o fronte de pareto obtido, dificultando a comparação dos algoritmos.

Como vantagem, o SIAMEP é um algoritmo simples que apresenta boa convergência com pequeno número de avaliações e bom espalhamento das soluções ao longo da fronteira de pareto unindo duas mutações: somática e diferencial modificada.

Uma dificuldade do algritmo está na operação de remoção das soluções do conjunto externo de pareto. Tal operação apresenta gasto computacional e implica, as vezes, em perda de soluções promissoras.

Para a otimização de rota de dutos, inicialmente é descrita a representação geométrica de uma rota, e na sequência são apresentadas a codificação do problema no algoritmo de otimização, as funções objetivos e as funções de restrição. Destaca-se que a representação geométrica do problema permite que rotas geradas ultrapassem obstáculos, com cruzamento ou sem cruzamento, contornando o obstáculo. Além disso são levados em consideração critérios de estabilidade e influência da batimetria na determinação da rota.

No estudo de caso foram avaliados três diferentes cenários T01, T03 e T05 e comparados com o trabalho de BAIOCO (2011). Em T01 e T03 o resultado é um fronte de pareto bem distribuído contendo 40 soluções. Em T05 o resultado é um fronte com múltiplas soluções próximas (similares) aparentando um resultado similar ao do tratamento uni-objetivo.

Pode-se acrescentar ainda que o algoritmo foi capaz de obter frontes de pareto bem distribuídos e ainda, que a não necessidade de desenvolvimento de uma função com pesos para ponderar cada um dos objetivos é algo de grande interesse que pode facilitar o desenvolvimento do projeto e agilizar a obtenção de resultados. A abordagem multi-objetivo gerou ao final do processo de otimização um conjunto de soluções ótimas que atendem aos critérios de estabilidade e não possuem violação em nenhuma das restrições. Tal conjunto de soluções engloba soluções não pecebidas pelo algoritmo uni-objetivo. E ainda, que para águas intermediárias (ou profundas), onde a influência do carregamento de onda é nula, o algoritmo multi-objetivo responderá de maneira mais forte aos critérios de comprimento do duto e suas restrições de comportamento.

7.2 Proposta para Trabalhos Futuros

Como o trabalho foi dividido em duas partes, uma relativa ao desenvolvimento do algoritmo multi-objetivo e outra relativa à sua aplicação na otimização de rota de dutos, as sugestões de trabalhos futuros serão também divididas em duas partes.

O algoritmo multi-objetivo desenvolvido pode ser melhorado. Tais melhoramentos podem resultar do esforço em se compreender melhor o comportamento do algoritmo frente aos parâmetros de configuração ou mesmo a implementação de novas ferramentas que venham a contribuir com a capacidade de busca, convergência e manutenção da diversidade, como:

- Verificação do impacto da escolha do número de parâmetros a sofrerem mutação (escopo da mutação);
- Melhorar a mutação diferencial modificada colocando um parâmetro similar ao β da mutação somática para atuar, também nesta mutação, como redutor da ordem de grandeza dos parâmetros mutados. Isto deverá permitir maior faixa de variação da mutação e, conseqüentemente, maior aplicabilidade em problemas gerais.
- Outras formas de decaimento do parâmetro β da mutação somática (ou híbrida) poderiam ser testadas;
- Estudos na mutação híbrida incluindo variação no percentual de mudança entre a mutação somática e diferencial modificada;
- Melhoramentos poderiam ser feitos na função de mutação, disponível em artigos científicos como KELSEY *et al*. (2003);
- Novas implementações de tratamento de restrições para o caso multi-objetivo como o artigo de YEN (2009) ou BARBOSA & LEMONGE (2003, 2008) que já foi apreciado em LUCENA (2011) para o caso uni-objetivo;
- Estudo para melhoramentos dos operadores do algoritmo multi-objetivo: seleção e supressão da memória. Este dois operadores influenciam fortemente o resultado final do algoritmo em termos de convergência e manutenção da diversidade;
- Utilização de outros algoritmos multi-objetivos bio-inspirados para se comparar o desempenho aplicado a problemas de engenharia e funções matemáticas;

Seguindo o escopo do trabalho, na parte relativa à otimização de rota de dutos, podem ser feitos os seguintes novos desenvolvimentos (alguns deles já estão em andamento):

- Incorporar novos critérios para avaliação das rotas candidatas, incluindo a verificação do comportamento em vãos livres de acordo com a norma DNV RP-F105 (2006), em uma etapa "local" de otimização, para refinar ainda mais ao redor das soluções ótimas geradas em uma etapa "global" onde esses critérios não são considerados. Isso inclui a implementação de recursos para a análise estrutural do duto apoiado após o lançamento (para definir a configuração de equilíbrio e identificar os vãos livres) (JACOVAZZO, 2011), que vai gerar também, como outro resultado, a avaliação correta da tração residual que é empregada no critério de raio mínimo;

- Incorporar critérios mais refinados de verificação de estabilidade hidrodinâmica, de acordo com a norma DNV RP-F109 (2007) (no trabalho atual considerou-se apenas o critério de estabilidade absoluta);
- Permitir a otimização de rotas passando por um terceiro ponto fixo, intermediário entre os pontos extremos A e B (ou, alternativamente, permitr a otimização "por partes, primeiro de um trecho A-B e depois de outro trecho B-C, neste caso garantindo a continuidade da curvatura da rota);
- Revisar as funções de penalização e incluir novos melhoramentos, em especial na função que penaliza a declividade longitudinal; por esta possuir influência na determinação da rota quando a influência do peso de lastro é pequena, caso de águas intermediárias e profundas onde os carregamentos de onda são desprezíveis.
- Estudar a implementação de métodos de penalização adaptativa (APM) (BARBOSA & LEMONGE, 2003, 2008) para a modelagem multi-objetivo da otimização de rota de dutos submarinos.
- Comparar o resultado gerado por um estudo de caso multi-objetivo com o resultado proposto por um projetista.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBRECHT, C. H., 2005. *Algoritmos Evolutivos Aplicados à Síntese e Otimização de Sistemas de Ancoragem*. Tese D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

BÄCK, T., HOFFMEISTER, F. & SCHWEFEL, H.-P., 1991, "A Survey of Evolution Strategies", In: *Proceedings of the 4th International Conference on Genetic Algorithms.* L. B. Belew and Booker, R. K. (eds.), pp. 2-9, San Diego, USA, Jul.

BARBOSA, H.J.C. & LEMONGE, A.C.C., 2003, "A New Adaptive Penalty Scheme for Genetic Algorithms". *Information Sciences*, v. 156, n. 3-4, pp. 215-251, Nov.

BARBOSA, H.J.C. & LEMONGE, A.C.C., 2008, "An Adaptive Penalty Method for Genetic Algorithms in Constrained Optimization Problems". In: Iba, H., (ed.), *Frontiers in Evolutionary Robotics,* 1 ed., chapter 2, Vienna, Austria, *I-Tech Education and Publishing*.

BAIOCO, J. S., 2011, *Síntese e Otimização de Rotas de Dutos Submarinos Considerando Critérios de Estabilidade Hidrodinâmica*. Dissertação M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

BARDANACHVILI, C. A., 2006, *Otimização multiobjetivo com Estratégias Evolutivas Aplicada a Projetos Estruturais.* Tese D. Sc., Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

BEKKER, J, ALDRICH, C., 2011, "The cross-entropy method in multi-objective optimisation: An assessment". *European Journal of Operational Research*, v. 211, n. 1, pp. 112-121, May.

BERBERT, P. C., 2008, *Sistema Imunológico Artificial para Otimização Multiobjetivo*. Dissertação M.Sc., Universidade Estadual de Campinas, Campinas, São Paulo, SP, Brasil.

CAMPELO, F., GUIMARÃES, F. G., SALDANHA, R. R., IGARASHI, H., NOGUCHI, S., LOWTHER, D. A., & RAMIREZ, J. A., 2004, "A Novel Multiobjective Immune Algorithm using Nondominated Sorting". In: *11th International IGTE Symposium on Numerical Field Calculation in Electrical Engineering*.

CASTRO, L. N., 2001, *Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais*. Tese de D.Sc. UNICAMP, Campinas, São Paulo, Brasil.

CASTRO, L. N., & ZUBEN, F. J. V., 2000, “The Clonal Selection Algorithm with Engineering Applications.” In: *Proceedings of Genetic and Evolutionary Computation Conference,* pp. 36-37, Las Vegas, USA, Jul.

CASTRO, L. N., & ZUBEN, F. J. V., 2001, “Learning and Optimization Using the Clonal Selection Principle”, *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, v. 6, n. 3, pp.239-251, Jun.

CASTRO, R. E., 2001, *Otimização de Estruturas com Multi-objetivos via Algoritmos Genéticos de Pareto*. Tese D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

CHEN, J., & MAHFOUF, M., 2006, “A population adaptive based immune algorithm for solving multi-objective optimization problems”. In: *5th International Conference on Artificial Immune Systems*, v. 4163, pp. 280-293, Springer Berlin / Heidelberg.

CHEN, J., LIN, Q., JI, Z., 2010, “A Hybrid Immune Multiobjective Optimization Algorithm”. *European Journal of Operational Research*, 204, pp. 294-302, Out.

CHEN, M.-R., & LU, Y.-Z., 2008, “A Novel Elitist Multiobjective Optimization Algorithm: Multiobjective Extremal Optimization”. *European Journal of Operational Research*, Elsevier, v. 188, n. 3, pp. 637-651, Ago.

COELLO, C. A. C. & CHRISTIANSEN, A. D., 1998, “Two New GA-based Methods for Multiobjective Optimization”, *Civil Engineering and Environmental Systems,* Gordon and Breach Science Publishers, v. 15, n. 3, pp. 207-243.

COELLO, C. A. C. & CORTÉS, N. C., 2002, “Solving Multiobjective Optimization Problems Using an Artificial Immune System”, Genetic Programming and Evolvable Machines, v. 6, n. 2, pp.163-190, Jan.

COELHO, G. P. & VON ZUBEN, F. J., 2006, “Omni-aiNet: An Immune Inspired Approach for Omni Optimization”. In: *Lecture Notes in Computer Science*. Bersini, H.; Carneiro, J. (Eds.). Proceedings of the 5th International Conference on Artificial Immune Systems, ICARIS 2006, Springer, 4163, pp. 294-308, Oeiras, Portugal.

COLLETTE, Y., & SIARRY, P., 2003, *Multiobjective Optimization: Principles and case studies*. New York. Springer-Verlag Berlin Heidelberg. ISBN 3-540-40182-2.

DEB, K., AGRAWAL, S., PRATAP, A. & MEYARIVAN, T., 2000a, *A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II*. In: Technical Report 200001, Indian Institute of Technology, Kanpur: Kanpur Genetic Algorithms Laboratory (KanGAL).

DEB, K., PRATAP, A., e MOITRA, S. 2000b, “Mechanical Component Design for Multiple Objectives Using Elitist Non-Dominated Sorting GA”. Technical Report 200002. Indian Institute of Technology, Kanpur Genetic Algorithms Laboratory (KanGAL).

DEB, K., 2001, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Indian Institute of Technology, Kanpur, India. ISBN 047187339X.

DEB, K., PRATAP, A., AGRAWAL, S., & MEYARIVAN, T., 2002, "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation*, v. 6, no 2, pp. 182-197. Abr.

DNV - RP - F105, 2006, *Free Spanning Pipelines*. Det Norske Veritas, 1ª Edition.

DNV - RP - F109, 2007, *On-Bottom Stability Design of Submarine Pipelines*. Det Norske Veritas, 1ª Edition.

FEOKTISTOV, V., 2006, *Differential Evolution: In Search of Solutions*. Springer-Verlag Berlin Heidelberg. v. 5, 195 pp, USA.

FERNANDES, D. H., JACOB, B. P., de LIMA, B. S. L. P., & ALBRECHT, C. H., 2008, "Offshore Pipeline Route Assessment via Evolutionary Algorithms". In: *Proceedings of* XXIX CILAMCE, November, Maceió, Brazil.

FOCUS, 2002, *Possibilities in the Pipeline*. Environmental Health Perspectives, v. 10, n. 1.

FONSECA, C. M. & FLEMING, P. J., 1993, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization". In: *Proceedings of the Fifth International Conference on Genetic Algorithm*, Stephanie Forrest, editor, pp 416-423 San Mateo, California, Morgan Kaufmann.

FRESCHI, F. & REPETTO, M., 2005, "Multiobjective Optimization by a Modified Artificial Immune System Algorithm", In: *4th International Conference on Artificial Immune Systems*, Lecture Notes In Computer Science, v. 3627, pp. 248-261 Banff, Alberta, Canadá.

GAO, J. & WANG, J., 2010, "A Novel Artifical Immune System for Multiobjective Optimization". *Computer and Operational Research*, v. 37, pp. 50-61, Abr.

GOLDBERG, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Adison-Wesley, USA.

GUZMÁN, M. A., DELGADO, A., DE CARVALHO, J., 2010, "A Novel Multiobjective Optimization Algorithm Based on Bacterial Chemotaxis". *Engeneering Applications of Artificial Intelligence,* v. 23, n. 3, pp. 292-301, Abr.

HONGYUN, M. & SANYANG, L., 2003, "ISPEA: Improvement for the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization with Immunity", In *Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications*, IEEE Computer Society, pp. 368-372.

HORN, J., NAFPLIOTIS, N. & GOLDBERG, D. E., 1994, “A Niched Pareto Genetic Algorithm for Multiobjective Optimization”, In: *Proceedings of the First IEEE Conference on Evolutionary Computation*, IEEE World Congress on Computational Intelligence, v. 1, pp. 82-87, IEEE Press, Piscataway, New Jersey, USA.

JACOB, B. P., 2006. *Programa PROSIM: Simulação Numérica do Comportamento de Unidades Flutuantes Ancoradas*, Versão 3.0 – Manual de Entrada de Dados, COPPE/UFRJ, Programa de Engenharia Civil, Rio de Janeiro.

JACOB, B. P., 2006, *Programa PROSIM: Simulação Numérica do Comportamento de Unidades Flutuantes Ancoradas*, Versão 3.0 – Manual Teórico, COPPE/UFRJ, Programa de Engenharia Civil, Rio de Janeiro.

JACOB, B. P., ALBRECHT, C. H., LIMA JUNIOR, M.H.A., BAIOCO, J.S.,*et al.*, 2011, *2º Relatório de Andamento do projeto CENPES-Petrobras "Ferramentas Computacionais para Síntese e Otimização de Rota de Dutos"*, LAMCSO - Laboratório de Métodos Computacionais e Sistemas *Offshore*, Rio de Janeiro, Brasil.

JACOVAZZO, B. M., 2011, *Ferramentas Numéricas para Análise, Projeto e Otimização de Operações de Instalação e Movimentação de Linhas e Dutos Offshore*. Qualificação para Tese D. Sc., Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

JIAO, L., GONG, M., SHANG, R., DU, H., & LU, B., 2005, “Clonal Selection with Immune Dominance and Anergy Based Multiobjective Optimization”. In: *3rd International Conference on Evolutionary Multi-Criterion Optimization*, Lecture notes in computer science, Springer, v. 3410, pp. 474-489, Guanajuato, Mexico.

KAVEH, A., LAKNEJADI, K., 2011, “A Novel Hybrid Charge System Search and Particle Swarm Optimization Method for Multi-Objective Optimization”. *Expert System with Applications* – Article In Press.

KELSEY, J. & TIMMIS, J., 2003, “Immune Inspired Somatic Contiguous Hypermutation for Function Optimization”. *Lecture Notes in Computer Science - LNCS*. Springer-Verlag Berlin Heildelberg. GECCO, v. 2723, pp. 207-218.

KNOWLES, J. D. & CONRE, D. W., 2000. “Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy”. *Evolutionary Computation*, v. 8, n.2, pp. 149-172.

LIMA JUNIOR, M. A., 2010, *Verificação de Critério de Estabilidade de Dutos Apoiados no Leito Marinho*. Dissertação M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

LEE, S. H., 2000. *Introdução ao Projeto Geométrico de Rodovias*. Apostila da Disciplina ECV 5115. Universidade Federal de Santa Catarina (UFSC). Florianópolis. Disponível

em: <http://www.scribd.com/doc/17355019/Apostila-de-projeto-geometrico-de-estradas> Acesso em: 11 jan. 2011.

LU, B., JIAO, L., DU, H., & GONG, M., 2005, “IFMOA: Immune Forgetting Multiobjective Optimization Algorithm”. In: *1st International Conference on Advances in Natural Computation*, Lecture Notes in Computer Science, v. 3612, pp. 399-408.

LUCENA, R. R., 2011, *Aplicação de um Método de Penalização Adaptativa em Sistemas Imunológicos Artificiais para Otimização de Problemas com Restrição*. Dissertação M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

LUH, G., CHUEH, C., & LIU, W., 2003, “MOIA: Multi-Objective Immune Algorithm”. *Engineering Optimization*, v.35, n. 2, pp. 143-164, Abr.

LUH, G. C., & CHUEH, C. H., 2004, “Multi-objective Optimal Design of Truss Structure with Immune Algorithm”. *Computers and Structures*, v. 82, n. 11-12, pp. 829-844.

LUO, B., ZHENG, J., XIE, J., & WU. J., 2008, “Dynamic Crowding Distance – A New Diversity Maintenance Strategy for MOEAs”, In: *Fourth International Conference on Natural Computation*, IEEE Computer Society, v. 1, pp. 580-585.

MICHALEWICZ, Z., 1999, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd Edition, Springer Verlag, Berlin Heidelberg.

MEISINGSET, H., HOVE, J., & OLSEN, G., 2004, “Optimization of Pipeline Routes”. In: *Proceedings of the Fourteenth International Offshore and Polar Engineering Conference*, pp. 50-55, Toulon, France.

OLIVEIRA, G. T. S., & SARAMAGO, S. F. P., 2007, “A Contribution to the study about Differential Evolution”. *Ciência e Engenharia*, v. 16, n. 1/2, pp. 1-8. Dez.

PALLI, N., AZRAM, S., MCCLUSKEY, P., & SUNDARARAJAN, R., 1999. “An interactive multistage ϵ-inequality constraint method for multiple objectives decision making”. *ASME Journal of Mechanical Design*, v. 120, n. 4, pp. 678–686, Dez.

PRCI, 2002, *Submarine Pipeline On-Bottom Stability*, Pipeline Research Council International, v. 1 and 2.

PRICE, K. V., STORN, R. M. & LAMPINEN, J. A., 2005, *Differential Evolution – A Pratical Approach to Global Optiization*. Springer-Verlag Berlin Heidelberg. 538 pp.

RECHENBERG, I., 1965, “Cybernetic solution path of an experimental problem”. Royal Aircraft Establishment, Library Translation, n. 1122, Farborough, England.

RICHARDSON, J. T., PALMER, M. R., LIEPINS, G., & HILLIARD, M., 1989, “Some Guidelines for Genetic Algorithms with Penalty Functions”. In: *Proceedings of the Third*

International Conference on Genetic Algorithms, Schaffer, J. D., editor, pp. 191-197. Morgan Kaufmann.

SCHAFFER, J. D., 1984, *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms,* Ph. D. thesis, Vanderbilt University, Nashville, Tennesse, USA.

SCHOTT, J. R., 1995, *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*, Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

SILVERMAN, B. W., 1986, *Density estimation for statistics and data analysis*. London: Chapman and Hall. School of Mathematics University of Bath, UK.

SRINIVAS, N. & DEB, K., 1994, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms". *Evolutionary Computation*, v. 2, n. 3, pp. 221-248.

TAN, K. C., GOH, C. K., MAMUN, A. A., & EI, E. Z., 2008, "An Evolutionary Artificial Immune System for Multi-objective Optimization". *European Journal of Operational Research*, v. 187, pp. 371-392, Jun.

TANAKA, M. WATANABE, H. FURUKAWA, Y. TANINO, T., 1995, "GA-Based Decision Support System for Multicriteria Optimization". In: *Proceedings of IEEE Int. Conf. Systems, Man and Cybernetics*, v. 2, PP 1556-1561.

TICONA, W. G. C., 2003, *Aplicação de Algoritmos Genéticos Multi-Objetivo para Alinhamento de Seqüências Biológicas*. Dissertação M.Sc., Universidade de São Paulo, São Carlos, SP, Brasil.

VIEIRA, I. N., ALBRECHT, C. H., LIMA, B. S. L. P., *et al.*, 2010, "Towards a Computational Tool fot the Synthesis and Optimization of Submarine Pipeline Routes". In: Proceedings of the Twentieth International Offshore and Polar Engineering Conference, Beijing, China.

VIEIRA, I. N., 2009, *Algoritmos Bio-Inspirados Aplicados à Otimização de Risers Rígidos em Catenária*. Dissertação M.Sc., Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

VELDHUIZEN, D. A. V., 1999, *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA.

VERITEC RP – E305, 1998, *On-Bottom Stability Design of Submarine Pipelines*, Veritas Offshore Technology and Services .

WANG, X. L., & MAHFOUF, M., 2006, “ACSAMO: An Adaptive Multiobjective Optimization Algorithm Using the Clonal Selection Principle”. In: *2nd European Symposium on Nature-Inspired Smart Information Systems*, pp. 1-12.

YEN, G. G., 2009, “An Adaptative Penalty Function for Handling Constraint in Multi-objective Evolutionary Optimization”, *Constraint-Handling in Evolutionary* Optimization, *Studies in Computational Intelligence. E. Mezura-Montes (Ed.)*, v. 198, pp. 121-143.

ZHANG, Z., 2007, “Immune Optimization Algorithm for Constrained Nonlinear Multiobjective Optimization Problems”. *Applied Soft Computing Journal*, v. 7, n. 3, pp; 840-857, Jun.

ZHANG, Z., 2006, “Constrained Multiobjective Optimization Immune Algorithm: Convergence and Application”. *Computers and Mathematics with Applications*, v. 52, n. 5, pp. 791-808, Set.

ZITZLER, E. & THIELE, L., 1998, *An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach*. In: Technical Report 43, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Switzerland.

ZITZLER, E., LAUMANNS, M. & THIELE, L., 2001, *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. In: Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Switzerland.

ANEXO A – RESULTADO PARA FUNÇÕES MATEMÁTICAS

A seguir são apresentados os resultados para os testes com as funções matemáticas de benchmark.

Tabela A.1: Resultados SIAMEP ZDT1, ZDT2, ZDT3

ZTD1		ZTD2		ZTD3	
F1(x)	F2(x)	F1(x)	F2(x)	F1(x)	F2(x)
0,00000	1,00000	0,00000	1,00000	0,00000	1,00000
0,00154	0,96073	0,04486	0,99799	0,00414	0,93515
0,00690	0,91695	0,07749	0,99400	0,01323	0,87965
0,01453	0,87946	0,10936	0,98804	0,02261	0,83489
0,02474	0,84271	0,12893	0,98338	0,03417	0,78511
0,03480	0,81346	0,16597	0,97245	0,05094	0,72338
0,04573	0,78615	0,19798	0,96080	0,08400	0,66971
0,06172	0,75156	0,22531	0,94924	0,18226	0,66949
0,07857	0,71969	0,25693	0,93399	0,18972	0,62464
0,09501	0,69176	0,28830	0,91688	0,19684	0,57587
0,11164	0,66588	0,32486	0,89446	0,20229	0,53570
0,13088	0,63823	0,36240	0,86867	0,20802	0,49202
0,15050	0,61206	0,40195	0,83843	0,21216	0,46030
0,18834	0,56602	0,43658	0,80940	0,21907	0,40842
0,22533	0,52531	0,47341	0,77588	0,22559	0,36260
0,27737	0,47335	0,52504	0,72433	0,23150	0,32537
0,30810	0,44494	0,55227	0,69500	0,23815	0,29014
0,34005	0,41686	0,58527	0,65746	0,25748	0,24217
0,36862	0,39286	0,60620	0,63253	0,40944	0,24051
0,41020	0,35953	0,62886	0,60454	0,41350	0,18683
0,44221	0,33501	0,64998	0,57752	0,41775	0,13262
0,46754	0,31623	0,67604	0,54296	0,42194	0,08205
0,49895	0,29364	0,69985	0,51021	0,42898	0,00622
0,53268	0,27015	0,72614	0,47272	0,43189	-0,02106
0,55357	0,25597	0,75447	0,43078	0,43923	-0,07708
0,58264	0,23669	0,77531	0,39890	0,45347	-0,12418
0,60712	0,22082	0,79523	0,36761	0,61870	-0,12946
0,63272	0,20456	0,81100	0,34228	0,62173	-0,18067
0,65823	0,18869	0,82483	0,31966	0,62488	-0,23070
0,68943	0,16968	0,84334	0,28877	0,62843	-0,28227
0,73012	0,14553	0,85812	0,26364	0,63373	-0,34876
0,75956	0,12847	0,87432	0,23556	0,63995	-0,40828
0,78410	0,11451	0,89304	0,20247	0,65216	-0,45822
0,81498	0,09724	0,90783	0,17585	0,82529	-0,49726
0,84128	0,08279	0,92089	0,15196	0,82710	-0,53164
0,87730	0,06336	0,93520	0,12540	0,83027	-0,58708
0,90851	0,04684	0,95047	0,09660	0,83391	-0,64276
0,94345	0,02868	0,96597	0,06690	0,83751	-0,68902
0,97761	0,01126	0,98631	0,02719	0,84223	-0,73495
1,00000	0,00000	1,00000	0,00000	0,85214	-0,77333

Tabela A.2: Resultados SIAMEP ZDT4 e ZDT6

ZTD4		ZTD6	
F1(x)	**F2(x)**	**F1(x)**	**F2(x)**
0,00000	1,00071	0,28079	0,92116
0,00221	0,95383	0,30373	0,90775
0,00563	0,92579	0,32142	0,89669
0,01085	0,89659	0,33960	0,88467
0,01807	0,86639	0,35841	0,87154
0,02705	0,83657	0,38047	0,85524
0,03777	0,80633	0,39563	0,84347
0,05164	0,77415	0,41377	0,82879
0,06658	0,74299	0,43925	0,80706
0,08383	0,71110	0,46154	0,78698
0,10634	0,67461	0,48182	0,76785
0,12942	0,64114	0,50083	0,74916
0,14836	0,61537	0,51997	0,72963
0,17457	0,58277	0,53892	0,70956
0,20037	0,55327	0,55759	0,68909
0,23020	0,52132	0,58146	0,66190
0,27136	0,48073	0,60295	0,63646
0,31144	0,44305	0,62525	0,60907
0,34591	0,41313	0,65422	0,57199
0,36990	0,39304	0,67536	0,54388
0,39499	0,37204	0,69528	0,51658
0,42500	0,35027	0,71400	0,49021
0,45355	0,32717	0,72786	0,47021
0,48371	0,30538	0,74922	0,43867
0,51698	0,28299	0,76801	0,41016
0,56095	0,25248	0,78351	0,38612
0,59582	0,22859	0,79912	0,36140
0,63211	0,20563	0,81448	0,33662
0,65566	0,19085	0,83161	0,30842
0,69323	0,16818	0,84940	0,27853
0,72325	0,15001	0,86902	0,24481
0,74870	0,13513	0,88269	0,22086
0,78467	0,11471	0,89954	0,19083
0,82185	0,09417	0,91433	0,16401
0,84675	0,08021	0,93147	0,13236
0,87795	0,06434	0,94493	0,10711
0,90462	0,04962	0,95910	0,08012
0,94725	0,02737	0,97960	0,04039
0,97310	0,01402	0,98782	0,02421
1,00000	0,00112	1,00000	0,00001

Tabela A.3: Resultados SIAMEP CONSTR_EX e TNK

CONSTR_EX		TNK	
F1(x)	**F2(x)**	**F1(x)**	**F2(x)**
0,38926	8,99269	0,04221	1,03832
0,39440	8,82706	0,06662	1,02597
0,40082	8,48643	0,09561	1,00422
0,41131	8,24055	0,11744	0,98200
0,42225	7,89807	0,13958	0,95589
0,42548	7,49202	0,16753	0,93656
0,43090	7,39664	0,19881	0,92909
0,43853	6,98469	0,44855	0,92803
0,44177	6,88609	0,46269	0,90117
0,45351	6,46034	0,47329	0,88167
0,46864	6,29664	0,48860	0,84373
0,47763	5,95557	0,50789	0,82086
0,47841	5,70674	0,52378	0,79715
0,48623	5,41807	0,54944	0,78273
0,49384	5,24397	0,58529	0,77400
0,50262	4,94456	0,65633	0,77265
0,50985	4,77672	0,69193	0,76860
0,52190	4,42536	0,72209	0,75767
0,53890	4,01207	0,75321	0,74102
0,55231	3,71230	0,76071	0,71432
0,56591	3,41458	0,77354	0,63281
0,59019	3,02708	0,77365	0,62187
0,61339	2,75488	0,77473	0,59594
0,62001	2,30313	0,77687	0,56029
0,63946	2,10967	0,78292	0,54330
0,65326	1,74961	0,80672	0,50845
0,67078	1,49081	0,83786	0,50669
0,69793	1,43280	0,85435	0,50384
0,72215	1,38475	0,87588	0,47879
0,74721	1,33832	0,90182	0,47138
0,77449	1,29117	0,92575	0,44982
0,80125	1,24805	0,92959	0,20642
0,83152	1,20262	0,93619	0,19678
0,85929	1,16376	0,94038	0,16646
0,88988	1,12375	0,96249	0,15181
0,90992	1,09900	0,97196	0,12912
0,92851	1,07700	0,98285	0,11059
0,95617	1,04584	1,00910	0,08635
0,97972	1,02070	1,03052	0,06749
1,00000	1,00000	1,03942	0,04305