

**Thiago Fontes Santos**

*Operador para distribuição de soluções para  
algoritmos evolutivos via adaptação da matriz  
de covariância*

Belo Horizonte - MG, Brasil

02 de março de 2012

**Thiago Fontes Santos**

*Operador para distribuição de soluções para  
algoritmos evolutivos via adaptação da matriz  
de covariância*

Esta tese foi apresentado como parte dos  
requisitos necessários para a obtenção do  
Grau de Doutor em Matemática.

Orientador:

Dr. Ricardo Takahashi (UFMG)

INSTITUTO DE CIÊNCIAS EXATAS  
DEPARTAMENTO DE MATEMÁTICA  
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Belo Horizonte - MG, Brasil

02 de março de 2012

ATA DA QUADRAGÉSIMA DEFESA DE TESE DO ALUNO THIAGO FONTES SANTOS, REGULARMENTE MATRICULADO NO PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE MINAS GERAIS, REALIZADA DIA 02 DE MARÇO DE 2012.

Aos dois dias do mês de março de 2012, às 13h00, na Sala 3060, reuniram-se os professores abaixo relacionados, formando a Comissão Examinadora homologada pelo Colegiado do Programa de Pós-Graduação em Matemática, para julgar a defesa de tese do aluno **Thiago Fontes Santos**, intitulada: "*Operador para distribuição de soluções para algoritmos evolutivos via adaptação da matriz de covariância*", requisito final para obtenção do Grau de doutor em Matemática. Abrindo a sessão, o Senhor Presidente da Comissão, Prof. Ricardo Hiroshi Caldeira Takahashi, após dar conhecimento aos presentes o teor das normas regulamentares do trabalho final, passou a palavra ao aluno para apresentação de seu trabalho. Seguiu-se a arguição pelos examinadores com a respectiva defesa do aluno. Após a defesa, os membros da banca examinadora reuniram-se sem a presença do aluno e do público, para julgamento e expedição do resultado final. Foi atribuída a seguinte indicação: o aluno foi considerado aprovado, por unanimidade. O resultado final foi comunicado publicamente ao aluno pelo Senhor Presidente da Comissão. Nada mais havendo a tratar, o Presidente encerrou a reunião e lavrou a presente ata, que será assinada por todos os membros participantes da banca examinadora. Belo Horizonte, 02 de março de 2012.



**PROF. RICARDO HIROSHI CALDEIRA TAKAHASHI**  
Orientador - (UFMG)



**PROF. FELIPE CAMPELO FRANÇA PINTO**  
Examinador - (Engenharia Elétrica / UFMG)



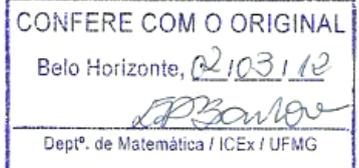
**PROF. LUIZ HENRIQUE DUCZMAL**  
Examinador - (Estatística / UFMG)



**PROF. ALEXANDRE CLÁUDIO BOTAZZO DELBEM**  
Examinador - (ICMC / USP)



**PROF. HÉLIO JOSÉ CORRÊA BARBOSA**  
Examinador - (LNCC)



# *Resumo*

Este trabalho propõe uma opção de algoritmo de otimização que funciona bem em qualquer quantidade de funções objetivo. Nosso algoritmo é baseado num modelo estocástico apresentado em 2002 por Schaffler *et al.* Devido à deficiência deste algoritmo em distribuir as soluções ao longo do conjunto Pareto-ótimo, apresentamos um operador que consegue resolver esse problema. Como usamos métodos de direção de busca para otimização multiobjetivo, o custo computacional é relativamente baixo.

**Palavras-chave:** Otimização multiobjetivo, algoritmos evolutivos.

# *Abstract*

This paper proposes an algorithm that works well on any number of objective functions. Our algorithm is based on a stochastic model presented in 2002 by Stefan Schaffler. Due to the deficiency of distributing this algorithm the solutions along the Pareto-optimal set, we present an operator which can solve this problem. Since we use [completar o ingles aqui] gradient method for multiobjective optimization, the cost computational is relatively low.

**Keywords:** Multiobjective optimization, evolutionary algorithms.

# *Agradecimentos*

Agradeço a Deus por ter me proporcionado mais esta realização, dando-me coragem necessária para enfrentar as ocasiões difíceis e à minha família pelo apoio e o incentivo durante todo esse período que estive distante. Em particular, aproveito a oportunidade para externar o quanto amo minha irmã Shirleny Santos, que ouviu várias vezes desabaços meus em relação a problemas diversos da vida.

Ao departamento de matemática da Universidade Federal de Ouro Preto por ter sido flexível com a situação de um docente em estudo numa outra instituição.

Ao professor Ricardo Takahashi pela amizade, orientação, paciência e benevolência durante todo o doutorado.

Ao professor Fernando Oliveira pela orientação acadêmica no início do doutorado.

À Elizabeth Wanner por ter me ajudando nos meus primeiros passos no *MATLAB*.

Ao amigo Anderson Duarte por ter dado várias sugestões no final da elaboração desta tese e pelo companheirismo no mesmo local de trabalho.

Ao meu grande amigo Rodrigo Couto, por trocarmos por diversas vezes produtivas conversas em matemática. Aproveito e digo que amigo como esse está difícil de encontrar.

Ao meu amigo Sebastião Xavier, que me suportou dividindo sala na Universidade Federal de Ouro Preto.

Aos meus amigos Gilberto Cuzzuol, Geraldo César e Gil Fidelix.

A todos meus amigos da Igreja Batista Getsemani, em particular a Nivton Campos e esposa e a Alex Passamani. Com estas pessoas, eu entendo as palavras do sábio Salomão ao dizer que existem amigos mais chegados que irmão.

A minha futura esposa, Elaine Costa, a quem dedico esta obra.

# *Sumário*

## **Lista de Figuras**

## **Lista de Tabelas**

<b>Introdução</b>	p. 13
<b>1 Motivação para o método</b>	p. 17
1.1 Proximidade ao conjunto Pareto-ótimo . . . . .	p. 17
1.2 Dimensão do espaço de objetivos . . . . .	p. 26
<b>2 Fundamentos teóricos</b>	p. 28
2.1 Definições e exemplos . . . . .	p. 28
2.2 Métodos de otimização irrestrita . . . . .	p. 32
2.2.1 Método de descida . . . . .	p. 33
2.2.2 Seção áurea . . . . .	p. 38
2.3 Otimização via equações diferenciais estocásticas . . . . .	p. 40
2.3.1 Parâmetro $\epsilon > 0$ . . . . .	p. 44
<b>3 Algoritmo de otimização SSW</b>	p. 46
3.1 Descrição . . . . .	p. 46
3.2 Experimentos . . . . .	p. 47
3.2.1 Problemas com 2 objetivos . . . . .	p. 49
3.2.2 Problemas com 3 objetivos . . . . .	p. 51
3.3 Resultados . . . . .	p. 54

<b>4</b>	<b>Um operador de mutação para regiões factíveis</b>	p.56
4.1	Preliminares . . . . .	p.57
4.2	Atualizando a Matriz de Covariância $C$ . . . . .	p.57
4.2.1	Gerando descendentes . . . . .	p.58
4.2.2	Classificação não-dominada . . . . .	p.58
4.2.3	"Crowding – distance" . . . . .	p.59
4.2.4	Relação $\prec_{c,A}$ . . . . .	p.59
4.2.5	Caminho de evolução $p_c$ . . . . .	p.60
4.3	Atualizando o tamanho de passo $\sigma$ . . . . .	p.61
4.3.1	Caminho de Evolução $p_\sigma$ . . . . .	p.62
4.4	Proposta de operador: ES-mut . . . . .	p.62
4.5	Experimentos . . . . .	p.65
<b>5</b>	<b>Um procedimento de busca local</b>	p.70
5.1	Busca local . . . . .	p.70
5.2	Seção áurea para problemas multiobjetivo . . . . .	p.71
<b>6</b>	<b>Análise de desempenho</b>	p.76
6.1	Medidas de avaliação . . . . .	p.77
6.1.1	Maximum Spread . . . . .	p.77
6.1.2	Contagem de esferas . . . . .	p.78
6.1.3	S-Metric . . . . .	p.79
6.2	Avaliação de desempenho . . . . .	p.80
6.2.1	Problemas com poucos objetivos . . . . .	p.81
6.2.2	Problemas com muitos objetivos . . . . .	p.82
6.3	Resultados . . . . .	p.92
6.4	Conclusões . . . . .	p.93

<b>Apêndice A – Alguns lemas técnicos</b>	p. 94
<b>Apêndice B – Distribuição normal multivariada <math>\mathcal{N}(\mu, C)</math></b>	p. 98
B.1 Relação da hessiana com a matriz de covariância . . . . .	p. 101
<b>Apêndice C – Método Numérico para EDE</b>	p. 102
<b>Referências Bibliográficas</b>	p. 105

# Lista de Figuras

1.1	Ilustração do cone de dominância a partir do ponto $x_d$ . . . . .	p. 18
1.2	Ilustração das retas $l_1$ e $l_2$ e a região $I_d$ . . . . .	p. 18
1.3	Gráfico da função $p(d)$ . . . . .	p. 20
1.4	Exemplo mostra os conjuntos $\Delta^1(e_1, e_2)$ , $S_{d, \delta}^1$ , $S_{e_1, r_1}^1$ , $S_{e_2, r_2}^1$ , $C_{d, \theta}$ , $H(d, \alpha, \delta)$ e os vetores $n_d$ e $x_d$ para $n = 2$ . . . . .	p. 22
1.5	O gráfico em coordenadas logarítmicas da função $p_n(d)$ para diferentes valores de $n$ . . . . .	p. 26
2.1	Exemplo de Pareto-dominância para o caso bi-objetivo. Os pontos pretos são os elementos Pareto-ótimos e pontos vermelhos são os dominados por eles. . . . .	p. 29
2.2	Gráfico com os valores das funções objetivo. . . . .	p. 31
2.3	Gráfico da fronteira Pareto-ótima. . . . .	p. 31
2.4	seção áurea do intervalo $[a, b]$ . . . . .	p. 39
3.1	Esquema de iteração do algoritmo SSW para o caso bi-objetivo $f(x) = (Objetivo1, Objetivo2)$ . . . . .	p. 47
3.2	Resultados dos algoritmos para o problema FON. . . . .	p. 49
3.3	Resultados dos algoritmos para o problema ZDT1. . . . .	p. 50
3.4	Resultados dos algoritmos para o problema ZDT2. . . . .	p. 51
3.5	Resultados dos algoritmos para o problema DTLZ1. . . . .	p. 52
3.6	Resultados dos algoritmos para o problema DTLZ2. . . . .	p. 53
3.7	Resultados dos algoritmos para o problema DTLZ5. . . . .	p. 54
4.1	Ilustração da geração dos descendentes, escolha dos melhores e determinação do vetor $x_{medio}$ em uma iteração para um problema bi-objetivo. . . . .	p. 63

4.2	Problema FON . . . . .	p.66
4.3	Problema ZDT1 . . . . .	p.67
4.4	Problema ZDT2 . . . . .	p.67
4.5	Problema DTLZ1 . . . . .	p.68
4.6	Problema DTLZ2 . . . . .	p.68
4.7	Problema DTLZ5 . . . . .	p.69
6.1	Contagem das esferas para o caso bi-objetivo. Note que temos 20 pontos no espaço de objetivos mas somente 9 esferas. . . . .	p.78
6.2	S-Metric para o caso bi-objetivo. Neste caso, calculamos a área da região dominada pelos pontos $p_1$ , $p_2$ e $p_3$ com ponto de referência $R$ (parte verde da imagem!). . . . .	p.79
6.3	Gráfico boxplot com 3 algoritmos. . . . .	p.81
6.4	Resultados da medida Maximum Spread para problemas bi-objetivo. . . . .	p.82
6.5	Resultados da medida Maximum Spread para problemas com três objetivos. . . . .	p.82
6.6	Resultados da medida Contagem de Esferas para problemas bi-objetivo. . . . .	p.83
6.7	Resultados da medida Contagem de Esferas para problemas com três objetivos. . . . .	p.83
6.8	Resultados da medida S-Metric para problemas bi-objetivo. . . . .	p.84
6.9	Resultados da medida S-Metric para problemas com três objetivos. . . . .	p.84
6.10	Impacto da quantidade de funções objetivo na relação de Pareto-dominância. . . . .	p.85
6.11	Resultados da medida Maximum Spread para problema DTLZ1. . . . .	p.87
6.12	Resultados da medida Maximum Spread para problema DTLZ2. . . . .	p.87
6.13	Resultados da medida Maximum Spread para problema DTLZ5. . . . .	p.88
6.14	Resultados da medida Contagem de Esferas para problema DTLZ1. . . . .	p.88
6.15	Resultados da medida Contagem de Esferas para problema DTLZ2. . . . .	p.89
6.16	Resultados da medida Contagem de Esferas para problema DTLZ5. . . . .	p.89

6.17	Resultados da medida S-metric para problema DTLZ1. . . . .	p.90
6.18	Resultados da medida S-metric para problema DTLZ2. . . . .	p.90
6.19	Resultados da medida S-metric para problema DTLZ5. . . . .	p.91
6.20	Gráfico com os valores de $\ q(\cdot)\ ^2$ médio, em escala logarítmica, para $m = 40$ . . . . .	p.92
B.1	Exemplo mostra a função densidade de probabilidade para uma distribuição normal bivariada com desvio padrão desigual. . . . .	p.100
C.1	Ilustração de uma iteração do método de Euler. . . . .	p.103

# *Lista de Tabelas*

1.1	Casos particulares de $p_n(d)$ . . . . .	p. 25
5.1	Operações de contração sobre o segmento-verdadeiro $[\alpha_1, \alpha_2]$ . . . . .	p. 73
5.2	Operações de contração sobre o segmento-verdadeiro $[\alpha_1, \alpha_2]$ . . . . .	p. 73
6.1	Valor médio de $\ q(x)\ ^2$ nas soluções do algoritmo NSGA-2 para o problema DTLZ1. . . . .	p. 90
6.2	Valor médio de $\ q(x)\ ^2$ nas soluções do algoritmo SSW-CMA para o problema DTLZ1 . . . . .	p. 91
6.3	Valor médio de $\ q(x)\ ^2$ nas soluções do algoritmo NSGA-2 para o problema DTLZ2 . . . . .	p. 91
6.4	Valor médio de $\ q(x)\ ^2$ nas soluções do algoritmo SSW-CMA para o problema DTLZ2 . . . . .	p. 91
6.5	Valor médio de $\ q(x)\ ^2$ nas soluções do algoritmo NSGA-2 para o problema DTLZ5 . . . . .	p. 91
6.6	Valor médio de $\ q(x)\ ^2$ nas soluções do algoritmo SSW-CMA para o problema DTLZ5 . . . . .	p. 92

# Introdução

No final do século XIX, o estudo dos fenômenos do equilíbrio econômico sugeriu a ideia da maximização simultânea de várias funções, que leva ao conceito de soluções eficientes. Nas palavras de Vilfredo Pareto [28]:

*"Vamos dizer que os membros de uma coletividade desfrutam máxima ofelimidade em uma determinada posição, quando é impossível encontrar uma maneira de mudar a partir dessa posição muito ligeiramente, de tal modo que a ofelimidade apreciada por cada um dos indivíduos aumenta, assim como para a coletividade. Ou seja, qualquer pequeno deslocamento a partir dessa posição, necessariamente, tem o efeito de aumentar a ofelimidade que beneficia determinados indivíduos, diminuindo o proveito dos demais."*

Essa ideia, livre de contexto e, no caso geral de encontrar as soluções eficientes de problemas de minimização ou maximização nos quais exista algum conflito entre as diversas funções-objetivo, deu origem ao estudo da otimização multiobjetivo.

Uma classe importante de algoritmos para lidar com esse tipo de problema é o conjunto de técnicas de otimização evolutiva multiobjetivo. O conceito de seleção natural é incorporado nesse tipo de técnica. Este grupo de algoritmos são chamados de evolutivos (AEs). Tais procedimentos, que realizam buscas aleatorizadas, são caracterizados pelo processamento simultâneo de um conjunto de tentativas de solução (a "população").

Recentemente, abordagens híbridas tem sido propostas a fim de tornar o AE mais eficiente, no sentido de solicitar um número menor de avaliações de função para encontrar indivíduos candidatos à solução, e mais efetivo, no sentido de identificar soluções com boas qualidades.

Na literatura, os algoritmos evolutivos híbridos são chamados de algoritmos meméticos ou algoritmos genéticos híbridos (AGH). Uma estrutura geral desse tipo de algoritmo é mostrada no algoritmo 1.

A operação de seleção é o processo de gerar uma nova população através de réplicas de alguns indivíduos e eliminação de outros. O operador de cruzamento combina

---

**Algoritmo 1:** Modelo padrão de AGH

---

```
1 Inicializa os parâmetros
2 Inicializa o população
3 repita
4   • operador de seleção
5   • operador de cruzamento
6   • operador de mutação
7   • operador de busca local
8 até critério de parada ser atingido;;
```

---

a informação de dois ou mais indivíduos para gerar outros. O operador de mutação utiliza informação contida em um indivíduo para, estocasticamente, gerar outro indivíduo. O operador de busca local faz melhorias em cada indivíduo a fim de encontrar uma nova solução eficiente.

Em problemas de otimização multiobjetivo ou vetorial, existe mais de uma solução para o problema. A relação de ordem para escolhe tais soluções é chamada de *Pareto-dominância*. Esta relação é parcial, pois existem vetores que são não-comparáveis. As melhores soluções são chamadas de soluções não-dominadas. Algo muito comum é usar um segundo critério para selecionar soluções não-dominadas, obtendo assim uma relação de ordem total. Um critério de desempate conhecido na literatura é o quanto cada solução contribui para a diversidade das soluções ou o que os autores chamam de "Crowding-distance"(cf. [13]).

Bem, até o momento, descrevemos em linhas gerais sobre o que trata os problemas de otimização vetorial, algoritmos para encontrar soluções e critérios de seleção. Agora, vamos focar sobre quais problemas estamos interessados em resolver.

O primeiro problema, é bem peculiar aos algoritmos de otimização baseados em Pareto-dominância é o o efeito da proximidade do conjunto Pareto-ótimo. Em [34], foi feito um calculo para mostrar esse efeito. O estudo mostrou que nas proximidades do conjunto Pareto-ótimo, torna-se exponencialmente mais difícil produzir melhorias na solução à medida que a solução se aproxima do conjunto-solução.

O segundo problema que estamos interessados, é sobre uma classe problemas que tem sido objeto de estudos nos últimos anos. São os chamados *Many-Objectives Optimization Problems*, ou na sua tradução livre *Problemas com muitos objetivos* (cf. [2, 20, 25, 29, 31, 34]). Em [20], alguns dificuldades característicos desse tipo de problema são relatadas, tais como:

1. Degradação da habilidade de busca dos algoritmos de otimização baseados em Pareto-dominância;
2. Crescimento exponencial do quantidade de soluções necessários para aproximação da fronteira Pareto-ótima inteira;
3. Dificuldade da visualização das soluções.

Com o aumento da quantidade de funções objetivo, quase todas as soluções são consideradas não-dominadas. Assim, a propriedade de convergência dos AEs é severamente comprometida. Esta dificuldade foi verificada através de comparações com múltiplas execuções de algoritmos de otimização em [22]. Além disso, como a fronteira Pareto-ótima é uma hiper-superfície no espaço de objetivos, o número de soluções necessários para aproximar cresce com proporcional ao crescimento do espaço de objetivos, ou seja, pode ser que precisemos de milhares de soluções não-dominadas para aproximar toda a fronteira Pareto-ótima. Uma estratégia já utilizada para tratar problemas com muitos objetivos seria reduzir a quantidade de objetivos (cf. [10, 12]).

Em [30], é apresentada uma proposta que pode ser usada para tratar os problemas citados acima. A técnica é baseada na simulação de uma equação diferencial estocástica. Um sistema dinâmico estocástico com estrutura Itô é definido, cujos os pontos de equilíbrio são soluções do problema de otimização. Uma direção de descida, calculada usando os gradientes das funções objetivos, foi usada nesse estudo com o objetivo de definir as trajetórias do sistema dinâmico. Dessa forma, os pontos Pareto-críticos de primeira ordem tornam-se os pontos de equilíbrio da parte determinística do sistema dinâmico estocástico. Além disso, é mostrado que qualquer pequena vizinhança arbitrariamente próxima de um ponto Pareto-crítico é visitado, quase-certamente, pelo sistema dinâmico. Em função das iniciais dos seus autores, vamos nos referir ao algoritmo apresentado em [30] por SSW. Algumas restrições são necessárias para a execução do algoritmo proposto. Para o calculo do gradientes das funções precisamos que tais sejam pelo menos de classe  $C^2$ , ou seja, que possuam derivadas parciais até a ordem 2 e estas sejam contínuas.

Em [34], é feita uma hibridização do algoritmo SSW com um procedimento de busca local apresentado em [36] afim de resolver a dificuldade de encontrar soluções nas proximidade do conjunto Pareto-ótimo.

Uma avaliação de desempenho do algoritmo SSW foi feito em [32], comparando os resultados com algoritmos evolutivos clássicos como NSGA2, ver em [13], e TPM

proposto em [35]. O estudo mostrou que o algoritmo SSW tem eficiência em encontrar soluções Pareto-ótimas, no entanto apresenta serias dificuldades na distribuição das soluções ao longo do conjunto Pareto-ótimo.

Nesta tese, procuraremos acoplar algum operador ao algoritmo SSW afim de resolver o problema do espalhamento das soluções por toda a extensão do conjunto Pareto-ótimo. As ideias que vamos utilizar é baseado em estratégias evolutivas. A proposta será usar baseada na adaptação da matriz de covariâncias (CMA-ES) feito para problemas mono-objetivos apresentado em [15]. Posteriormente, o CMA-ES foi estendido para problemas multiobjetivos (cf. [18]). Ao final da construção, veremos que nosso operador pode ser incorporado a qualquer algoritmo. Por conseguinte, teremos um bom algoritmo com boa propriedade de convergência e boa distribuição das soluções, funcionando para problemas com qualquer quantidade de funções objetivo.

Esta tese está estruturada na seguinte maneira:

- No **capítulo 1** Motivações;
- No **capítulo 2** há noções gerais de otimização multiobjetivo;
- No **capítulo 3** apresentamos o algoritmo SSW;
- No **capítulo 4** construímos um operador de mutação e apresentamos um novo algoritmo baseado no SSW;
- No **capítulo 5** apresentamos um procedimento de busca local;
- No **capítulo 6** avaliamos o desempenho do novo algoritmo comparando com outros algoritmos evolutivos;

# 1 *Motivação para o método*

Discutiremos neste momento algumas situações em que os algoritmos baseados em Pareto-dominância\* perdem eficiência.

Entenda por eficiência a capacidade que o algoritmo tem de encontrar soluções Pareto-ótimas para um determinado problema.

Para esse tipo de algoritmo, como por exemplo NSGA2, ver por exemplo [13], podemos citar dois problemas que são inerentes a eles:

1. Proximidade ao conjunto Pareto-ótimo;
2. Dimensão do espaço de objetivos.

## 1.1 Proximidade ao conjunto Pareto-ótimo

Vamos considerar o seguinte problema de minimização

$$\min_{x \in \mathbb{R}^2} f(x) = (f_1(x), f_2(x))$$

onde,

$$\begin{aligned} f_1(x) &= \|x - e_1\|_2^2 \\ f_2(x) &= \|x - e_2\|_2^2 \end{aligned} \tag{1.1}$$

e  $e_1 = (1, 0)$  e  $e_2 = (0, 1)$ .

O conjunto Pareto-ótimo deste problema é o conjunto

$$P = \{(x, y) \in \mathbb{R}_+^2; x + y = 1\}.$$

Queremos definir uma forma de medir o sucesso de um determinado ponto em encontrar um outro ponto que o melhore através de uma mutação com probabilidade

---

\*A definição de Pareto-dominância será apresentada no capítulo 2.

uniforme.

Dado um ponto da forma  $x_d := (d, d)$ , vamos ilustrar, ver a figura 1.1, as regiões do plano tais que melhoram  $f_1$ ,  $f_2$  e ambas.

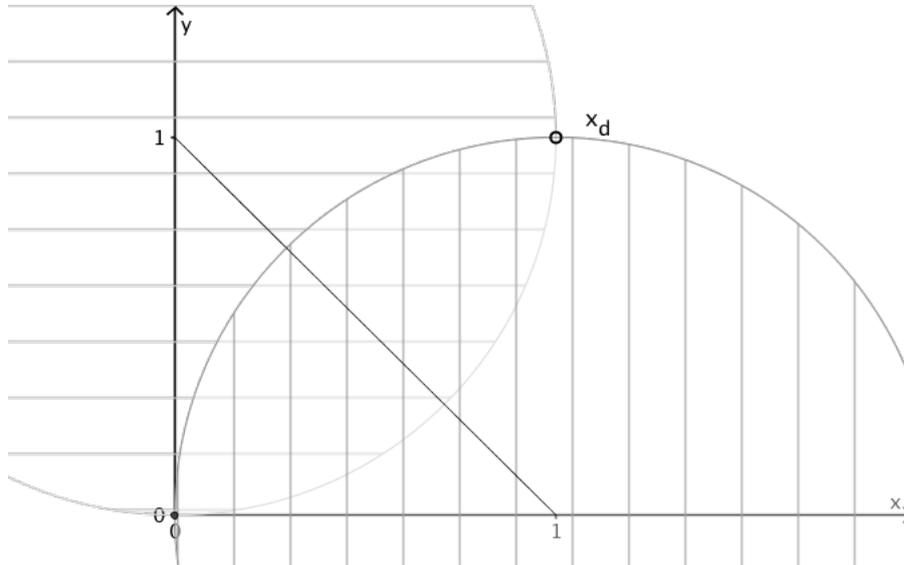


Figura 1.1: Ilustração do cone de dominância a partir do ponto  $x_d$ .

O círculo ( parte dele) com listras verticais é a região que melhora a função  $f_1$  a partir de  $x_d$  e o outro, com listras horizontais, a região que melhora a função  $f_2$ . A região de interseção desses círculos, formando quadrados, é a região que melhora as duas funções simultaneamente a partir de  $x_d$ . Vamos nos referir a esta região como  $I_d$ .

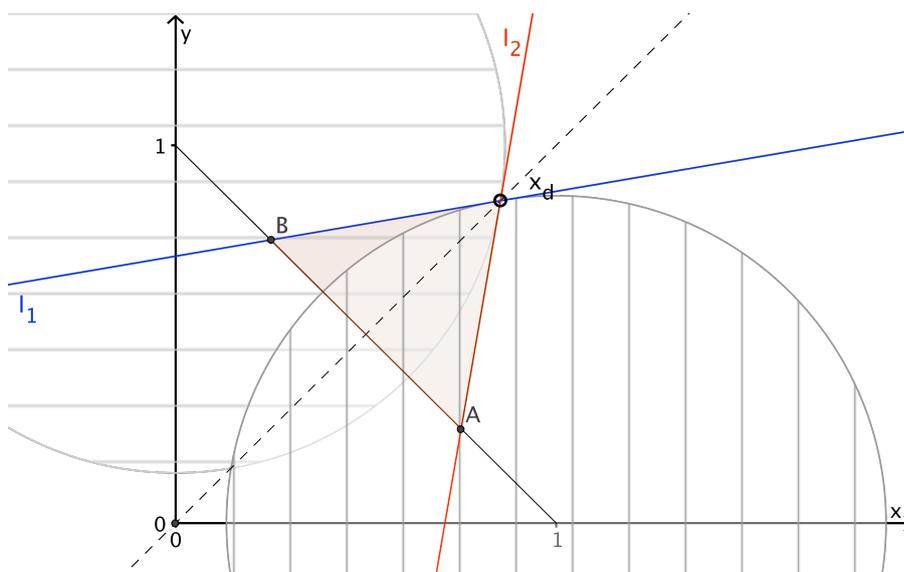


Figura 1.2: Ilustração das retas  $l_1$  e  $l_2$  e a região  $I_d$ .

Seja  $C_d$  o círculo com centro em  $x_d$  e raio 1. Denote por  $l_1$  e  $l_2$  as retas passando por

$x_d$  e perpendiculares a  $n_1 := x_d - e_1$  e  $n_2 := x_d - e_2$  respectivamente.

Na figura 1.2 é possível visualizar as retas  $l_1$  e  $l_2$  bem como a região  $I_d$ , sendo esta a interseção dos círculos  $C_1$  e  $C_2$  com o triângulo  $ABx_d$ . Tal região varia conforme fazendo o ponto  $x_d$  pela reta  $y = x$  que está representa em pontilhado.

**Observação 1.** *Se denotarmos por  $m_1$  e  $m_2$  as inclinações das retas  $l_1$  e  $l_2$  respectivamente, é fácil ver que:*

- $m_1, m_2 \rightarrow -1$  quando  $d \rightarrow \infty$ , ou seja, para valores de  $d$  arbitrariamente grandes, as retas  $l_1$  e  $l_2$  são iguais e por serem perpendiculares à reta  $y = x$ , dividem o círculo  $C_d$  na metade.
- $m_1, m_2 \rightarrow 1$  quando  $d \rightarrow 1/2$ , ou seja, para valores de  $d$  arbitrariamente próximos de  $\frac{1}{2}$ , as retas  $l_1$  e  $l_2$  são iguais à reta  $y = x$  e interceptam  $C_d \cap I_d$  no mesmo ponto.

Assim, vamos definir como taxa de sucesso de se obter a partir de  $x_d$  um ponto que melhore as funções simultaneamente pela razão:

$$p(d) := \frac{|t_1^d - t_2^d|}{2\pi}$$

onde  $t_i^d$ ,  $i = 1, 2$ , é o ponto de interseção do segmento de reta de  $l_i$  com  $C_d \cap I_d$ .

Isto é simplesmente a razão do comprimento de arco de  $t_1^d$  a  $t_2^d$  dividido pelo comprimento total que é  $2\pi$ . A figura 1.3 mostra o comportamento de  $p(d)$ .

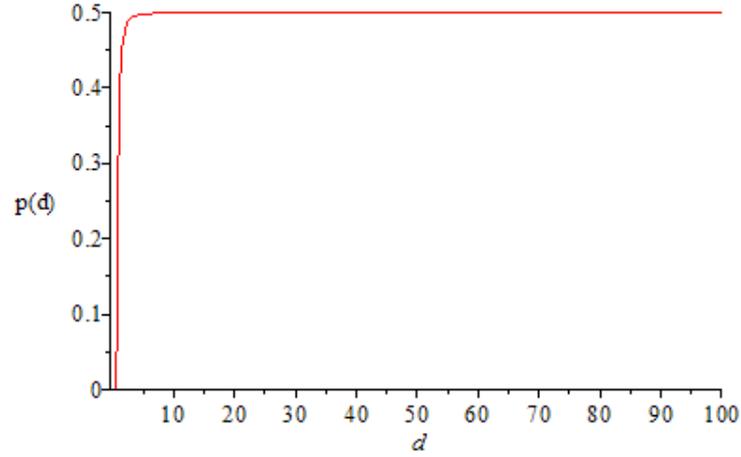
Pela observação 1, podemos dizer que:

- $\lim_{d \rightarrow \infty} p(d) = \frac{1}{2}$ , ou seja, para pontos arbitrariamente distantes do conjunto Pareto-ótimo temos 50% de chance de obter um novo ponto que melhore  $x_d$ ;
- $\lim_{d \rightarrow 1/2} p(d) = 0$ , ou seja, para pontos arbitrariamente próximos do conjunto Pareto-ótimo a chance de obter um novo ponto que melhore  $x_d$  é zero.

Para mostrar esse efeito em outras quantidades de funções objetivo, vamos considerar o problema 1.1 generalizado

$$\min_{x \in \mathbb{R}^n} f(x) = (f_1(x), f_2(x), \dots, f_n(x)) \quad (1.2)$$

onde cada  $f_i(x) := \|x - e_i\|$  em que  $e_i$  é o vetor de  $\mathbb{R}^n$  cujas coordenadas são todas nulas exceto na  $i$ -ésima sendo esta 1.

Figura 1.3: Gráfico da função  $p(d)$ .

O conjunto Pareto-ótimo deste problema, ver por exemplo [31], é o  $(n-1)$ -simplexo  $\Delta^{n-1}$  definido por

$$\left\{ \sum_{i=1}^n \lambda_i e_i : \lambda \in [0,1]^n \text{ and } \sum_{i=1}^n \lambda_i = 1 \right\}. \quad (1.3)$$

Consideremos o ponto em  $\mathbb{R}^n$  cujas coordenadas são  $x_d = (d, d, \dots, d)$ . Denotemos por  $S_{d,\delta}^{n-1} \subset \mathbb{R}^n$  a esfera com centro em  $x_d$  e raio  $\delta > 0$ . Seja  $C_{d,\theta} \subset \mathbb{R}^n$  um cone com vértice em  $x_d$  e ângulo interno  $2\theta$  que contém os pontos que dominam  $x_d$ . Observe que se  $(x_1, x_2, \dots, x_n) \in C_{d,\theta}$  então

$$(x_n - d)^2 = \sum_{i=1}^{n-1} \frac{(x_i - d)^2}{\alpha^2}$$

onde  $\frac{1}{\alpha} = \tan(\pi/2 - \theta)$ .

**Observação 2.** Como  $2\theta$  é o ângulo interno do cone  $C_{d,\theta}$  então  $\theta$  é o ângulo entre os vetores  $x_d$  e  $n_d \in T_p S_{e_1,r}^{n-1}$ , onde  $r = \|x_d - e_1\|$ . De fato, nós podemos tomar  $n_d$  pertencente a qualquer plano tangente à  $S_{e_i,r_i}^{n-1}$ , com  $r_i = \|x_d - e_i\|$ , no ponto  $x_d$ . Note que a superfície  $S_{e_i,r_i}^{n-1}$  contém os pontos que melhoram as funções  $f_i$ 's, definidas em (??), a partir de  $x_d$ .

Para determinar o vetor  $n_d$ , observe que a superfície  $S_{e_1,r}^{n-1}$  pode ser vista como uma superfície de nível. Basta considerar a função

$$F(x) = f_1(x) - \|x_d - e_1\|^2, \quad (1.4)$$

e facilmente ver que  $S_{e_1,r}^{n-1} = F^{-1}(0)$ . Portanto, o plano tangente a  $S_{e_1,r}^{n-1}$  no ponto  $x_d$  é o complemento ortogonal da reta determinada pelo vetor  $\nabla F(x_d)$ . Como  $\nabla F(x_d) = \nabla f_1(x_d) = (2(d-1), \underbrace{2d, \dots, 2d}_{(x_2, \dots, x_n)})$

então podemos tomar

$$n_d = (\underbrace{d, \dots, d}_{(x_1, \dots, x_{n-1})}, -((n-1)d-1)). \quad (1.5)$$

Claro que  $n_d$  pertence ao complemento ortogonal da reta determinada pelo vetor  $\nabla F(x_d)$  pois

$$\begin{aligned} \nabla F(x_d) \cdot n_d^T &= 2(d(d-1) + (n-2)d^2 - \\ &\quad d((n-1)d-1)) \\ &= 2(d^2 - d + (n-2)d^2 - \\ &\quad d^2(n-1) + d) \\ &= 0 \end{aligned}$$

Pela definição de ângulo entre vetores, nós temos que

$$\begin{aligned} \cos(\theta) &= \frac{n_d \cdot x_d^T}{\|n_d\| \|x_d\|} \\ &= \frac{(n-1)d^2 - d^2(n-1) + d}{\sqrt{(n-1)d^2 + ((n-1)d-1)^2} \sqrt{nd^2}} \\ &= \frac{d}{\sqrt{(n-1)d^2 + ((n-1)d-1)^2} \sqrt{nd}} \\ &= \frac{1}{\sqrt{n}} \frac{1}{\sqrt{(n-1)d^2 + ((n-1)d-1)^2}}. \end{aligned}$$

Na figura (1.4), podemos observar todos os conjuntos descritos anteriormente para o caso  $n = 2$ .

**Lema 1.1.** Com respeito as variáveis  $d$  e  $\theta$  citadas acima podemos afirmar que:

- a) Se  $d \rightarrow +\infty$  então  $\theta \rightarrow \pi/2$ ;
- b) Se  $d \rightarrow 1/n$  então  $\theta \rightarrow 0$ .

*Demonstração.* Com efeito, pelas observações anteriores, nós temos que

- Se  $d \rightarrow +\infty$  então  $\cos(\theta) \rightarrow 0$  e portanto  $\theta \rightarrow \pi/2$ ;

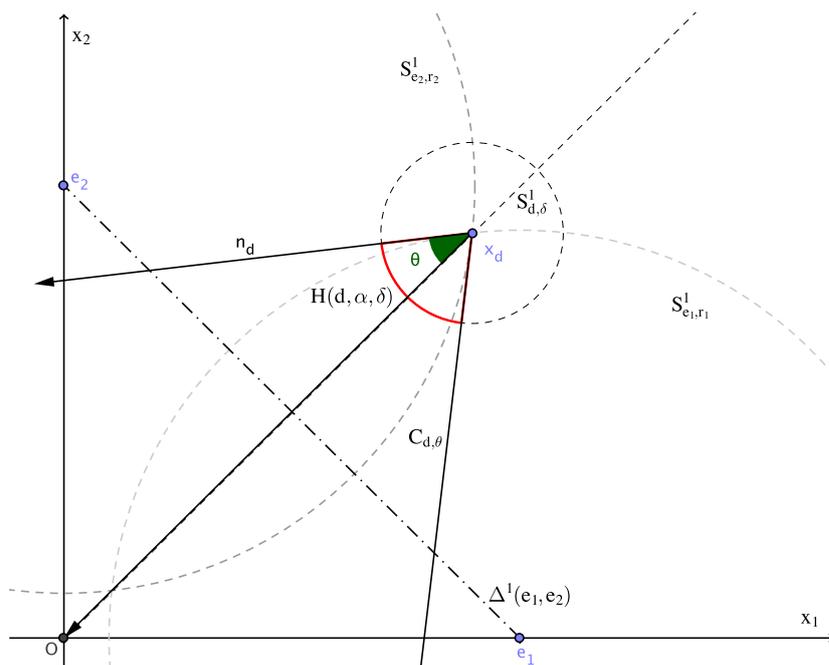


Figura 1.4: Exemplo mostra os conjuntos  $\Delta^1(e_1, e_2)$ ,  $S^1_{d, \delta}$ ,  $S^1_{e_1, r_1}$ ,  $S^1_{e_2, r_2}$ ,  $C_{d, \theta}$ ,  $H(d, \alpha, \delta)$  e os vetores  $n_d$  e  $x_d$  para  $n = 2$ .

- Se  $d \rightarrow 1/n$  então

$$\begin{aligned}
 \cos(\theta) &\rightarrow \frac{1}{\sqrt{n}} \frac{1}{\sqrt{(n-1)(1/n)^2 + ((n-1)(1/n) - 1)^2}} \\
 &= \frac{1}{\sqrt{n}} \frac{1}{\sqrt{(n-1)/n^2 + 1/n^2}} \\
 &= \frac{1}{\sqrt{n}} \frac{1}{\sqrt{1/n^2}} \\
 &= 1
 \end{aligned}$$

Neste caso,  $\theta \rightarrow 0$ .

□

**Lema 1.2.** Com respeito as variáveis  $d$  e  $\alpha$  citadas acima podemos afirmar que:

- Se  $d \rightarrow +\infty$  então  $\alpha \rightarrow +\infty$ ;
- Se  $d \rightarrow 1/n$  então  $\alpha \rightarrow 0$ .

*Demonstração.* Este resultado segue dos lemas anteriores. Como  $\alpha = 1/\tan(\pi/2 - \theta)$  então

- Se  $d \rightarrow +\infty$  então  $\theta \rightarrow \pi/2$  e portanto  $\alpha \rightarrow +\infty$ .

- Se  $d \rightarrow 1/n$  então  $\theta \rightarrow 0$  e portanto  $\alpha \rightarrow 0$ .

□

Seja  $H(d, \alpha, \delta)$  a hiper-calota esférica obtido da interseção dos conjuntos  $S_{d,\delta}^{n-1}$  e  $C_{d,\theta}$ , cuja altura e raio depende de  $(\alpha, \delta)$ . Tal dependência é determinada no lema (1.3).

**Lema 1.3.** Se  $H(d, \alpha, \delta)$  é como acima então sua altura é  $\frac{\delta^2}{1+\alpha^2}$  e seu raio é  $\frac{\delta^2}{1+\alpha^{-2}}$ .

*Demonstração.* Primeiro, nós iremos descrever a interseção  $S_{d,\delta}^{n-1} \cap C_{d,\theta}$ . Se  $x = (x_1, x_2, \dots, x_n) \in S_{d,\delta}^{n-1} \cap C_{d,\theta}$  então

$$(x_n - d)^2 = \sum_{i=1}^{n-1} \frac{(x_i - d)^2}{\alpha^2}$$

$$\sum_{i=1}^n \frac{(x_i - d)^2}{\alpha^2} = \delta^2$$

Combinando estas desigualdades, nós temos que

$$(x_n - d)^2 = \frac{\delta^2}{1+\alpha^2}$$

$$\sum_{i=1}^{n-1} (x_i - d)^2 = \frac{\delta^2}{1+\alpha^{-2}}$$

Logo, sua altura é  $\frac{\delta^2}{1+\alpha^2}$  e o raio é  $\frac{\delta^2}{1+\alpha^{-2}}$ . □

Vamos definir uma medida de probabilidade  $p_n(d)$  por

$$p_n(d) = \frac{A(H(d, \alpha, \delta))}{A(S_{d,\delta}^{n-1})}, \quad (1.6)$$

onde  $A(\cdot)$  calcula a área da hiper-superfície.

A função  $p_n(d)$  fornece uma maneira de encontrar a probabilidade de falha na busca em qualquer direção por meio de uma mutação com probabilidade uniforme de outro ponto dentro do cone. Iremos analisar dois comportamentos dessa função.

**Teorema 1.1.** Fixado  $n > 1$ , obtemos que:

- $\lim_{d \rightarrow +\infty} p_n(d) = 1/2$ ;
- $\lim_{d \rightarrow 1/n} p_n(d) = 0$ .

*Demonstração.* Pelo Lema (1.2), nós temos que:

- Se  $d \rightarrow +\infty$  então  $\alpha \rightarrow +\infty$ ;
- Se  $d \rightarrow 1/n$  então  $\alpha \rightarrow 0$ .

O lema (1.3) descreve a hiper-calota esférica  $H(d, \alpha, \delta)$  em relação da sua altura  $h$  e raio  $\sigma$ . Portanto,

1. Se  $d \rightarrow +\infty$  então  $h \rightarrow 0$  e  $\sigma = \delta$ ;
2. Se  $d \rightarrow 1/n$  então  $h \rightarrow \delta$  e  $\sigma = 0$ .

No caso (1), a hiper-calota esférica  $H(d, \alpha, \delta)$  representa a metade de  $S_{d,\delta}^{n-1}$  e portanto  $A(H(d, \alpha, \delta)) = A(S_{d,\delta}^{n-1})/2$ . Daí,  $\lim_{d \rightarrow +\infty} p_n(d) = 1/2$ . No outro caso,  $H(d, \alpha, \delta)$  é reduzido a um único ponto e portanto  $A(H(d, \alpha, \delta)) = 0$ . Logo,  $\lim_{d \rightarrow 1/n} p_n(d) = 0$ .  $\square$

Este importante resultado nos mostra que:

- i) Para valores arbitrariamente grande de  $d$ , o ponto  $x_d$  está distante do conjunto Pareto-ótimo, a probabilidade da geração de uma mutação dentro do cone torna-se próximo de 0.5;
- ii) Para valores de  $d$  próximos de  $1/n$ , o ponto  $x_d$  está próximo do conjunto Pareto-ótimo, a probabilidade da geração de uma mutação dentro do cone torna-se próximo de 0.

Seria interessante se pudéssemos descrever a função  $p_n(d)$  completamente. Na referência [27], os autores sugerem um forma de calcular  $A(S_{d,\delta}^{n-1})$  para qualquer  $n > 1$ .

**Proposição 1.1** ([27]). *Para qualquer  $n > 1$ ,*

$$A(S_{d,\delta}^{n-1}) = \frac{2\pi^{n/2}\delta^{n-1}}{\Gamma(n/2)},$$

onde  $\Gamma$  é a função Gamma.

Na referência [26] é feita uma relação entre  $A(S_{d,\delta}^{n-1})$  e  $A(H(d, \alpha, \delta))$ . Esta relação é o próximo resultado.

**Proposição 1.2** ([26]). Para qualquer  $n > 1$ ,

$$A(H(d, \alpha, \delta)) = (1/2)A(S_{d,\delta}^{n-1})I_{\sin^2(\theta)}\left(\frac{n-1}{2}, \frac{1}{2}\right),$$

onde  $I_z(p, q)$  é a função Beta Incompleta Regularizada.

A função  $I_z(p, q)$  é definida usando a função Beta como segue:

$$I_z(p, q) = \frac{B(z; p, q)}{B(p, q)}, \quad (1.7)$$

onde

$$B(z; p, q) = \int_0^z u^{p-1}(1-u)^{q-1} du$$

$$B(p, q) = \int_0^1 u^{p-1}(1-u)^{q-1} du$$

Alguns casos especiais de  $I_{\sin^2(\theta)}\left(\frac{n-1}{2}, \frac{1}{2}\right)$  foram exibidos em [3] e na tabela (1.1) contém alguns casos particulares.

Tabela 1.1: Casos particulares de  $p_n(d)$ .

n	$p_n(d)$
2	$\theta/\pi$
3	$(1 - \cos(\theta))/2$
4	$(2\theta - \sin(2\theta))/2\pi$
5	$(1 - (3/2)\cos(\theta) + (1/2)\cos^3(\theta))/2$

Agora, usando os dois resultados anteriores, obtemos uma importante descrição da função  $p_n(d)$ .

**Teorema 1.2.** A função  $p_n(d)$  definida em (1.6) pode ser reescrita por

$$p_n(d) = \frac{1}{2}I_{\sin^2(\theta)}\left(\frac{n-1}{2}, \frac{1}{2}\right), \quad (1.8)$$

onde  $\theta$  é o mesmo que foi definido na observação (2), ou seja,

$$\theta = \arccos\left(\frac{1}{\sqrt{n}} \frac{1}{\sqrt{(n-1)d^2 + ((n-1)d - 1)^2}}\right).$$

Este resultado generaliza o teorema (1.1), permitindo obter um comportamento geral da função  $p_n(d)$ .

A figura (1.5) mostra o comportamento de  $p_n(d)$  para alguns valores de  $n$ . Podemos notar que com o aumento da quantidade de funções objetivo, torna-se mais difícil obter boas soluções em torno do ponto  $x_d$ .

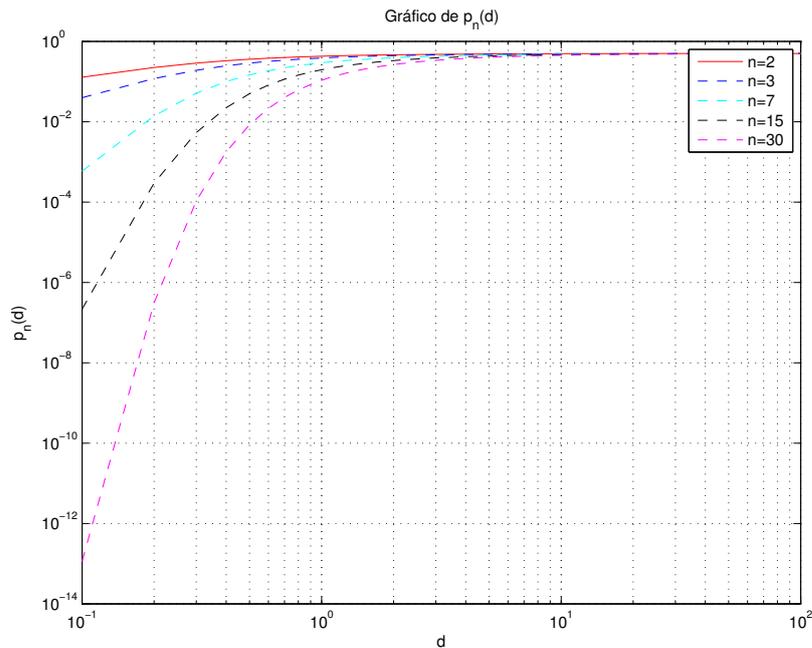


Figura 1.5: O gráfico em coordenadas logarítmicas da função  $p_n(d)$  para diferentes valores de  $n$ .

## 1.2 Dimensão do espaço de objetivos

Sobre a segunda dificuldade, o efeito do aumento da dimensão do espaço de objetivos, vamos ainda considerar o mesmo problema 1.2.

Observe que com respeito à origem  $\mathbf{0}$  do sistema de coordenadas, a região de  $\mathbb{R}^n$  de coordenadas todas positivas, que denotaremos por  $\mathbb{R}_+^n$ , inclui todos os pontos que são dominados por  $\mathbf{0}$ , enquanto as outras regiões têm os pontos que são não-dominados por  $\mathbf{0}$ .

Uma mutação com direções aleatórias com probabilidade uniforme irá apresentar a seguinte probabilidade de sucesso de estar em  $\mathbb{R}_+^n$ :

$$h(n) := \frac{1}{2^n} \quad (1.9)$$

É fácil ver que  $\lim_{n \rightarrow \infty} h(n) = 0$  ou seja, quando aumentamos a quantidade de funções objetivo, se torna mais difícil encontrarmos pontos em  $\mathbb{R}_+^n$  por meio de mutações com probabilidade uniforme.

Com a proposta vinda do algoritmo SSW-CMA, a ser apresentado neste trabalho, vemos uma possibilidade de um método baseado em direção de descida que não sofre esses efeitos e ainda mantém uma boa convergência e boa distribuição com um custo computacional relativamente baixo.

## 2 Fundamentos teóricos

Neste capítulo, vamos estabelecer os conceitos necessários sobre otimização multi-objetivo para o entendimento teórico dos assuntos abordados no presente trabalho.

Quando temos uma única função  $f(x)$  e queremos encontrar um valor mínimo ou máximo sobre um certo intervalo, existem testes analíticos, tais como aqueles envolvendo a derivada da função, que nos permitem dizer exatamente qual é esse valor.

Se a função é apenas contínua, existem teoremas clássicos que nos permitem dizer quando podemos encontrar tais tipos de valores (mínimo ou máximo), como por exemplo o teorema abaixo, cuja prova pode ser vista em [40]:

**Teorema 2.1** (Teorema de Weierstrass). *Sejam  $D \subset \mathbb{R}^n$  um conjunto fechado e limitado não-vazio e  $f : D \rightarrow \mathbb{R}$  uma função contínua.*

*Então o problema de minimizar e maximizar tem soluções globais em  $D$ .*

Agora, quando o assunto é minimizar (ou maximizar) duas ou mais funções simultaneamente esbarramos na primeira dificuldade, que é de comparar vetores. Note que até o momento, não temos uma relação de ordem que faça essa comparação.

### 2.1 Definições e exemplos

Sem perda de generalidade, vamos estudar o problema de minimizar duas ou mais funções simultaneamente. Este tipo de problema será chamado *problema de otimização multiobjetivo* ou simplesmente (POM) e escreveremos da forma que está abaixo:

$$\begin{cases} \min f(x) \\ \text{sujeito a: } x \in U, \end{cases} \quad (2.1)$$

em que  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $f(x) = (f_1(x), f_2(x), \dots, f_m(x))$  é uma função vetorial de classe  $C^2$  e  $U \subset \mathbb{R}^n$  é uma região factível de soluções definida por

$$U := \{x \in \mathbb{R}^n; g(x) \leq 0\},$$

onde  $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$  é um vetor de funções diferenciáveis.

**Definição 2.1** (Relação de ordem parcial). *Sejam  $u, v \in \mathbb{R}^n$ . O vetor  $u$  é menor que ou igual a  $v$ , e denotaremos por  $(u \leq v)$ , se  $u_i \leq v_i$  para todo  $i \in \{1, \dots, n\}$ .*

É fácil verificar que tal operação é uma ordem parcial para os vetores de  $\mathbb{R}^n$ , no sentido que podemos ter pontos que são não comparáveis. Na figura 2.1, os pontos em preto não são comparáveis.

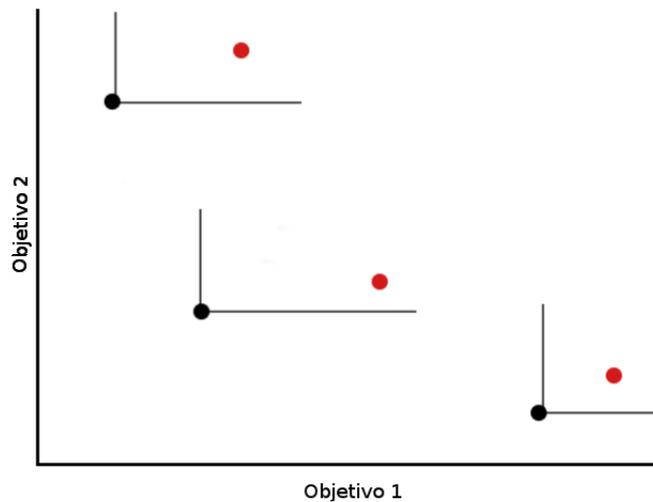


Figura 2.1: Exemplo de Pareto-dominância para o caso bi-objetivo. Os pontos pretos são os elementos Pareto-ótimos e pontos vermelhos são os dominados por eles.

**Definição 2.2.** *Considere o problema 2.1 definido acima. Um ponto  $\bar{x} \in U$  é dito ponto Pareto-ótimo se não existe  $y \in U$  tal que*

$$f(y) \neq f(\bar{x}) \text{ e } f(y) \leq f(\bar{x}). \quad (2.2)$$

O conjunto de todos esses pontos será denotado por  $U^*$ .

**Observação 3.** *Chamaremos esse conjunto de Pareto-ótimo, bem como seus elementos de pontos Pareto-ótimos. Algumas vezes vamos nos referir aos pontos de um conjunto Pareto-ótimo como as soluções eficientes do problema 2.1.*

**Observação 4.** *Quando  $U = \mathbb{R}^n$ , dizemos que o problema 2.1 é irrestrito.*

Uma condição necessária para eficiência foi proposta em [24] e será apresentada na sequência.

### Condições de Kuhn-Tucker para eficiência

Um elemento  $\hat{x} \in U$  satisfaz as condições necessárias de Kuhn-Tucker para eficiência se:

1. Todas as funções  $f_i$  do problema (2.1) são diferenciáveis;
2. Existem escalares  $\lambda_j \geq 0$ , com pelo menos um desigualdade estrita, tais que

$$\sum_{j=1}^m \lambda_j \nabla f_j(\hat{x}) = 0$$

$$\sum_{j=1}^m \lambda_j = 1.$$

Vejamos um exemplo simples com duas funções-objetivo onde queremos resolver o problema 2.1.

**Exemplo 2.1.1.** Considere as seguintes funções para o problema 2.1:

$$f_1(x) = x^2$$

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

Observando a figura 2.2, a intuição nos diz que o conjunto Pareto-ótimo é o intervalo  $[0,2]$ . Vamos constatar que de fato isso é verdade. Pelas condições de Kuhn-Tucker, se  $x^*$  é ponto Pareto-ótimo existem escalares  $\alpha_1$  e  $\alpha_2$  com  $\alpha_1 + \alpha_2 = 1$  tais que

$$\alpha_1 f_1'(x^*) + \alpha_2 f_2'(x^*) = 0$$

Daí obtemos que  $x^* = 2\alpha_2$ . Como  $\alpha_2 \in [0,1]$  então  $x^* \in [0,2]$ . Logo o conjunto Pareto-ótimo está contido no intervalo  $[0,2]$ .

Por outro lado, se  $x_0 \in [0,2]$  e existir um ponto  $y \in (2,\infty)$  tal que

$$f(y) \leq f(x_0)$$

então  $y^2 < x_0^2 \leq 4$  o que não pode acontecer pois  $y \in (2,\infty)$ . Portanto o conjunto das soluções eficientes é  $[0,2]$ .

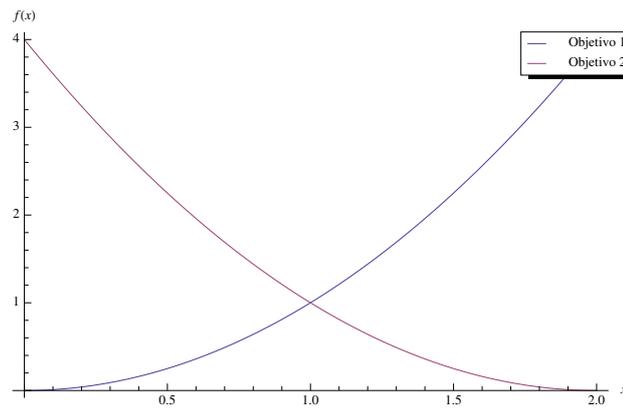


Figura 2.2: Gráfico com os valores das funções objetivo.

É muito comum exibir um gráfico, sempre que possível, com a imagem do conjunto Pareto-ótimo pela função dada no problema 2.1. Mostramos esse conjunto na figura 2.3.

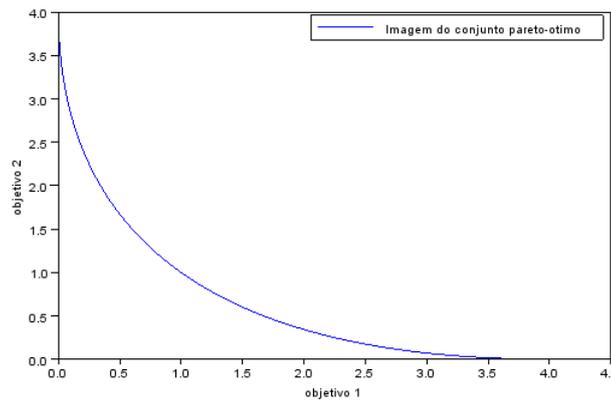


Figura 2.3: Gráfico da fronteira Pareto-ótima.

**Exemplo 2.1.2.** Neste problema, considere as seguintes funções objetivo  $f_i : \mathbb{R}^3 \rightarrow \mathbb{R}$

$$f_1(x) = (x_1 - 1)^2 + x_2^2 + x_3^2$$

$$f_2(x) = x_1^2 + (x_2 - 1)^2 + x_3$$

$$f_3(x) = x_1^2 + x_2^2 + (x_3 - 1)^2$$

Vamos verificar que o conjunto Pareto-ótimo deste problema é  $\Delta$ , onde

$$\Delta := \left\{ \sum_{i=1}^3 \lambda_i e_i : (\lambda_1, \lambda_2, \lambda_3) \in [0, 1]^3 \text{ com } \sum_{i=1}^3 \lambda_i = 1 \right\} e$$

$e_1 = (1,0,0)$ ,  $e_2 = (0,1,0)$  e  $e_3 = (0,0,1)$ .

Podemos reescrever cada função-objetivo por  $f_i(x) = \|x - e_i\|^2$ . Daí,

$$\nabla f_i(x) = 2(x - e_i).$$

Seja  $x \in \Delta$ . Existem  $\lambda_i \geq 0$  com  $\sum_{i=1}^3 \lambda_i = 1$  tais que  $x = \sum_{i=1}^3 \lambda_i e_i$ . Logo,

$$\begin{aligned} \sum_{i=1}^3 \lambda_i \nabla f_i(x) &= \sum_{i=1}^3 \lambda_i 2(x - e_i) \\ &= 2 \left( x \sum_{i=1}^3 \lambda_i - \sum_{i=1}^3 \lambda_i e_i \right) \\ &= 2(x - x) = 0 \end{aligned}$$

Isto nos diz que todo elemento de  $\Delta$  satisfaz a condição de Kuhn-Tucker para eficiência, ou seja,  $\Delta$  está contido no conjunto de soluções do nosso problema.

Por outro lado, seja  $x \in \Delta$  tal que exista  $y \in \mathbb{R}^3 \setminus \Delta$  tal que  $f(y) \leq f(x)$ . Vamos analisar as possibilidades para  $y$ .

Caso 1.  $y_1 + y_2 + y_3 > 1$ . Neste caso, existe pelo menos um índice  $i$  tal que  $y_i > 1$ . Digamos que seja  $i = 1$ . Daí,

$$1 < y_1 < y_1^2 < f_1(y) < f_1(x) < x_1 + x_2 + x_3 \quad (2.3)$$

o que não pode acontecer.

Caso 2.  $y_1 + y_2 + y_3 < 1$ . Neste caso, existe pelo menos um índice  $i$  tal que  $y_i < x_i$ . Digamos que seja  $i = 1$ . Daí,  $(y_1 - 1)^2 > (x_1 - 1)^2$  e portanto  $f_1(y) > f_1(x)$ , absurdo.

Logo, o conjunto Pareto-ótimo é  $\Delta$ .

## 2.2 Métodos de otimização irrestrita

Nesta seção discutiremos dois métodos para otimização sem restrição no espaço de variáveis. Essas mesmas técnicas serão usadas posteriormente nos algoritmos apresentados nesse trabalho.

### 2.2.1 Método de descida

Faremos uma pequena abordagem de como funciona para os casos em que estamos tentando minimizar apenas uma função e depois para o caso com duas ou mais funções simultaneamente.

Considere  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função dada. Uma das estratégias mais comuns para resolver o problema irrestrito

$$\min_{x \in \mathbb{R}^n} f(x)$$

será apresentada a seguir. Dada uma aproximação  $x^k \in \mathbb{R}^n$  da solução do problema, encontrarmos um ponto  $x^{k+1} \in \mathbb{R}^n$  tal que

$$f(x^{k+1}) < f(x^k).$$

Claro que podemos fazer isso de diversas maneiras. Uma boa estratégia é tomar uma direção  $d^k \in \mathbb{R}^n$  tal que  $f$  é decrescente a partir do ponto  $x^k$  nessa direção e calcular um comprimento de passo  $\alpha_k > 0$  que fornece um valor de  $f$  menor do que o ponto  $x^k$ ,

$$f(x^k + \alpha_k d^k) < f(x^k).$$

Deste modo, obtemos o iterando  $x^{k+1} = x^k + \alpha_k d^k$ . Repetimos o processo para o novo ponto  $x^{k+1}$ , etc.

Métodos baseados nessas ideias são chamados métodos de descida.

Uma questão quase que imediata a se perguntar é qual o tamanho de passo que podemos usar em cada problema e quais são as direções eficientes.

A fim de responder a primeira pergunta, vamos apresentar a *regra de Armijo*.

#### Regra de Armijo

Observe que queremos calcular um comprimento de passo que resulta em decréscimo suficiente da função  $f$  em relação ao valor  $f(x^k)$ .

Suponhamos que  $f$  seja diferenciável no ponto  $x^k$ . Fixamos os parâmetros  $\hat{\alpha} > 0$ ,  $\sigma > 0$ ,  $\theta \in (0, 1)$ . Tomamos,  $\alpha := \hat{\alpha}$ .

1. Verificamos se a desigualdade

$$f(x^k + \alpha d^k) < f(x^k) + \sigma \alpha \nabla f(x^k) \cdot (d^k)^T \quad (2.4)$$

se satisfaz ou não.

2. Se 2.4 não se satisfaz, tomamos  $\alpha := \theta \alpha$  e retornamos ao item anterior.

Caso contrário, aceitamos  $\alpha_k = \alpha$  como valor do comprimento de passo.

Em outras palavras,  $\alpha_k$  é o maior número entre todos os números de forma  $\alpha = \hat{\alpha} \theta^i$ ,  $i = 0, 1, 2, \dots$  que satisfaz a desigualdade 2.4.

Sobre o segundo questionamento, vamos apresentar o método do gradiente.

### Método do gradiente

Suponhamos que  $f$  seja diferenciável em  $\mathbb{R}^n$ . Falamos do método do gradiente quando a direção de descida  $d^k$  que se escolhe é o anti-gradiente da função  $f$  no ponto  $x^k$ , ou seja,

$$d^k = -\nabla f(x^k). \quad (2.5)$$

Observe que o gradiente aponta para uma direção segundo a qual a função  $f$  é crescente, pois se  $w = \nabla f(a)$  então

$$\frac{\partial f}{\partial w}(a) = \nabla f(a) \cdot (\nabla f(a))^T = \|\nabla f(a)\|_2^2 > 0. \quad (2.6)$$

Isto quer dizer que se  $\lambda(t)$  é um caminho diferenciável tal que  $\lambda(0) = a$  e  $\lambda'(0) = \nabla f(a)$ , então a função real

$$t \mapsto f(\lambda(t))$$

possui derivada positiva no ponto  $t = 0$ . Se supusermos que  $f$  e  $\lambda$  são de classe  $C^1$ , então a derivada de  $f \circ \lambda$  será ainda positiva em todos os pontos de um intervalo aberto com centro 0.

Um outro fato importante de se usar direções gradientes é que sua direção de decréscimo de  $f$  é mais rápida do que as outras. Isto quer dizer o seguinte: se  $v$  for um vetor tal que  $\|v\|_2 = \|\nabla f(a)\|_2$  então

$$\frac{\partial f}{\partial v}(a) \leq \frac{\partial f}{\partial (\nabla f(a))}(a).$$

De fato, pela desigualdade de Schwarz:

$$\frac{\partial f}{\partial v}(a) = \nabla f(a) \cdot v^T \leq \|\nabla f(a)\|_2 \|v\|_2 = \frac{\partial f}{\partial(\|\nabla f(a)\|_2)}(a).$$

Do ponto de vista da otimização, alguns métodos de minimização sem restrição de uma função objetivo duas vezes continuamente diferenciável  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  podem ser interpretados como soluções numéricas do problema de valor inicial (PVI):

$$\dot{x}(t) = -\nabla f(x(t)), \quad x(0) = x_0, \quad (\text{PVI})$$

As soluções  $x : [0, \infty] \rightarrow \mathbb{R}^n$  deste problema são o conjunto de pontos tais que, se  $\nabla f(x_0) \neq 0$  então

$$f(x(s)) \geq f(x(t)), \quad \text{para todo } 0 \leq s < t < \infty.$$

Vamos generalizar essa abordagem para função a objetivo  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . Daí, iremos formular um problema de valor inicial cuja a única solução é uma curva  $x : [0, \infty) \rightarrow \mathbb{R}^n$  formada de pontos dominados, ou seja,

$$f(x(s)) \geq f(x(t)) \text{ e } f(x(s)) \neq f(x(t)) \text{ para todo } 0 \leq s < t < \infty.$$

Para construção do nosso problema, primeiro vamos associar com  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $f(x) = (f_1(x), \dots, f_m(x))$ , ao seguinte problema de otimização quadrática (POQ(x)):

$$\min_{\alpha \in \mathbb{R}^m} \left\{ \left\| \sum_{i=1}^m \alpha_i \nabla f_i(x) \right\|_2^2 ; \alpha_i \geq 0, i = 1, \dots, m, \sum_{i=1}^m \alpha_i = 1 \right\} \quad (\text{POQ}(x))$$

onde  $\nabla f_i(x)$  representa o gradiente de  $f_i$ .

Algumas propriedades desse problema podem ser facilmente verificadas. As que precisamos estão nas duas proposições seguintes.

**Proposição 2.1.** *Para cada  $x \in \mathbb{R}^n$ , existe uma solução global de (POQ(x)).*

*Demonstração.* Considere o seguinte conjunto não-vazio

$$K = \left\{ (a_1, \dots, a_m) \in \mathbb{R}^m \mid a_i \geq 0 \text{ e } \sum_{i=1}^m a_i = 1 \right\}.$$

Observe que  $K$  é um conjunto fechado e limitado. De fato,

- $K$  é conjunto fechado pois  $K = f^{-1}(0)$  onde  $f : \mathbb{R}^m \rightarrow \mathbb{R}$  é a função contínua dada

$$\text{por } f(y) = 1 - \sum_{i=1}^m y_i.$$

- $K$  é Limitado pois se  $\alpha \in K$  então  $\|\alpha\|_2^2 = \sum_{i=1}^m (\alpha_i)^2 \leq \left(\sum_{i=1}^m \alpha_i\right)^2 = 1$ , ou seja,  $K \subset B(0;1)$ .\*

Como, para fixados  $y^1, \dots, y^m \in \mathbb{R}^n$ , a função  $h : K \rightarrow \mathbb{R}$  dada por

$$h(\alpha) = \left\| \sum_{i=1}^m \alpha_i y^i \right\|_2^2$$

é contínua, então pelo teorema de Weierstrass, o problema de minimizar a função  $h$  tem solução global em  $K$ . No caso do problema (POQ(x)), temos que cada  $y^i = \nabla f_i(x)$ .  $\square$

**Observação 5.** Como a função norma,  $f(x) = \|x\|$ , é estritamente convexa o problema em (POQ(x)) não pode ter mais de uma solução. Logo a função  $q(x) = \sum_{i=1}^m \widehat{\alpha}_i \nabla f_i(x)$  onde  $\widehat{\alpha}$  é uma solução de (POQ(x)) está bem definida.

Agora, precisamos identificar uma direção de descida para nossas funções objetivo. O próximo teorema nos dá a resposta.

**Teorema 2.2.1** ([30]). Seja  $q : \mathbb{R}^n \rightarrow \mathbb{R}^n$  definida por

$$q(x) = \sum_{i=1}^m \widehat{\alpha}_i \nabla f_i(x)$$

onde  $\widehat{\alpha}$  é uma solução de (POQ(x)). Então, ou  $q(x) = 0$  ou  $-q(x)$  é uma direção de descida para todas as funções objetivo  $f_1, \dots, f_m$  em  $x$ . Além disso, a função  $q$  é localmente Lipschitz contínua.

*Demonstração.* Considere o seguinte conjunto

$$K(x) = \left\{ \sum_{i=1}^m \alpha_i \nabla f_i(x), \alpha_i \geq 0, i = 1, \dots, m, \sum_{i=1}^m \alpha_i = 1 \right\}$$

e vamos assumir que  $0 \notin K(x)$  para todo  $x \in \mathbb{R}^n$  fixado. Queremos mostrar que  $q(x)^T v \geq 0$  para todo  $v \in K(x)$ . Suponha que exista um vetor  $v(x) \in K(x)$  com  $q(x)^T v(x) \leq 0$ . Então, obtemos as seguintes propriedades dos vetores  $\lambda(q(x) - v(x))$ ,  $0 \leq \lambda \leq 1$ :

- $v(x) + \lambda(q(x) - v(x)) \in K(x)$ , para todo  $0 \leq \lambda \leq 1$ ;

---

\*  $B(0;1)$  é a bola de centro na origem e raio 1.

- $q(x)^T(\lambda(q(x) - v(x))) > 0$ , para todo  $0 < \lambda \leq 1$ .

Seja  $\hat{\lambda}$  solução global do problema de minimização quadrática abaixo:

$$\min_{0 \leq \lambda \leq 1} \left\{ \|v(x) + \lambda(q(x) - v(x))\|_2^2 \right\}$$

Claro que  $\hat{\lambda}$  não pode ser 0 pois se fosse iria contrariar a definição de  $q(x)$ . Agora, vamos mostrar que  $\hat{\lambda}$  não pode ser 1. Suponha que seja. Então

$$\|q(x)\|_2^2 \leq \|v(x) + \lambda(q(x) - v(x))\|_2^2, \text{ para todo } 0 < \lambda \leq 1$$

Desenvolvendo a norma-2 no lado direito, obtemos que

$$(1 + \lambda)\|q(x)\|_2^2 \leq (1 - \lambda)\|v(x)\|_2^2 + 2\lambda q(x)^T v(x), \text{ para todo } 0 < \lambda \leq 1$$

Em particular para  $\lambda = 1$  temos que

$$\|q(x)\|_2^2 \leq q(x)^T v(x)$$

Como  $q(x)^T v(x) \leq 0$  então  $q(x) = 0$ , o que não pode ocorrer pois estamos supondo que  $0 \notin K(x)$ . Logo  $0 < \hat{\lambda} < 1$  e portanto

$$\|v(x) + \hat{\lambda}(q(x) - v(x))\|_2^2 < \|q(x)\|_2^2.$$

Desde que  $v(x) + \hat{\lambda}(q(x) - v(x)) \in K(x)$  obtemos uma contradição com a definição de  $q(x)$ . Segue então que

$$v(x)^T q(x) > 0, \text{ para todo } v(x) \in K(x)$$

Uma prova de que  $q(x)$  é localmente lipschitziana pode ser vista em [30].  $\square$

**Observação 6.** É claro que se  $x^* \in \mathbb{R}^n$  tal que  $q(x^*) = 0$  então  $x^*$  satisfaz as condições de Kuhn-Tucker para eficiência.

As propriedades da função  $q(x)$  nos permitem generalizar o (PVI) para problemas de otimização escalar para o seguinte problema de valor inicial para problemas de otimização vetorial:

$$\dot{x}(t) = -q(x(t)), \quad x(0) = x_0 \quad (2.7)$$

onde  $q$  é a mesma definida acima.

**Teorema 2.2.2.** Considere o problema de otimização vetorial (2.1) e o problema de valor inicial

correspondente (2.7) com  $q(x_0) \neq 0$ . Defina

$$R_{\leq} := \{x \in \mathbb{R}^n; f(x) \leq f(x_0)\},$$

e assumamos que  $R_{\leq}$  é limitado. Então existe uma única solução  $x : [0, \infty) \rightarrow \mathbb{R}^n$  tal que

$$f(x(s)) \geq f(x(t)) \text{ e } f(x(s)) \neq f(x(t)) \text{ para todo } 0 \leq s < t < \infty.$$

*Demonstração.* Essa demonstração pode ser encontrada em [30]. □

### 2.2.2 Seção áurea

Nesta seção falaremos de um dos métodos mais conhecidos de otimização unidimensional, a saber, método da seção áurea. Tal técnica deriva do tradicional método da bisseção, ver por exemplo a referência [11].

Recentemente, esta ideia foi generalizada para problemas de otimização multiobjetivo, ver a referência [36]. Deixaremos para comentar este caso no capítulo 5.

Para o que segue, vamos supor funções  $f : [a, b] \rightarrow \mathbb{R}$  unimodais, ou seja, possuem um único minimizador global  $x^*$  em  $[a, b]$  e além disso

1.  $f|_{[a, x^*]}$  é estritamente decrescente;
2.  $f|_{[x^*, b]}$  é estritamente crescente

Uma característica desse tipo de funções está no lema abaixo.

**Lema 2.1.** *Seja  $f : [a, b] \rightarrow \mathbb{R}$  uma função unimodal e  $x^*$  um minimizador global de  $f$  em  $[a, b]$ . Segue que para quaisquer  $y, z \in [a, b]$  tal que  $y < z$ , vale o seguinte:*

1. Se  $f(y) \leq f(z)$ , então  $x^* \in [a, z]$ .
2. Se  $f(y) \geq f(z)$ , então  $x^* \in [z, b]$ .

*Demonstração.* Faremos apenas a prova do item (1) pois o outro caso é análogo. Se  $f(y) \leq f(z)$  mas  $x^* > z$  então  $y < z < x^*$  e, portanto  $f(y) > f(z)$  pelo fato de que  $f$  é unimodal. Obtemos uma contradição com a hipótese. □

A seção áurea refere à partição do intervalo em duas partes tais que a relação do comprimento do intervalo original com o comprimento da parte maior é igual à relação do comprimento da parte maior com comprimento da parte menor.

Note que se  $y$  é um ponto de tal partição de  $[a, b]$  e está mais próximo de  $a$  do que de  $b$  então

$$\frac{b-a}{b-y} = \frac{b-y}{y-a}.$$

Daí,

$$y = y(a, b) = a + \frac{3 - \sqrt{5}}{2}(b - a) \approx a + 0.382(b - a).$$

Por outro lado, se  $z$  é um ponto e está mais próximo de  $b$  do que de  $a$  então

$$z = z(a, b) = a + \frac{\sqrt{5} - 1}{2}(b - a) \approx a + 0.618(b - a).$$

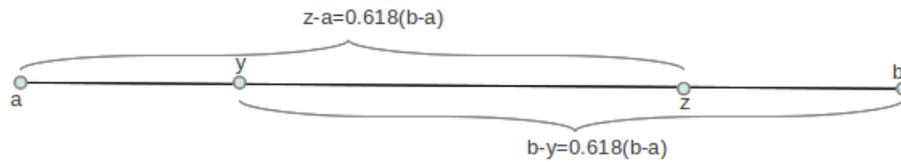


Figura 2.4: seção áurea do intervalo  $[a, b]$

Com um cálculo elementar, podemos afirmar o seguinte:

**Lema 2.2.** *Sejam  $y$  e  $z$  o menor e o maior ponto da partição de  $[a, b]$ , respectivamente. Então*

1.  $z - a = b - y = \frac{(\sqrt{5} - 1)(b - a)}{2}$ ;
2.  $y$  é o maior ponto da partição  $[a, z]$  e  $z$  é o menor ponto da partição  $[y, b]$ .

Como estamos trabalhando com funções unimodais, podemos dizer que  $x^* \in [a_{k+1}, b_{k+1}]$ . Além disso,

$$b_{k+1} - a_{k+1} = \frac{(\sqrt{5} - 1)(b_k - a_k)}{2} = \dots = \left(\frac{\sqrt{5} - 1}{2}\right)^k (b - a).$$

Logo, quanto maior for o número de passos, menor será nosso erro para encontrar o valor de  $x^*$ .

O método da seção áurea pode ser apresentado conforme algoritmo 2.

**Algoritmo 2:** Método seção áurea**Entrada:**  $I = [a, b]$ ,  $f(x)$ ,  $N$  é o número de passo a ser dado**Saída:** valor aproximado de  $x^*$ 

```

1  $a_1 = a;$ 
2  $b_1 = b;$ 
3  $y_1 = a_1 + 0.382(b_1 - a_1);$ 
4  $z_1 = a_1 + 0.618(b_1 - a_1);$ 
5  $k \leftarrow 1;$ 
6 repita
7   se  $f(y_k) \leq f(z_k)$  então
8      $a_{k+1} \leftarrow a_k;$ 
9      $b_{k+1} \leftarrow z_k;$ 
10     $y_{k+1} \leftarrow a_{k+1} + 0.382(b_{k+1} - a_{k+1});$ 
11     $z_{k+1} \leftarrow y_k;$ 
12  senão
13     $a_{k+1} \leftarrow y_k;$ 
14     $b_{k+1} \leftarrow b_k;$ 
15     $y_{k+1} \leftarrow z_k;$ 
16     $z_{k+1} \leftarrow a_{k+1} + 0.618(b_{k+1} - a_{k+1});$ 
17  fim
18   $k \leftarrow k + 1;$ 
19 até  $k < N + 1;$ 
20  $x^* \approx \frac{b_{k+1} + a_{k+1}}{2};$ 

```

## 2.3 Otimização via equações diferenciais estocásticas

Seja  $\Omega$  o conjunto de todas as funções contínuas  $z : [0, \infty) \rightarrow \mathbb{R}^n$ ,  $n \in \mathbb{N}$  fixo. Definamos a seguinte métrica  $d$  em  $\Omega$

$$d(w_1, w_2) = \sum_{n=1}^{\infty} \frac{1}{2^n} \min \left[ \max_{0 \leq t \leq n} \|w_1(t) - w_2(t)\|, 1 \right]$$

Seja  $\mathcal{B}(\Omega)$  a menor  $\sigma$ -álgebra contendo todos os abertos de  $\Omega$  na topologia definida pela métrica  $d$ . A partir de agora, nós usaremos somente o espaços de probabilidade  $(\Omega, \mathcal{B}(\Omega), \mathcal{W})$ , onde  $\mathcal{W}$  representa a medida de Wiener  $n$ -dimensional. Uma construção dessa medida pode ser vista na referência [4].

Seja  $\bar{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$  a compactificação de  $\mathbb{R}$  com  $0 \cdot \infty = 0 \cdot (-\infty) = \infty \cdot 0 = (-\infty) \cdot 0 = 0$  e  $\mathcal{B}(\bar{\mathbb{R}})$  a  $\sigma$ -álgebra de Borel de  $\bar{\mathbb{R}}$ , dado por

$$B \in \mathcal{B}(\bar{\mathbb{R}}) \Leftrightarrow (B \cap \mathbb{R}) \in \mathcal{B}(\mathbb{R})$$

Para o que segue, são necessárias algumas definições básicas da teoria da medida.

**Definição 2.3.** Uma função  $f : \Omega \rightarrow \bar{\mathbb{R}}$  é chamada função numérica.

**Definição 2.4.** Uma coleção  $\{X_t\}$  de variáveis aleatórias  $n$ -dimensionais,  $t \geq 0$ , definidas em  $\Omega$  será chamada de processo estocástico.

Em outras palavras, um processo estocástico é um conjunto indexado de variáveis aleatórias.

**Definição 2.5.** Seja  $\{X_t\}$  um processo estocástico. Para cada  $\omega \in \Omega$  fixo, a função  $X_\omega : [0, \infty) \rightarrow \mathbb{R}^n$  dada por  $X_\omega(t) := X_t(\omega)$  é chamada de caminho de  $\{X_t\}$ .

**Definição 2.6.** Um processo estocástico  $\{X_t\}$  será dito contínuo se cada caminho de  $X_t$  for uma função contínua.

**Definição 2.7.** Um processo  $\{X_t, t \geq 0\}$  é dito movimento Browniano se satisfaz as seguintes condições:

1.  $X_0 = 0$ ;
2. Para todo  $0 \leq t_1 < \dots < t_n$ , os incrementos  $X_{t_n} - X_{t_{n-1}}, \dots, X_{t_2} - X_{t_1}$  são variáveis aleatórias independentes;
3. Se  $0 \leq s < t$ , o incremento  $X_t - X_s$  tem a distribuição normal<sup>†</sup>  $\mathcal{N}(0, t - s)$ ;
4. O processo  $\{X_t\}$  tem trajetórias contínuas.

**Observação 7.** O processo estocástico  $\{B_t\}$  dado por  $B_t(\omega) := \omega(t)$  é um movimento Browniano canônico  $n$ -dimensional.

Na busca de um conjunto representativo de soluções de (2.1), analisaremos a seguinte classe de equações diferenciais estocásticas<sup>‡</sup> (EDE) de Itô, onde  $\{B_t\}$  é um movimento Browniano  $n$ -dimensional e  $q(x)$  é a função definida no teorema (2.2.1):

$$dX_t = -q(X_t)dt + \epsilon dB_t, \quad X_0 = x_0 \quad (2.8)$$

onde  $\epsilon > 0$  e  $x_0 \in \mathbb{R}^n$ .

Um processo estocástico  $X_t(\cdot)$  é solução de 2.8 se

$$X_t = x_0 - \int_0^t q(X_s)ds + \epsilon \int_0^t dB_s, \quad \forall t > 0., \quad (2.9)$$

<sup>†</sup>Ver informações sobre distribuição normal no apêndice B

<sup>‡</sup>Noções elementares sobre equações diferenciais estocásticas podem ser vistas em [4, 23].

onde  $\int_0^t dB_s = B_t(\cdot) - B_0(\cdot)$ .

É razoável sabermos se a equação (2.8) tem solução e se esta é única sobre alguma hipótese. Para este fim, vamos supor o seguinte:

**Hipótese 1.** *Existe um  $\epsilon > 0$  tal que*

$$x^T q(x) \geq \frac{1+n\epsilon^2}{2} \max(1, \|q(x)\|_2)$$

para todo  $x \in \mathbb{R}^n \setminus \{x \in \mathbb{R}^n; \|x\|_2 \leq r\}$  com algum  $r > 0$ , onde  $q$  está definida no teorema (2.2.1).

Esta hipótese descreve um comportamento especial para a função  $q(x)$ , e portanto para  $f(x)$ , fora da bola de centro na origem e raio  $r > 0$ . Durante a prova do próximo teorema ficará mais claro a necessidade da hipótese.

**Teorema 2.3.1** ([30]). *Considere (2.8). Para todo  $x_0 \in \mathbb{R}^n$  e todo  $\epsilon > 0$  no qual a hipótese (1) é satisfeita, nós temos o seguinte:*

1. *Existe um único processo estocástico  $\{X_t\}$  que é solução de (2.8);*
2. *Todos os caminhos de  $\{X_t\}$  são contínuos;*
3.  $X_0 \equiv x_0$ .

*Demonstração.* Seja  $k \in \mathbb{N}$  e considere  $q^{(k)} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  dado por

$$q^{(k)}(x) := \begin{cases} q(x), & \text{se } \|x - x_0\|_2 < k \\ q\left(\frac{k(x - x_0)}{\|x - x_0\|_2}\right), & \text{caso contrário.} \end{cases}$$

Pelo Lema(A.1),  $q^{(k)}$  é Lipschitz com constante de Lipschitz  $L^k$ . Vamos analisar a seguinte equação integral estocástica com  $t \geq 0$  e  $\epsilon > 0$  tal que a hipótese (1) é satisfeita:

$$X_{t,k}(w) = x_0 + \epsilon(B_t(w) - B_0(w)) - \int_0^t q^{(k)}(X_{\tau,k}(w))d\tau, \quad (2.10)$$

Pelo teorema (A.1), temos a existência, unicidade e soluções contínuas de  $X_{t,k}(w)$  de (2.10),  $t \geq 0$ . O calculo de  $X_{t,k}(w)$  é dado por teorema (A.1) c) com

$$h_0(\tau) = x_0 + \epsilon(B_\tau(w) - B_0(w)), \quad 0 \leq \tau \leq t.$$

O teorema (A.2) nos garante que  $X_{t,k}$  é um processo estocástico. É claro que  $X_{0,k} \equiv x_0$ .

Agora, vamos mostrar como a solução  $X_{t,k}$  de (2.10) se torna solução de (2.8). Vamos definir o seguinte tempo aleatório  $s_k : \Omega \rightarrow \bar{\mathbb{R}}$ :

$$s_k(w) := \begin{cases} \inf \{t \geq 0 \mid \|X_{t,k}(w) - x_0\|_2 > k\}, & \text{se } \{t \geq 0 \mid \|X_{t,k}(w) - x_0\|_2 \neq \emptyset\} \\ \infty, & \text{caso contrário.} \end{cases}$$

Por definição,  $q^{(k)}(x) = q(x)$  e portanto para  $k \in \mathbb{N}$ ,  $0 \leq t \leq s_k(w)$  temos que

$$X_{t,k}(w) = x_0 + \epsilon(B_t(w) - B_0(w)) - \int_0^t q(X_{\tau,k}(w)) d\tau. \quad (2.11)$$

Como  $X_t := X_{t,k}(w)$  é solução de (2.8) para todo  $0 \leq t \leq s_k(w)$  e usando a unicidade das soluções de (2.11) nós temos somente que mostrar o seguinte:

$$\lim_{k \rightarrow \infty} s_k(w) = \infty, \text{ para cada } w \in \Omega.$$

Como  $\{s_k(w)\}_{k \in \mathbb{N}}$  é para cada  $w \in \Omega$  uma sequência monótona crescente, vamos assumir a existência de  $s > 0$  e  $\widehat{w} \in \Omega$  com

$$\lim_{k \rightarrow \infty} s_k(\widehat{w}) = s.$$

Por definição de  $\{s_k(w)\}_{k \in \mathbb{N}}$  nós temos que

$$\|X_{s_k(\widehat{w})}(\widehat{w}) - x_0\|_2 = k \text{ para todo } k \in \mathbb{N}.$$

Considere a função quadrática  $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$

$$x \mapsto \frac{1}{2} x^T x$$

Estamos interessados em

$$\begin{aligned} \frac{d}{dt} \phi(X_t(\widehat{w})) &= \epsilon(B_t(\widehat{w}) - B_0(\widehat{w})) + \epsilon(B_s(\widehat{w}) - B_0(\widehat{w})) \\ &= -(X_t(\widehat{w}) + \epsilon(B_s(\widehat{w}) - B_t(\widehat{w})))^T q(X_t(\widehat{w})) \\ &= -X_t(\widehat{w})^T q(X_t(\widehat{w})) + \epsilon(B_t(\widehat{w}) - B_s(\widehat{w}))^T q(X_t(\widehat{w})) \end{aligned}$$

Existe um  $\bar{t} \in [0, s)$  tal que

1.  $\epsilon \|B_{\hat{t}}(\widehat{w}) - B_s(\widehat{w})\|_2 < \frac{1 + n\epsilon^2}{4}$  para todo  $\hat{t} \leq t \leq s$ .
2.  $\|X_{\bar{t}}(\widehat{w})\|_2 > r$ , o mesmo  $r$  definido na hipótese (1).

Para todo  $t \in [\bar{t}, s)$  com  $\|X_t(\bar{w})\|_2 > r$  nós obtemos com a hipótese (1):

$$\begin{aligned} \frac{d}{dt} \phi(X_t(\bar{w}) + \epsilon(B_t(\bar{w}) - B_s(\bar{w}))) &\leq -\frac{1+n\epsilon^2}{2} \max(1, \|q(X_t(\bar{w}))\|_2) + \\ &+ \epsilon(B_t(\bar{w}) - B_s(\bar{w}))^T q(X_t(\bar{w})) \\ &\leq -\frac{1+n\epsilon^2}{4} (2\max(1, \|q(X_t(\bar{w}))\|_2) - \|q(X_t(\bar{w}))\|_2) \\ &\leq -\frac{1+n\epsilon^2}{4} \end{aligned}$$

e portanto

$$\|X_t(\bar{w}) + \epsilon(B_s(\bar{w}) - B_t(\bar{w}))\|_2 \leq \|X_t(\bar{w})\|_2 + \max_{0 \leq t \leq s} (\epsilon \|B_s(\bar{w}) - B_t(\bar{w})\|_2)$$

para todo  $t \in [\bar{t}, s)$ . Isto é uma contradição para  $\lim_{k \rightarrow \infty} \|X_{s_k}(\bar{w}) - x_0\|_2 = \infty$ .  $\square$

O próximo resultado, cuja prova pode ser vista na referência [30], nos diz que para uma solução Pareto-ótima  $\bar{x}$ , quase todos os caminhos de  $X_t$  interceptam a bola com centro em  $\bar{x}$  e algum raio  $r > 0$  escolhido arbitrariamente após um tempo finito, para todo  $x_0 \in \mathbb{R}^n$ .

**Proposição 2.2 ([30]).** *Dada uma solução Pareto-ótima  $\bar{x} \in \mathbb{R}^n$ , considere a função  $S_{\bar{x}, r} : \Omega \rightarrow \mathbb{R}^n$  definida por*

$$w \mapsto \begin{cases} \inf\{t \geq 0; \|X_t - \bar{x}\| \leq r\} & , \text{ se } \{t \geq 0; \|X_t - \bar{x}\| \leq r\} \neq \emptyset \\ \infty & , \text{ caso contrário.} \end{cases}$$

Considerando a equação diferencial estocástica definida em 2.8 e assumindo a hipótese 1 temos que

$$\mathcal{W}(\{\omega \in \Omega; S_{\bar{x}, r}(w) < \infty\}) = 1.$$

### 2.3.1 Parâmetro $\epsilon > 0$

Este parâmetro, que aparece na hipótese 1, é uma medida de balanço entre a curva dominada de pontos

$$X_t = x_0 - \int_0^t q(X_s) ds, \quad (2.12)$$

e a busca aleatória usando a distribuição Gaussiana,

$$X_t(w) = x_0 + B_t(w) - B_0(w). \quad (2.13)$$

O valor ótimo de  $\epsilon > 0$  depende de cada função-objetivo. Na prática, quando for

---

apresentado o algoritmo SSW (ver algoritmo 3), se durante a execução numérica do caminho  $X_t$  leva-se muito tempo para ficar perto de qualquer solução Pareto-ótima então  $\epsilon > 0$  deve ter um valor pequeno. Do contrário, tomamos um valor grande.

## 3 *Algoritmo de otimização SSW*

Neste momento, faremos a apresentação do algoritmo proposto em [30]. Este algoritmo está na categoria do algoritmos estocásticos de espalhamento ponto-a-ponto, onde em cada iteração nós usamos apenas um ponto para geração do seguinte. No entanto, usa-se um arquivo para guardar as soluções não-dominadas obtidas anteriormente.

A seguir, faremos a descrição desta proposta e na sequencia abordaremos alguns problemas conhecidos na literatura ( cf. [6, 11, 13, 14, 32] ). Por fim, alguns comentários preliminares com tais problemas será feito.

### 3.1 Descrição

Foi abordado no capítulo 2, o problema de otimização multiobjetivo definido em (2.1) é equivalente a encontrar uma solução para o problema de valor inicial vetorial (2.7). Por sua vez, para encontrar as soluções para este último problema, analisamos a equação diferencial estocástica dada em (2.8), a saber,

$$dX_t = -q(X_t)dt + \epsilon dB_t, \quad X_0 = x_0 \quad (3.1)$$

onde  $\epsilon > 0$  e  $x_0 \in \mathbb{R}^n$ . Esta equação sob certas hipóteses possui solução. O termo  $-q(X_t)dt$  é a parte determinística de descida, enquanto que o movimento Browniano  $dB_t$  é a busca local aleatória.

A fim de obtemos soluções para a equação (2.8), faremos uso de algum método numérico para equações diferenciais. Por ser mais simples, e dispormos de diferenciabilidade das funções do problema (2.1), usaremos o método de Euler para EDEs descrito no apêndice C. Assim, dado um ponto inicial  $x_0$  e um passo inicial  $\sigma > 0$ , os elementos  $x_j$  são obtidos da seguinte forma. Primeiro, obtenha dois outros pontos  $x_{j+1}^1$  e  $x_{j+1}^2$  usando a direção de descida  $-q$ . O ponto  $x_{j+1}^1$  é obtido com o valor  $x_j$  e  $\sigma$ .

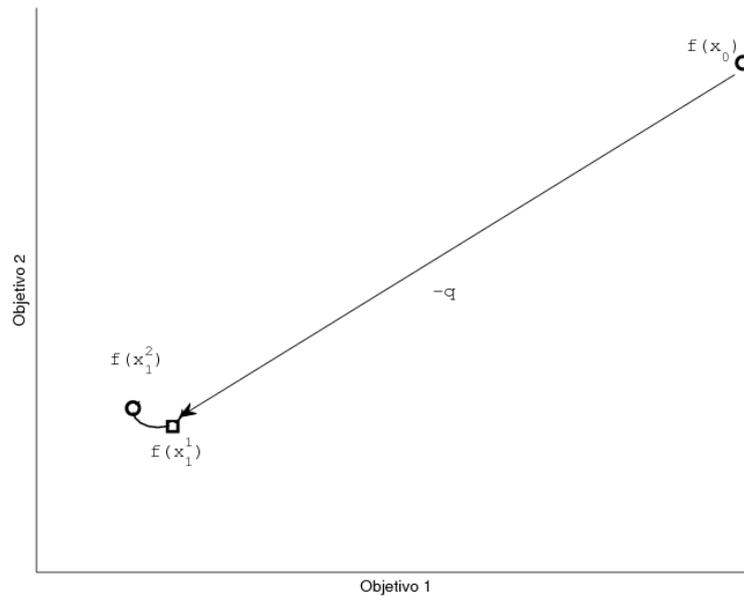


Figura 3.1: Esquema de iteração do algoritmo SSW para o caso bi-objetivo  $f(x) = (\text{Objetivo 1}, \text{Objetivo 2})$ .

Contudo, para o ponto  $x_{j+1}^2$  usaremos um elemento intermediário  $x(\sigma/2)$  que é obtido pelo método de Euler com  $x_{j+1}^1$  com  $\sigma/2$ . Desta maneira, poderemos ter um controle do tamanho de  $\sigma$ . Agora, verifique se  $|x_{j+1}^1 - x_{j+1}^2|_2 < \delta$ , onde  $\delta > 0$  é pre-fixado. Caso a desigualdade seja verdadeira, escolhemos um deles, digamos  $x_{j+1} = x_{j+1}^2$ . Se não, repetimos o processo para um tamanho de passo  $\sigma$  menor. A figura 3.1 mostra uma representação de como a partir do ponto  $Y$  obtemos o próximo iterado.

O algoritmo 3 contém o funcionamento completo da proposta SSW.

## 3.2 Experimentos

Nesta seção compararemos esta abordagem com o algoritmo NSGA2 (cf. [13]), tendo em vista que este tem resultados interessantes e já estar bem posto na literatura.

Esta seção não tem por intenção fazer uma comparação exaustiva sobre a qualidade de encontrar e espalhar as soluções dos algoritmos em questão. Isso será feito no capítulo 6. Exibiremos gráficos com as imagens do conjunto das soluções Pareto-ótimas obtidas com apenas uma execução de cada algoritmo a fim de, visivelmente, termos uma noção de como trabalha o algoritmo SSW. Convém lembrar que o foco do algoritmo SSW é funcionar com grandes quantidade de funções objetivo, como veremos no capítulo 6.

**Algoritmo 3: SSW**


---

**Entrada:**  $\epsilon > 0, \sigma > 0, \delta > 0, x_0 \in \mathbb{R}^n$   
**Saída:** P

```

1  P ← ∅;
2  j ← 0;
3  repita
4      seguir ← 0;
5       $\eta_1 \leftarrow \mathcal{N}(\mathbf{0}, I)$ ;
6       $\eta_2 \leftarrow \mathcal{N}(\mathbf{0}, I)$ ;
7       $\eta_3 \leftarrow \eta_1 + \eta_2$ ;
8      repita
9           $x_{j+1}^1 := x_j - \sigma q(x_j) - \epsilon \eta_3 \sqrt{\sigma/2}$ ;
10          $x(\sigma/2) := x_j - (\sigma/2)q(x_j) - \epsilon \eta_1 \sqrt{\sigma/2}$ ;
11          $x_{j+1}^2 := x(\sigma/2) - (\sigma/2)q(x(\sigma/2)) - \epsilon \eta_2 \sqrt{\sigma/2}$ ;
12         se  $\|x_{j+1}^1 - x_{j+1}^2\|_2 < \delta$  então
13              $x_j \leftarrow x_{j+1}^2$ ;
14             P ← P ∪ {xj};
15             seguir ← 1;
16         senão
17              $\sigma \leftarrow \frac{\sigma}{2}$ ;
18         fim
19     até seguir = 1;
20     j ← j + 1;
21 até critério de parada ser atingido;
```

---

A deficiência que veremos a seguir nos problemas apresentados já foi verificada em [32]. Reforçaremos tais resultados.

Fixaremos em 30.000 o número máximo de avaliações de função para todos os algoritmos abordados. Alguns trabalhos sugerem outros valores para o máximo de avaliação de função, por exemplo 20.000 a 100.000 em [32]. Neste nosso trabalho, queremos obter bons resultados com poucas avaliação de função.

As configurações dos algoritmos serão as seguintes:

- Para o NSGA2, fixaremos a população em 100 indivíduos. Este algoritmo usa um operador binário simulado (SBX) e mutação polinomial. A probabilidade de cruzamento é 0.9 e probabilidade de mutação é  $1/n$ , onde  $n$  é o número de variáveis de decisão. Além disso, distribuimos os índices dos operadores de cruzamento e mutação em  $\eta_c = 15$  e  $\eta_m = 20$ .
- Para o SSW, os parâmetros  $\sigma > 0$  e  $\epsilon > 0$  inicialmente serão 0.5 e 0.01, respectivamente.

### 3.2.1 Problemas com 2 objetivos

Os problemas FON, ZDT1 e ZDT2 serão da seguinte forma:

$$\min_{x \in \mathbb{R}^n} F(x)$$

onde  $F(x) = (f_1(x), f_2(x))$  e em cada problema especificaremos quais são as funções  $f_1$  e  $f_2$  bem como as restrições, quando houver, no espaço de variáveis.

1. Problema FON: É um problema cuja fronteira (ver figura 3.2) do conjunto Pareto-exato é não-convexa. As funções deste problema são:

$$f_1(x) = 1 - \exp\left(-\sum_{i=1}^3 \left(x_i - \frac{1}{\sqrt{3}}\right)\right),$$

$$f_2(x) = 1 - \exp\left(-\sum_{i=1}^3 \left(x_i + \frac{1}{\sqrt{3}}\right)\right),$$

$$x_i \in [-4, 4], \text{ para } i = 1, 2, 3.$$

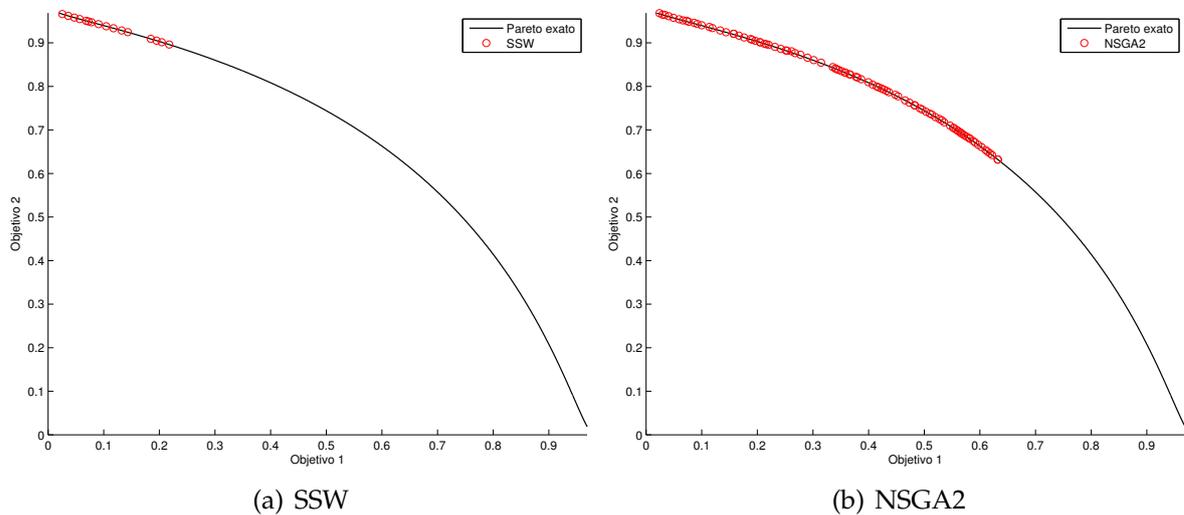


Figura 3.2: Resultados dos algoritmos para o problema FON.

Neste problema, vemos que ambos algoritmos conseguem alcançar o o conjunto Pareto-exato. É possível também perceber que os resultados vindos do algoritmos NSGA2 preenchem melhor o conjunto Pareto-exato quando comparados ao conjunto de soluções vindo do algoritmo SSW.

2. Problema ZDT1: É um problema cuja fronteira (ver figura 3.3) do conjunto Pareto-

exato é convexo à esquerda. As funções e restrições neste caso são:

$$\begin{aligned} f_1(x) &= x_1, \\ f_2(x) &= g(x) \left( 1 - \sqrt{\frac{x_1}{g(x)}} \right), \\ g(x) &= 1 + \frac{9}{30-1} \sum_{i=2}^{30} x_i^2 \\ x_i &\in [0,1], \text{ para } i = 1, 2, \dots, 30. \end{aligned}$$

Com 10.000 avaliações de função, percebemos que o NSGA2 ficou mais dis-

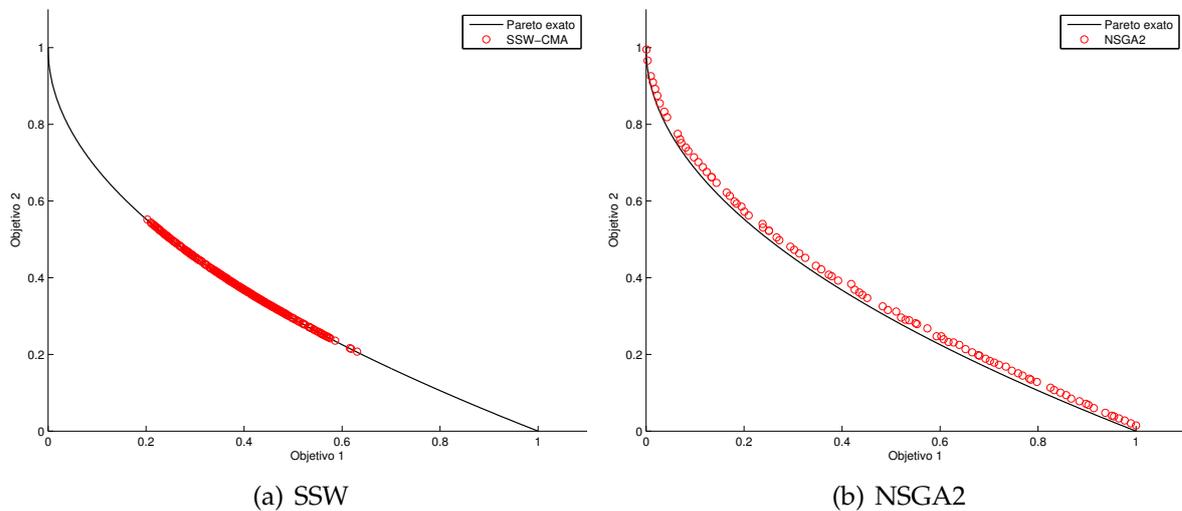


Figura 3.3: Resultados dos algoritmos para o problema ZDT1.

tante do conjunto Pareto-exato porém indicando que poderia convergir para toda extensão deste conjunto se fosse permitido um número maior de avaliação de função. Já os resultados do SSW, em relação a proximidade ao conjunto Pareto-exato é aparentemente melhor do que o NSGA2. No entanto, continua a não encontrar um bom conjunto representativo para o conjunto Pareto-exato.

3. Problema ZDT2: Interessante problema que apresenta uma fronteira (ver figura 3.4) Pareto-Ótima não convexa à esquerda. As funções neste caso são:

$$\begin{aligned} f_1(x) &= x_1, \\ f_2(x) &= g(x) \left( 1 - \left( \frac{x_1}{g(x)} \right)^2 \right), \\ g(x) &= 1 + \frac{9}{30-1} \sum_{i=2}^{30} x_i^2 \\ x_i &\in [0,1], \text{ para } i = 1, 2, \dots, 30. \end{aligned}$$

Este problema, assim como o experimento 2, o conjunto de soluções do NSGA2

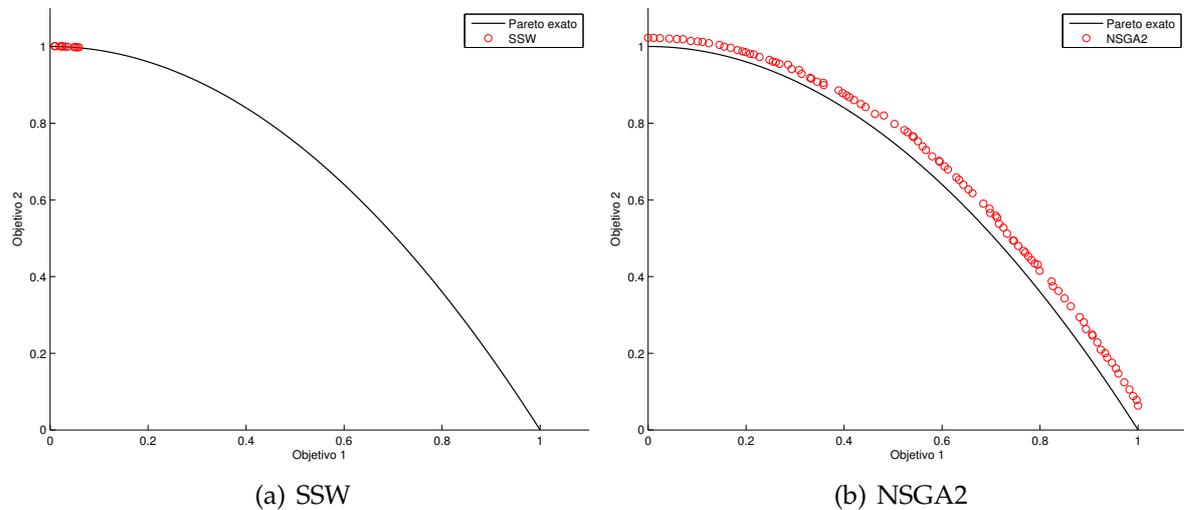


Figura 3.4: Resultados dos algoritmos para o problema ZDT2.

fica distante do conjunto Pareto-exato mas indicando que poderia convergir para o conjunto Pareto-exato com mais avaliações de função. Já os resultados do SSW fica mais perto do conjunto Pareto-exato com falhas na distribuição das soluções.

Com estes 3 problemas bi-objetivos, é notório que o algoritmo SSW encontra soluções Pareto-ótimas bem próximas do conjunto Pareto-exato. Contudo, percebemos várias parte do conjunto Pareto-exato que não foi encontrado pelo SSW. Diferente do que acontece com os resultados vindo do algoritmo NSGA2.

### 3.2.2 Problemas com 3 objetivos

Agora, os próximos problemas serão da seguinte forma:

$$\min_{x \in \mathbb{R}^n} F(x)$$

onde  $F(x) = (f_1(x), f_2(x), f_3(x))$  e em cada problema especificaremos quais são as funções  $f_1$ ,  $f_2$  e  $f_3$  bem como as restrições, quando houver, no espaço de variáveis.

1. Problema DTLZ1: É um problema cuja a fronteira (ver figura 3.5) do conjunto

Pareto-exato é linear.

$$f_1(x) = \frac{1}{2}x_1x_2(1 + g(x)),$$

$$f_2(x) = \frac{1}{2}x_1(1 - x_2)(1 + g(x)),$$

$$f_3(x) = \frac{1}{2}(1 - x_1)(1 + g(x)),$$

$$g(x) = \sum_{i=3}^{12} (x_i - 0.5)^2$$

$$x_i \in [0, 1], \text{ para } i = 1, 2, \dots, 12.$$

Podemos perceber (cf. figura 3.5) que o conjunto de soluções do NSGA2 en-

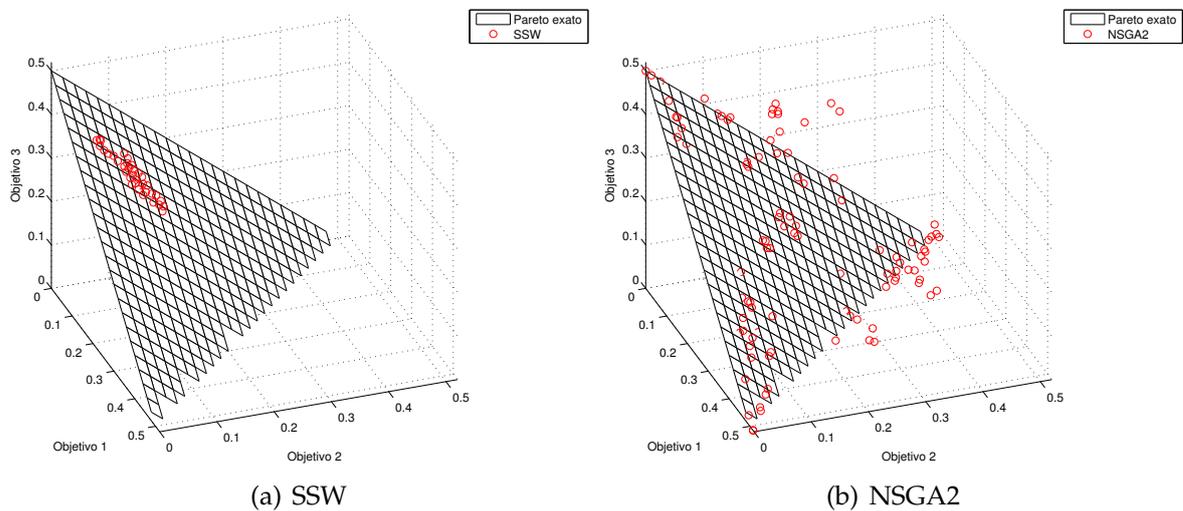


Figura 3.5: Resultados dos algoritmos para o problema DTLZ1.

contrar soluções do conjunto Pareto-exato. Talvez aumentando a quantidade de avaliação de função fosse possível ter um conjunto representativo de soluções. Já os resultados do SSW chegam perto do conjunto Pareto-exato porém se acumulam numa certa região.

2. Problema DTLZ2: As funções e restrições para este problema são:

$$f_1(x) = (1 + g(x)) \cos\left(x_1 \frac{\pi}{2}\right) \cos\left(x_2 \frac{\pi}{2}\right),$$

$$f_2(x) = (1 + g(x)) \cos\left(x_1 \frac{\pi}{2}\right) \sin\left(x_2 \frac{\pi}{2}\right),$$

$$f_3(x) = (1 + g(x)) \sin\left(x_1 \frac{\pi}{2}\right),$$

$$g(x) = \sum_{i=3}^{12} (x_i - 0.5)^2$$

$$x_i \in [0, 1], \text{ para } i = 1, 2, \dots, 12.$$

Este problema tem fronteira eficiente (ver figura 3.6) esférica satisfazendo a equação  $f_1^2 + f_2^2 + f_3^2 = 1$  para  $f_1, f_2 \in [0, 1]$ . Análogo ao que problema DTZL1, os

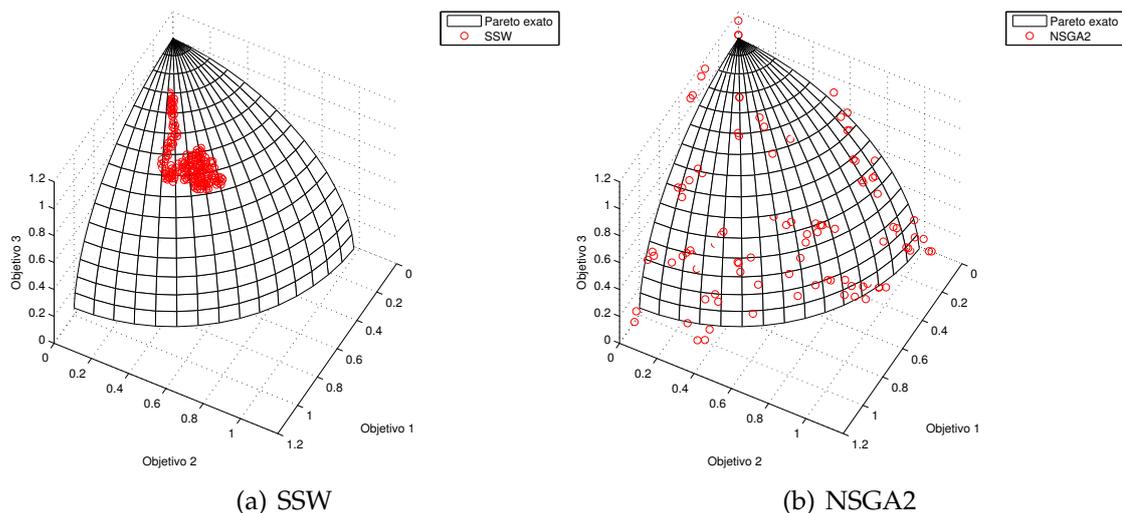


Figura 3.6: Resultados dos algoritmos para o problema DTLZ2.

resultados do SSW alcançam o conjunto Pareto-exato mas fica acumulada numa certa região. Os resultados do NSGA2 talvez fossem melhor se permitíssemos mais avaliação de função.

3. Problema DTLZ5: Faremos uma pequena variação do problema anterior, inserindo uma outra variável,  $\theta_i$ . Vejamos os dados do problema:

$$f_1(x) = (1 + g(x)) \cos\left(\theta_1 \frac{\pi}{2}\right) \cos\left(\theta_2 \frac{\pi}{2}\right),$$

$$f_2(x) = (1 + g(x)) \cos\left(\theta_1 \frac{\pi}{2}\right) \sin\left(\theta_2 \frac{\pi}{2}\right),$$

$$f_3(x) = (1 + g(x)) \sin\left(\theta_1 \frac{\pi}{2}\right),$$

onde

$$\begin{aligned}\theta_1 &= x_1, \\ \theta_2 &= \frac{\pi}{4(1+g(x))}(1+2g(x)x_2) \\ g(x) &= \sum_{i=3}^{12} (x_i - 0.5)^2, \\ x_i &\in [0,1], \text{ para } i = 1, 2, \dots, 12.\end{aligned}$$

Este é um problema com 12 variáveis tendo por solução ótima a curva  $f_3^2 =$

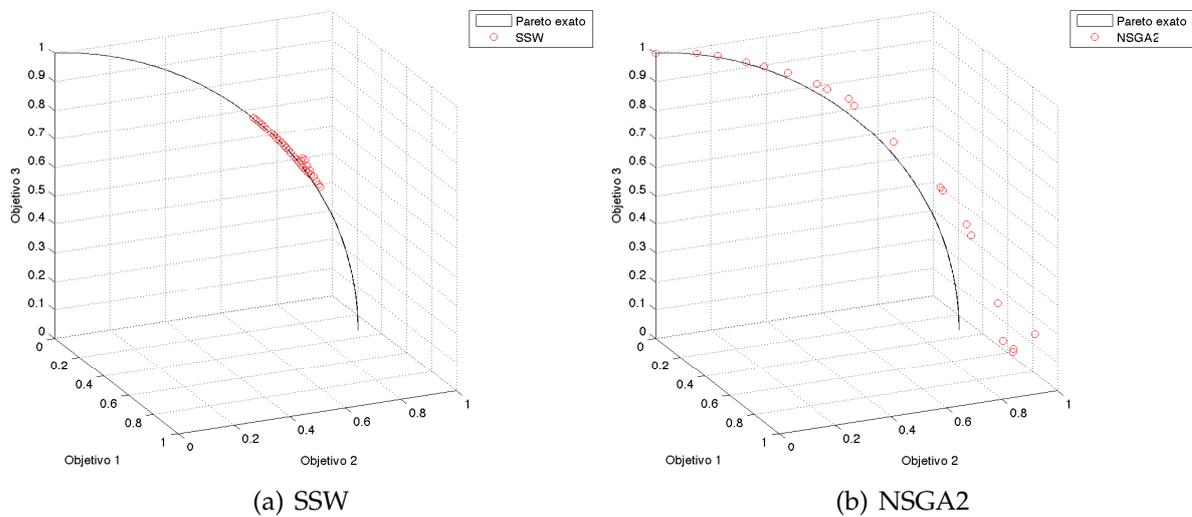


Figura 3.7: Resultados dos algoritmos para o problema DTLZ5.

$1 - f_1^2 - f_2^2$  com  $f_1, f_2 \in [0, \sqrt{0.5}]$  (ver figura 3.7). Assim como os demais problemas, o conjunto de soluções do SSW ficam mais perto do Pareto-exato que o NSGA2.

Os problemas com 3 funções objetivo apresentados acima, apontam o mesmo problema já visto com problemas bi-objetivo: o algoritmo SSW possui sérias dificuldades em espalhar as soluções.

### 3.3 Resultados

Preliminarmente podemos dizer que o algoritmo SSW tem a propriedade de encontrar soluções próximas do conjunto Pareto-ótimo porém não tem a capacidade de espalhar essas soluções por toda a extensão do conjunto solução exato. Com medidas de avaliação de desempenho\* adequadas para tratar distribuição de soluções, o

\*Falaremos de algumas medidas no capítulo 6.

algoritmo SSW foi testado em [32] apontando de fato esse problema.

Com vistas nesse problema, o capítulo 4 versará sobre uma maneira de corrigirmos tal dificuldade.

## 4 *Um operador de mutação para regiões factíveis*

Nesta seção discutiremos sobre um operador que ajude a melhorar a distribuição das soluções ao longo do conjunto de soluções-candidatas geradas pelo algoritmo SSW. Como foi verificado no capítulo 3, experimentalmente o algoritmo SSW deixa a desejar nesse ponto.

Para a construção do operador, usaremos ideias baseadas em estratégias evolutivas para otimização. Esse tipo de técnica foi criada por volta dos anos 60 e desenvolvida na década de 70 por Ingo Rechenberg, Hans-Paul Schwefel e seus colaboradores.

Um dos algoritmos mais conhecidos nesse sentido é a estratégia evolutiva de adaptar a matriz de covariância, também conhecida por CMA-ES\* (cf. [17, 15]). Este é um método estocástico para otimização para funções não-convexas e não-lineares. Algumas das vantagens nesse tipo de algoritmo residem em suas propriedades invariantes (cf. [16]), a saber:

1. Invariância diante de transformações preservando a ordem (monotonicidade estrita) dos valores das funções objetivos;
2. Invariância frente a transformações preservando ângulo do espaço de variáveis;
3. Invariância escalar se os escalares forem escolhidos adequadamente.

Tais propriedades são sempre desejáveis pois implicam em um comportamento uniforme sobre classes de funções.

A proposta original do CMA-ES feita em [16] foi expandida em [17], por adicionar a atualização de posto  $\mu$  à matriz de covariância<sup>†</sup> (cf. seção 4.2.5). Inicialmente, o CMA-ES foi aplicado para problemas mono-objetivos (cf. [15]) e avaliado o seu desempenho

---

\*Deriva da sua escrita na língua inglesa "*Covariance Matrix Adaptation-Evolution Strategy*".

<sup>†</sup>No apêndice B tem algumas informações sobre distribuição normal multivariada.

em [16]. Posteriormente, foi feita uma extensão para problemas multiobjetivos (cf. [18]), em que o autor denominou "MO-CMA-ES". Em linhas gerais, o CMA-ES precisa atualizar dois parâmetros: a) A matriz de covariância  $C$  e b) o tamanho e passo  $\sigma$ . Alguns estudos já foram feitos com vista melhorar a atualização desse dois elementos (cf. [19, 37]).

Nosso operador será baseado no modelo CMA-ES, ou seja, a heurística das atualizações da matriz de covariância e o tamanho de passo  $\sigma$  serão as mesmas que foram apresentando em [18, 16]. Além disso, será criado dois vetores ( $x_{medio}$  e  $x_{passo}$ ) que irão ajudar na orientação do algoritmo SSW na busca de novas soluções. Veremos que estes vetores serão cruciais para a atualização de vários parâmetros provenientes da estrutura do CMA-ES.

## 4.1 Preliminares

Para o que segue, denotaremos por  $x_{ssw}^{(g)}$  um individuo dado pelo algoritmo SSW na iteração  $g$ , tal que  $\|q(x)\|_2 < 10^{-2}$ , onde  $q: \mathbb{R}^n \rightarrow \mathbb{R}^n$  foi definida na seção 2.2.1. Esta condição evita que apliquemos a mutação em indivíduos que estão distantes do Pareto-ótimo, pois não temos interesses nesses indivíduos.

**Definição 4.1.** O conjunto com todos os descendentes de um individuo  $x_{ssw}^{(g)}$  será denotado por  $A(x_{ssw}^{(g)})$  e definido por

$$A(x_{ssw}^{(g)}) = \{x_j^{x_{ssw}}; 1 \leq j \leq \lambda\} \quad (4.1)$$

onde  $\lambda > 2$  é o tamanho da população dos descendentes. Denotaremos por  $\hat{A}(x_{ssw}^{(g)})$  o conjunto das "melhores"<sup>‡</sup> soluções do conjunto acima.

## 4.2 Atualizando a Matriz de Covariância $C$

Neste momento, vamos definir elementos para que seja possível atualizar a matriz  $C$ . A versão original (cf. [16]) no processo de atualização dessa matriz foi melhorado em [17] a fim de diminuir o tempo de complexidade da atualização.

O objetivo final da adaptação da matriz de covariâncias é, com um certo cuidado, aproximar as linhas de contorno das funções objetivo. Para funções quadráticas convexas este valor é aproximadamente a inversa da matriz hessiana (cf. apêndice B).

<sup>‡</sup>A noção de "melhor" será definida na seção 4.2.4.

Nesse processo, dois elementos se destacam : a) o caminho de evolução (representado por  $p_c$ ) e b) a matriz de posto  $\mu$  (denotada por  $C_\mu$ ). O vetor  $p_c$  às vezes é chamado de atualização de posto 1 da matriz  $C$  e será calculado a partir dos indivíduos de  $\hat{A}(x_{ssw}^{(g)})$ .

### 4.2.1 Gerando descendentes

A população de novos indivíduos candidatos a solução são gerados por amostragem de uma distribuição normal multivariada. A equação básica é

$$x_k^{x_{ssw}} \sim x_{ssw}^{(g)} + \sigma^{(g)} \mathcal{N}(0, C^{(g)}) \text{ para } k = 1, 2, \dots, \lambda, \quad (4.2)$$

onde

- $\sim$  denota a mesma distribuição sobre o lado esquerdo e direito.
- $\mathcal{N}(0, C^{(g)})$  é a distribuição normal multivariada com média 0 e matriz de covariância  $C^{(g)}$ .
- $x_k^{x_{ssw}} \in \mathbb{R}^n$ ,  $k$ -ésimo descendente associado a  $x_{ssw}$ .
- $\sigma^{(g)} \in \mathbb{R}_+$ , desvio padrão global, tamanho de passo, na iteração  $g$ .
- $C^{(g)} \in \mathbb{R}^n \times \mathbb{R}^n$ , matriz de covariância na iteração  $g$ .
- $\lambda \geq 2$ , tamanho da população descendentes.

A questão natural é saber como calcular  $C^{(g')}$  e  $\sigma^{(g')}$  para uma próxima solução  $x_{ssw}^{(g')}$ . As próximas seções cuidam de responder tais perguntas.

### 4.2.2 Classificação não-dominada

Na seção 2.2 abordamos o conceito de elementos não-dominados de um conjunto. Aqui, vamos definir de outra maneira o mesmo conceito, introduzindo a ideia de nível de não-dominância, pois será útil assim proceder.

Dada uma população  $A$  de soluções-candidatas, nós as classificaremos de acordo o seu nível de dominância. O conjunto das soluções não-dominadas de  $A$  será denotada por  $ndom(A)$ . Os elementos de  $ndom(A)$  recebem nível 1. Os demais níveis, definimos recursivamente por considerar o conjunto sem essas soluções com nível abaixo.

Formalizando,

$$\begin{aligned} \text{dom}_0(A) &= A \\ \text{dom}_l(A) &= \text{dom}_{l-1}(A) \setminus \text{ndom}_l(A) \\ \text{ndom}_l(A) &= \text{ndom}(\text{dom}_{l-1}(A)) \text{ para todo } l \in \{1, 2, \dots, |A|\}. \end{aligned}$$

Dado  $a \in A$ , nós definimos o nível de não-dominância  $r(a, A) = i \Leftrightarrow a \in \text{ndom}_i(A)$ . Essa forma de classificação foi introduzida em [13] e os mesmos autores calcularam a complexidade da classificação de  $n$  elementos, a saber,  $O(mn^2)$ , onde  $m$  é a quantidade de funções objetivo do problema. Para problemas em que o valor de  $m$  é muito grande, existem alguns estudos que visam diminuir os custos computacionais (cf. [7, 25]).

### 4.2.3 "Crowding – distance"

Na referência [13], as soluções não-dominadas de um conjunto  $A$  com um mesmo nível de não-dominância são classificadas de acordo com o quanto elas contribuem com a diversidade dos valores das funções objetivos em  $A$ . Isso é medido pelo que os autores denominaram de "Crowding-distance".

Para  $k > 1$  funções objetivos, a crowding-distance de  $a \in A$ , que denotaremos por  $CD(a, A)$  é dada por

$$CD(a, A) = \sum_{i=1}^k \frac{cd_i(a, A)}{f_i^{\max} - f_i^{\min}} \quad (4.3)$$

onde  $f_i^{\max}$  e  $f_i^{\min}$  são estimativas dos valores de máximo e mínimo, respectivamente, da  $i$ -ésima função objetivo e

$$cd_i(a, A) := \begin{cases} \infty, & \text{se } f_i(a) = \min \{f_i(a'); a' \in A\} \text{ ou } f_i(a) = \max \{f_i(a'); a' \in A\}. \\ \min \{f_i(a'') - f_i(a'); a', a'' \in A \setminus \{a\} : f_i(a') \leq f_i(a) \leq f_i(a'')\}, & \text{caso contrário.} \end{cases}$$

### 4.2.4 Relação $\prec_{C,A}$

Agora, juntando o conceito de nível de não-dominância e a crowding-distance, podemos estabelecer uma relação de ordem total (diferente da feita em 2.2 que era

parcial) da seguinte maneira:

$$a \prec_{c,A} a' \Leftrightarrow r(a,A) < r(a',A) \text{ ou} \\ [ r(a,A) = r(a',A) \text{ e } CD(a, \text{nodom}_{r(a,A)}(A)) > CD(a', \text{nodom}_{r(a',A)}(A))] ]$$

para  $a, a' \in A$ .

Vamos escolher o individuo com maior  $CD(\cdot, \cdot)$ , pois assim induzimos uma diversidade no espaço de objetivos.

### 4.2.5 Caminho de evolução $p_c$

Nós chamaremos uma sequência de sucessivos passos, a estratégia tomada sobre um número de gerações, um caminho de evolução. Tal caminho pode ser expresso por uma soma de consecutivos passos. Na prática, construímos o caminho de evolução,  $p_c \in \mathbb{R}^n$ , iniciando em  $p_c^{(0)} = \mathbf{0}$  com uma taxa de aprendizado  $0 < c_c \leq 1$ , por

$$p_c^{(g+1)} = (1 - c_c)p_c^{(g)} + \left( \sqrt{c_c(2 - c_c)\mu_{eff}} \right) x_{passo} \quad (4.4)$$

onde,

- $p_c^{(g)} \in \mathbb{R}^n$  é o caminho de evolução na geração  $g$ .
- $c_c \leq 1$  é a taxa de aprendizagem para atualizar o caminho de evolução.
- $x_{passo}^{(g)}$  é um vetor de  $\mathbb{R}^n$ .
- $\mu_{eff} := \left( \sum_{i=1}^{\mu} w_i^2 \right)^{-1}$ , os valores  $w_i$  serão definidos em (4.10).

O vetor  $x_{passo}^{(g)}$  mantém um registro das atualizações anteriores, para que assim tenhamos informações para atualizar o  $p_c$  de. Na seção 4.4 apresentaremos como escolhemos esse vetor.

Sobre a taxa de aprendizado, notemos que se  $c_c = 0$  então não estamos atualizando o caminho de evolução. Além disso, se  $c_c = 1$  e  $\mu_{eff} = 1$  então  $p_c^{(g+1)} = x_{passo}^{(g)}$ . Já o termo  $\sqrt{c_c(2 - c_c)\mu_{eff}}$  é uma renormalização constante para  $p_c$  (cf. [16]).

### Atualizando a matriz de covariância $C$

Considere primeiro a matriz  $C_\mu^{(g+1)}$  dada por

$$C_\mu^{(g+1)} = \sum_{i=1}^{\mu} w_i (x_i - x_{medio})(x_i - x_{medio})^T, \quad (4.5)$$

onde  $x_i \in \hat{A}(x_{ssw}^{(g)})$  para todo  $i \in \{1, 2, \dots, \mu\}$  e  $x_{medio} \in \mathbb{R}^n$  será definido mais adiante. Esta matriz é um estimador para  $C^{(g)}$  observando a variação da distribuição dos passos amostrados,  $(x_i - x_{medio})$ . Vamos nos referir a essa matriz por de matriz de posto  $\mu$  (cf. [17]).

Com a matriz dada em (4.5) podemos estender a regra de atualização da matriz  $C$  para populações de descendentes grande, escolhendo  $\mu$  vetores para atualizar a matriz  $C$  na geração seguinte.

A regra para a adaptação da matriz  $C^{(g+1)}$  será a combinação do caminho de evolução  $p_c$  com  $C_\mu$ , além da própria matriz da geração anterior. Em outras palavras, escolhendo  $C^{(0)} = \mathbf{I}$  e uma constante de aprendizado  $0 < c_\mu \leq 1$ ,

$$C^{(g+1)} = (1 - c_1 - c_\mu) \underbrace{C^{(g)}}_{\text{Geração anterior}} + c_1 \underbrace{p_c^{(g+1)} (p_c^{(g+1)})^T}_{\text{posto 1}} + \frac{c_\mu}{\sigma^{(g)}} \underbrace{C_\mu^{(g+1)}}_{\text{posto } \mu} \quad (4.6)$$

onde  $c_1 \approx 2/n^2$  e  $c_\mu \approx \min\{\mu_{eff}/n^2, 1 - c_1\}$ .

### 4.3 Atualizando o tamanho de passo $\sigma$

Com o objetivo de adaptar a matriz de covariância na seção anterior, inserimos um parâmetro  $\sigma^{(g)}$  na equação (4.6).

Uma razão para termos tal valor alterado em cada geração seria o fato de que um comprimento de passo global pode não ser bem aproximado por (4.6), em particular se  $\mu_{eff}$  é escolhido muito grande.

Para controlar  $\sigma^{(g)}$ , utilizaremos um caminho de evolução, isto é, a soma dos sucessivos passos.

### 4.3.1 Caminho de Evolução $p_\sigma$

Neste momento, construiremos o caminho de evolução para o tamanho de passo  $\sigma$ , seguindo a heurística tradicional:

- Quando o caminho de evolução é curto, passos sucessivos podem cancelar um ao outro. Se isto ocorre, devemos diminuir o valor de  $\sigma$ .
- Quando o caminho de evolução é longo, os passos são apontados para direções similares. Ocorrendo isso, o valor de  $\sigma$  deve aumentar.

Construímos o caminho de evolução,  $p_\sigma \in \mathbb{R}^n$ , iniciando em  $p_\sigma^{(0)} = \mathbf{0}$  com uma taxa de aprendizado  $0 < c_\sigma \leq 1$ , por

$$p_\sigma^{(g+1)} = (1 - c_\sigma)p_\sigma^{(g)} + \left( \sqrt{c_\sigma(2 - c_\sigma)\mu_{eff}} \right) (C^{(g)})^{-\frac{1}{2}} \cdot x_{passo} \quad (4.7)$$

$$(C^{(g)})^{-\frac{1}{2}} := B^{(g)} (D^{(g)})^{-1} (B^{(g)})^T \quad (4.8)$$

onde

$$(C^{(g)}) = B^{(g)} (D^{(g)})^2 (B^{(g)})^T$$

é a autodecomposição de  $(C^{(g)})$  em que  $B^{(g)}$  é uma base ortonormal de autovetores, e os elementos da diagonal da matriz  $D^{(g)}$  são raízes quadradas dos correspondentes autovalores positivos.

Enfim, vamos atualizar  $\sigma$  pela equação abaixo

$$\sigma^{(g+1)} = \sigma^{(g)} \exp \left( \frac{c_\sigma}{d_\sigma} \left( \frac{\|p_\sigma^{(g+1)}\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1 \right) \right) \quad (4.9)$$

onde

- $d_\sigma \approx 1$  é chamado de parâmetro de amortecimento;

- $\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\| = \frac{\sqrt{2}\Gamma\left(\frac{n+1}{2}\right)}{\Gamma(n/2)} \approx \sqrt{n} \left( 1 - \frac{1}{4n} + \frac{1}{21n^2} \right)$  é o comprimento esperado de  $p_\sigma$  com respeito a uma seleção aleatória (cf. [16]).

## 4.4 Proposta de operador: ES-mut

Nesta seção, vamos definir um operador para o espaço de variáveis com o propósito de espalhar as soluções vindo do algoritmo SSW. Uma implementação com esse

objetivo é necessário, pois no algoritmo SSW quando estamos próximos do conjunto Pareto-exato, a componente de busca estocástica  $dB_t$  tem mais contribuição do que a componente determinística  $-q(X_t)$  (cf. equação 3.1).

Nas equações (4.4, 4.6, 4.7) ficaram pendentes como escolhemos os vetores  $x_{medio}$  e  $x_{passo}$ . Conforme já foi visto, tais elementos são fundamentais na atualização da matriz  $C$  e  $\sigma$ . Para problemas mono-objetivo, esses elementos são simples de serem calculados (cf. [15]). Para problemas multiobjetivo, precisamos pensar diferente.

A maneira mais natural para definirmos o elemento  $x_{medio}$  é como combinação das melhores soluções geradas com respeito a relação  $\prec_{c,A}$  definida na seção (4.2.4). Esta relação utiliza um segundo critério para selecionar duas soluções com um mesmo posto de dominância, escolhendo a solução que melhor contribui para o espalhamento das soluções.

A figura 4.1(a) exibi a heurística de geração dos descendentes baseado na equação (4.2.1). Já a figura 4.1(b) exemplifica como escolhemos os melhores descendentes dentre os que foram gerados na figura 4.1(a), bem como tomamos o vetor  $x_{medio}$ .

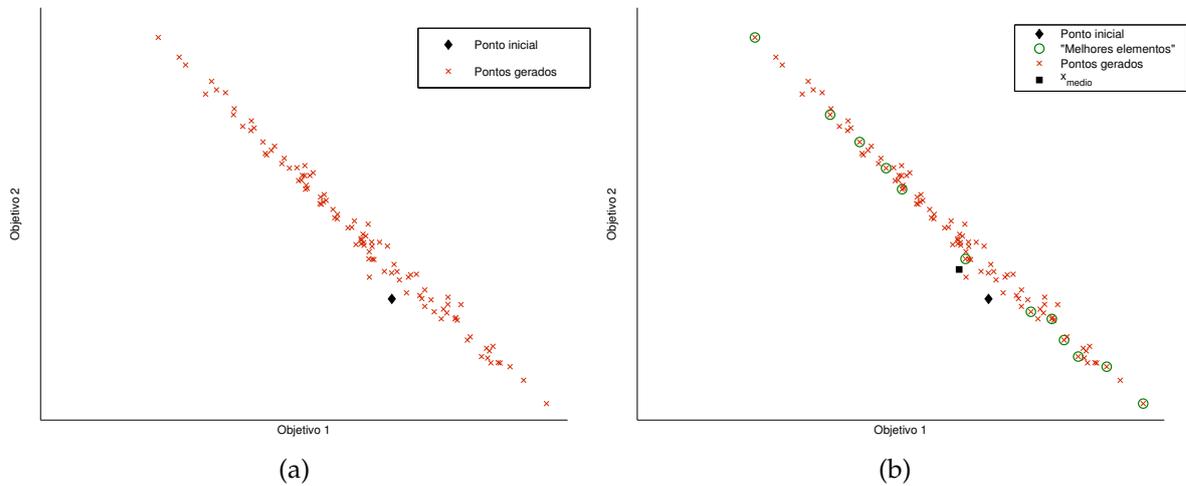


Figura 4.1: Ilustração da geração dos descendentes, escolha dos melhores e determinação do vetor  $x_{medio}$  em uma iteração para um problema bi-objetivo.

Formalmente seria o seguinte: dada uma solução  $x_{ssw}^{(g)}$  vindo do algoritmo SSW, definiremos  $x_{medio}$  como sendo a média ponderada dos elementos de  $\hat{A}(x_{ssw}^{(g)})$ , ou seja, se  $|\hat{A}(x_{ssw}^{(g)})| = \mu$  então

$$x_{medio} = \sum_{i=1}^{\mu} w_i x_i, \quad (4.10)$$

com

$$\sum_{i=1}^{\mu} w_i = 1, w_1 \geq w_2 \geq \dots \geq w_{\mu} > 0$$

onde  $x_i \in \hat{A}(x_{ssw}^{(g)})$  para todo  $i \in \{1, 2, \dots, \mu\}$ .

Em seguida, definiremos o vetor  $x_{passo}$  por

$$x_{passo} = \frac{x_{medio} - x_{ssw}^{(g)}}{\sigma^{(g)}} \quad (4.11)$$

Com esses elementos bem definidos, podemos descrever como funcionar o operador ES-mut:

1. Geramos os indivíduos pela equação 4.2;
2. Determinamos o conjunto  $\hat{A}(\cdot)$  pela definição 4.1);
3. Calcula-se os vetores  $x_{medio}$  e  $x_{passo}$  pelas equações (4.10) e (4.11);
4. Atualiza-se os caminhos de evoluções e a matriz de covariâncias pelas equações (4.4), (4.7) e (4.6), respectivamente.

Depois de feito os passos acima, permitimos que o algoritmo SSW utilize na sua próxima iteração o vetor  $x_{medio}$ . Usar esse vetor no SSW permitirá encontrar outras regiões no espaço de busca para o problema (2.1).

Por fim, denotaremos por SSW-CMA ao algoritmo SSW com o operador ES-mut definido acima.

No capítulo 6, submeteremos esse algoritmo a algumas medidas para avaliar seu desempenho em diversos problemas. Na próxima seção faremos alguns experimentos com uma única execução e um valor máximo de avaliação. Veremos que nossa proposta se saiu bem quando comparado com o algoritmo NSGA2. As razões pelas quais isso acontece vem dos fatos listados abaixo:

1. Uso de direções de descidas na busca de soluções;
2. Configuração de parâmetros auto-adaptativo;
3. Utilizar a "crowding-distance" como segundo critério para escolher as melhores soluções;

#### 4. Criação e utilização dos vetores $x_{medio}$ e $x_{passo}$ .

O pseudo-código deste novo algoritmo esta descrito em 4.

---

#### Algoritmo 4: SSW-CMA

---

**Entrada:**  $\epsilon > 0, \sigma > 0, \sigma_2 > 0, \delta > 0, x_0 \in \mathbb{R}^n$   
**Saída:**  $P$

- 1  $P \leftarrow \emptyset;$
- 2  $j \leftarrow 0;$
- 3 **repita**
- 4      $seguir \leftarrow 0;$
- 5      $\eta_1 \leftarrow \mathcal{N}(\mathbf{0}, I);$
- 6      $\eta_2 \leftarrow \mathcal{N}(\mathbf{0}, I);$
- 7      $\eta_3 \leftarrow \eta_1 + \eta_2;$
- 8     **repita**
- 9          $x_{j+1}^1 := x_j - \sigma_1 q(x_j) - \epsilon \eta_3 \sqrt{\sigma_1/2};$
- 10          $x(\sigma/2) := x_j - (\sigma_1/2)q(x_j) - \epsilon \eta_1 \sqrt{\sigma_1/2};$
- 11          $x_{j+1}^2 := x(\sigma_1/2) - (\sigma_1/2)q(x(\sigma_1/2)) - \epsilon \eta_2 \sqrt{\sigma_1/2};$
- 12         **se**  $\|x_{j+1}^1 - x_{j+1}^2\|_2 < \delta$  **então**
- 13              $x_j \leftarrow x_{j+1}^2;$
- 14              $P \leftarrow P \cup \{x_j\};$
- 15              $seguir \leftarrow 1;$
- 16              $(x_j, \sigma_2, p_{\sigma_2}, p_c, C, P) \leftarrow ES - mut(x_j, \sigma_2, p_{\sigma_2}, p_c, C, P);$
- 17         **senão**
- 18              $\sigma \leftarrow \frac{\sigma}{2};$
- 19         **fim**
- 20     **até**  $seguir = 1;$
- 21      $j \leftarrow j + 1;$
- 22 **até** critério de parada ser atingido;

---

## 4.5 Experimentos

Para verificar a teoria apresentada acima, usamos como referência a mesmas funções teste da seção 3. Em todos os experimentos, usamos  $\lambda = 100$  e em 30.000 o número máximo de avaliação de função. Usamos como referência os mesmos parâmetros do SSW usados em 3.2 no SSW-CMA.

A fim de termos uma referência de como estar o algoritmo SSW-CMA, faremos alguns experimentos com duas e três funções objetivos. Contudo, como temos também um foco em problemas com muitos objetivos, no capítulo 6 faremos experimentos com tal classe funções.

As figuras (4.2-4.4) mostram uma comparação, com apenas uma execução dos algoritmos, em problemas com duas funções objetivos (problemas FON, ZDT1 e ZDT2).

---

**Função ES-mut**( $x, \sigma, p_\sigma, p_c, C, P$ )
 

---

**Saída:** [ $x_{\text{medio}}, \sigma, p_\sigma, p_c, C, P$ ]

```

1  $A \leftarrow x$ ;
2  $\text{succ} \leftarrow \emptyset$ ; Vamos guardar os índices dos indivíduos de sucesso aqui.
3 para  $i \leftarrow 1$  até  $\lambda$  faça
4   |  $x_i \sim \mathcal{N}(x, \sigma^2, C)$ ; gerando os descendentes conforme equação (4.2);
5   |  $A \leftarrow A \cup \{x_i\}$ ;
6 fim
7 para  $i \leftarrow 1$  até  $\lambda$  faça
8   | se  $x_i <_{c,A} x$  então
9   |   |  $P \leftarrow P \cup \{x_i\}$ ; guarda individuo melhor que seu parente x;
10  |   fim
11  |    $\text{succ} = \text{succ} \cup \{i\}$ ; guarda índice de sucesso;
12 fim
13 calcula  $x_{\text{medio}}$  e  $x_{\text{passo}}$  conforme (4.10, 4.11);
14 atualiza  $p_\sigma$  conforme 4.7;
15 atualiza  $p_c$  conforme 4.4;
16 atualiza  $C$  conforme 4.6;
17 atualiza  $\sigma$  conforme 4.9;

```

---

Os conjuntos de soluções obtidos pelo SSW-CMA indicam um boa distribuição das soluções, encontram toda a extensão. Já as figuras (4.5-4.7) mostram comparações com funções com três funções objetivo. Os resultados vindo do SSW-CMA indicam ser bem melhor do que aqueles vindos do NSGA2.

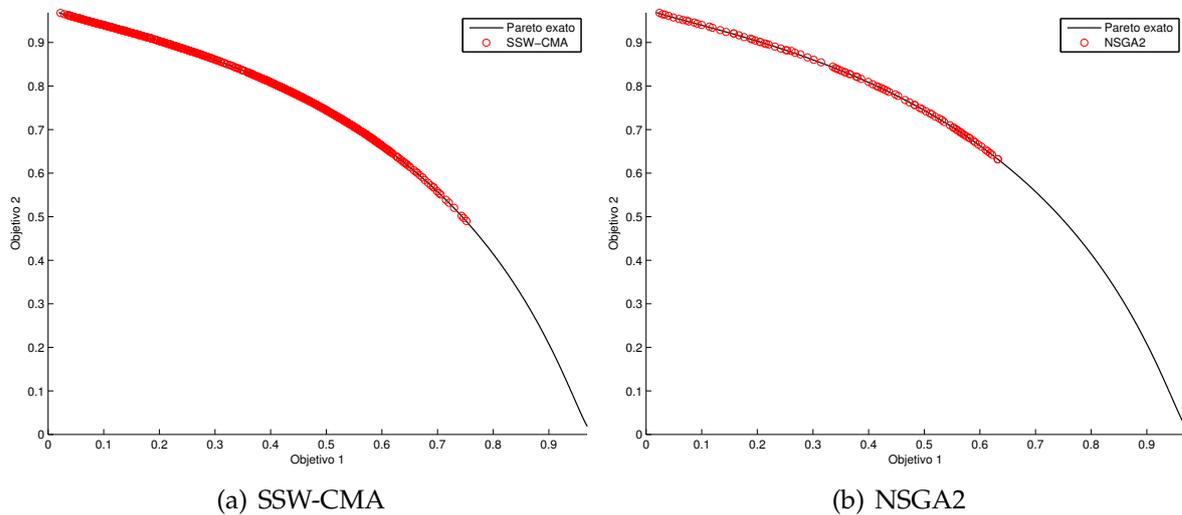


Figura 4.2: Problema FON

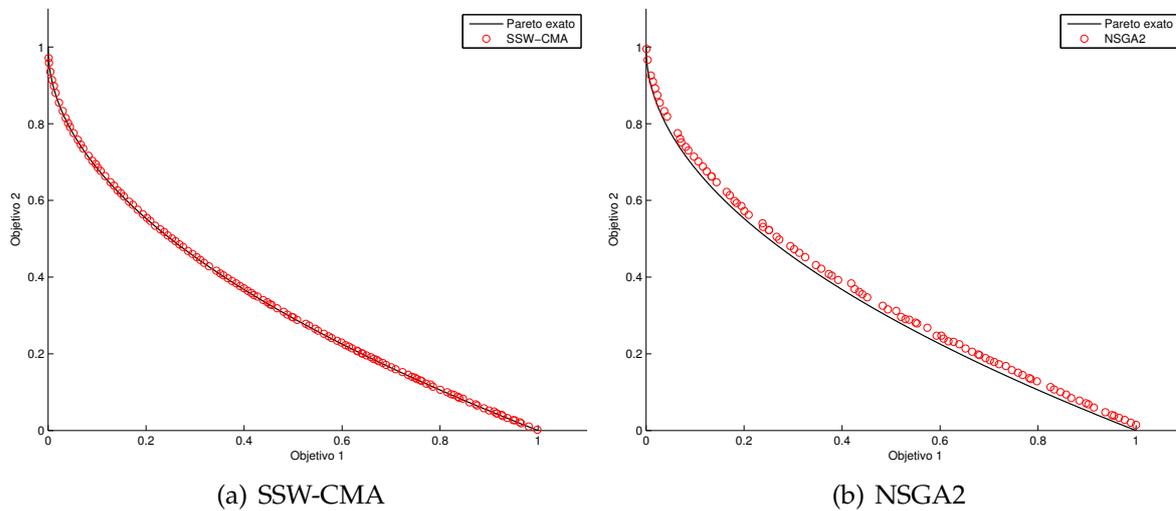


Figura 4.3: Problema ZDT1

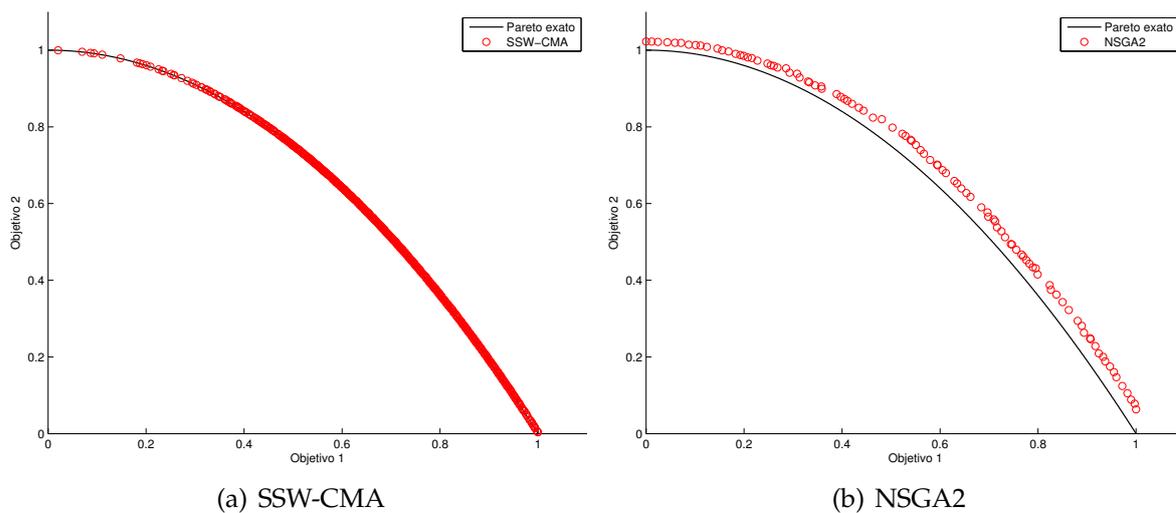


Figura 4.4: Problema ZDT2

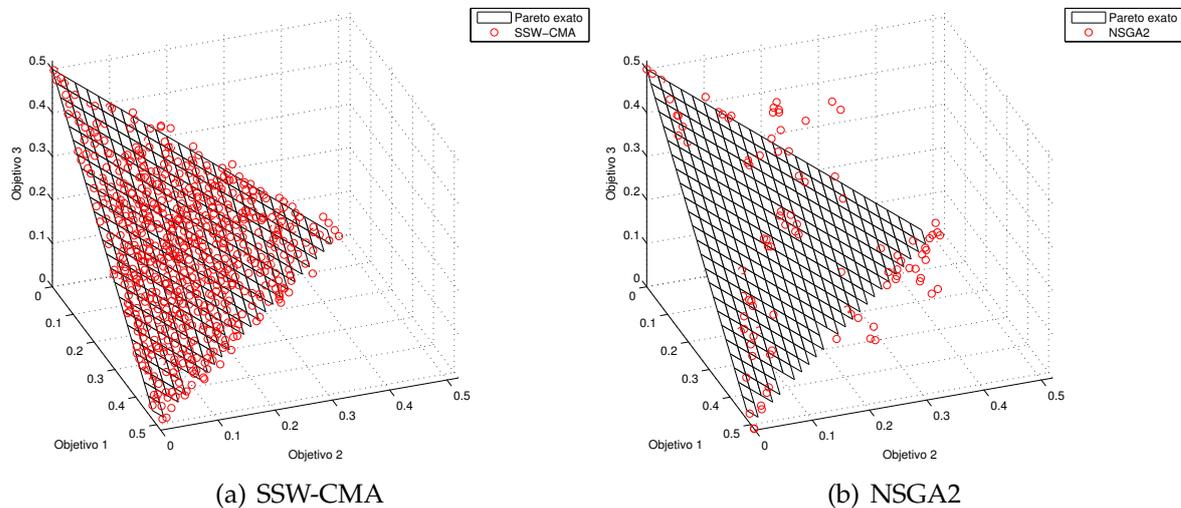


Figura 4.5: Problema DTLZ1

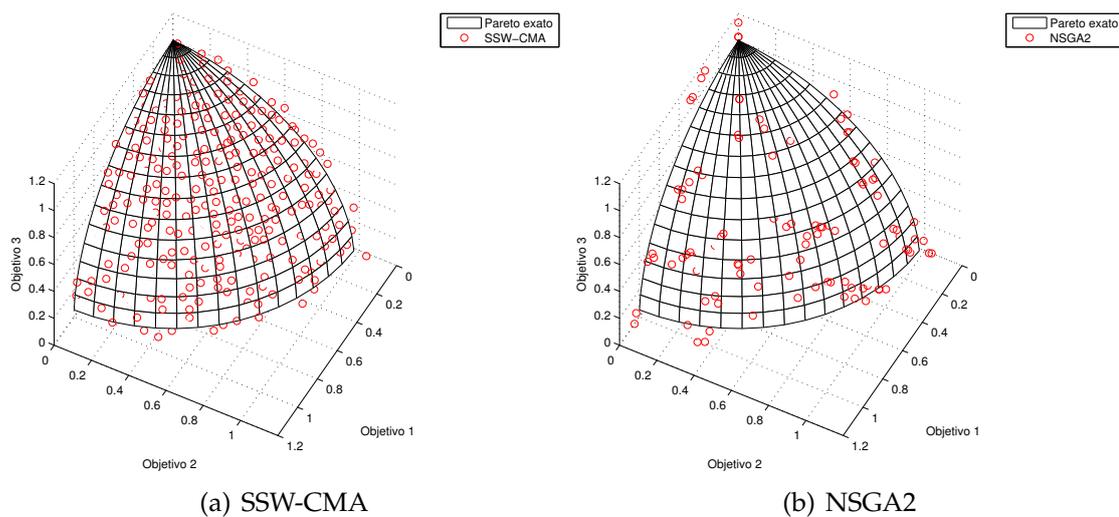


Figura 4.6: Problema DTLZ2

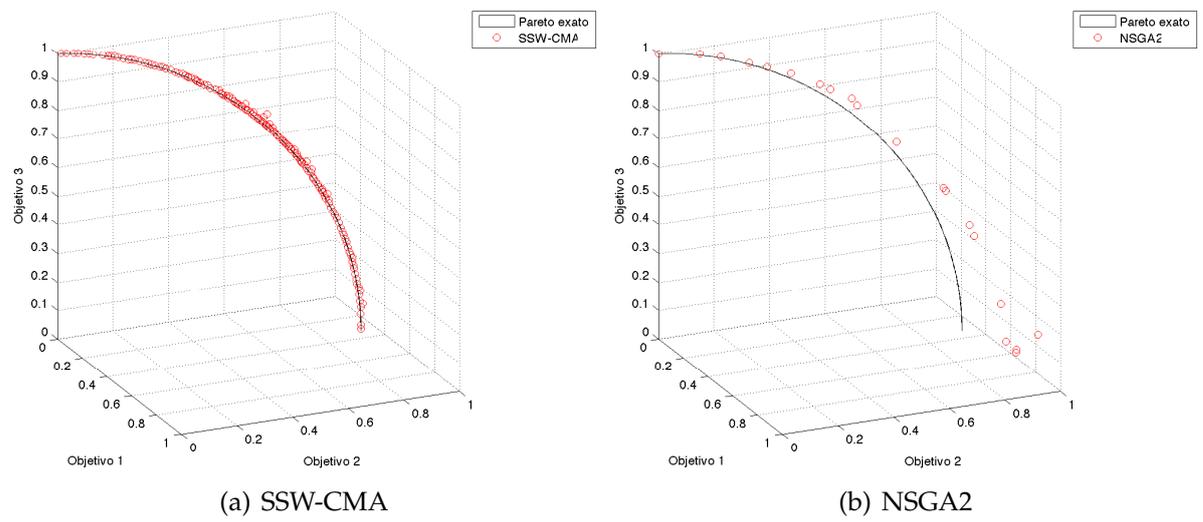


Figura 4.7: Problema DTLZ5

## 5 *Um procedimento de busca local*

Neste capítulo, nos preocuparemos em encontrar uma melhor aproximação de um dado conjunto representativo de soluções do problema 2.1 dado por um certo algoritmo, quer seja SSW-CMA ou qualquer outro.

Uma metaheurística já presente na literatura é o procedimento de busca local (cf. [5, 9, 39]), o qual define, para cada solução, uma vizinhança composta por um conjunto de soluções "mais próximas" do conjunto Pareto-exato.

A proposta que apresentaremos encontra, para cada solução, uma outra solução que domina a inicial, porém mais próxima do conjunto Pareto-exato. Tal ideia utiliza um procedimento com seção áurea para problemas de otimização multiobjetivo (cf. [36]). Essa abordagem híbrida para o algoritmo SSW com tal busca local foi proposta [34]. O foco desse estudo foi deficiência que tem o algoritmo SSW de gerar soluções quando se estar próximo do conjunto Pareto-exato pois a componente estocástica de busca prevalece.

A limitação dessa proposta é a necessidade de que as funções do problema 2.1 sejam pelo menos de classe  $C^2$ . No entanto, utilizamos essa abordagem pela vantagem de poder fixar, a priori, o quão próximo a nova solução estará do conjunto Pareto-exato.

Este procedimento será acoplado aos algoritmos SSW, SSW-CMA e NSGA2 a fim de todas as soluções encontradas estejam próximas do conjunto Pareto-exato.

### 5.1 Busca local

Como já sabemos, todo método de descida necessita de uma direção que melhore todas as funções simultaneamente. Para tanto, usaremos a função

$$q : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

apresentada em 2.2.1, pois quando  $x$  não é um ponto Pareto-ótimo,  $-q(x)$  é uma direção de descida para todas as funções do problema 2.1.

Considerando  $\{x_1, x_2, \dots, x_p\} \subset \mathbb{R}^n$  um conjunto de soluções dado pelo algoritmo A, vamos usar o seguinte procedimento de busca local dado no algoritmo 5:

---

**Algoritmo 5:** Procedimento de busca local

---

**Entrada:**  $q : \mathbb{R}^n \rightarrow \mathbb{R}^n, \{x_1, x_2, \dots, x_p\}, \text{tol}q > 0, \epsilon > 0$

**Saída:**  $\{x_1^*, x_2^*, \dots, x_p^*\}$

1 **para**  $i$  de 1 até  $p$  **faça**

2      $x \leftarrow x_i;$

3     **repita**

4          $d \leftarrow -q(x);$

5          $x \leftarrow \text{seção-áurea-multiobjetivo}(x, d, f, \epsilon);$

6     **até**  $\|q(x)\|^2 \geq \text{tol}q;$

7      $x_i^* \leftarrow x;$

8 **fim**

---

A operação que aparece na linha 5 do algoritmo acima, será descrita na próxima seção.

## 5.2 Seção áurea para problemas multiobjetivo

Dados os pontos  $x, d \in \mathbb{R}^n$ , considere o segmento de reta parametrizado em  $\alpha$  por

$$\{y \in \mathbb{R}^n; y = x + \alpha d, 0 \leq \alpha \leq 1\} \quad (5.1)$$

Para um valor  $x \in \mathbb{R}^n$  fixado, a função  $f(x)$  dada pelo problema 2.1, pode ser reescrita por  $F : \mathbb{R} \rightarrow \mathbb{R}^m$  definida por

$$F(\alpha) := f(x + \alpha d) \quad (5.2)$$

Nossa busca será conduzida em  $\mathcal{R} = [0, 1]$ . Desde modo, o problema de otimização multiobjetivo restrito a  $\mathcal{R}$  torna-se o problema de encontrar  $\mathcal{R}^*$ , onde

$$\mathcal{R}^* = \{\alpha^* \in \mathcal{R}; \nexists \alpha \in \mathcal{R} \text{ tal que } F(\alpha) \leq F(\alpha^*)\} \quad (5.3)$$

Também vamos definir os conjuntos que contém candidatos a solução que dominam

$\alpha = 0$  nos conjuntos  $\mathcal{R}$  e  $\mathcal{R}^*$ , respectivamente:

$$\mathcal{R}_d = \{\alpha \in \mathcal{R}; F(\alpha) \leq F(0)\}$$

$$\mathcal{R}_d^* = \{\alpha \in \mathcal{R}^*; F(\alpha) \leq F(0)\}$$

O mínimo individual de cada função objetivo restrito à  $\mathcal{R}$  é

$$\rho_i = \arg \min F_i(\alpha)$$

$$\text{sujeito a: } \alpha \in \mathcal{R},$$

Os próximos lemas nos dão uma caracterização dos conjuntos  $\mathcal{R}$  e  $\mathcal{R}^*$ .

**Lema 5.1** ([34]). *Considerando o problema de otimização dado em 5.3 e assumindo que cada função  $F_i$  é unimodal em  $\mathcal{R}$  então conjunto  $\mathcal{R}^* = [\rho_a, \rho_b]$  onde*

$$\rho_a = \min\{\rho_1, \rho_2, \dots, \rho_p\}$$

e

$$\rho_b = \max\{\rho_1, \rho_2, \dots, \rho_p\}.$$

**Observação 8.** *Como funções quase-convexas\* satisfazem as condições unimodalidade 2.2.2 sobre qualquer segmento de reta, o lema é válido para essa classe de funções.*

O próximo lema é a generalização do lema 2.1 para o caso multiobjetivo.

**Lema 5.2** ([34]). *Sejam  $\alpha_1, \alpha_2 \in \mathcal{R}$  tal que  $0 < \alpha_1 < \alpha_2 < 1$ . Então*

1. *Se  $F(\alpha_1) \leq F(\alpha_2)$  então  $\mathcal{R}^* \cap [\alpha_2, 1] = \emptyset$ .*
2. *Se  $F(\alpha_2) \leq F(\alpha_1)$  então  $\mathcal{R}^* \cap [0, \alpha_1] = \emptyset$ .*
3. *Se  $F(\alpha_1) \not\leq F(\alpha_2)$  e  $F(\alpha_2) \not\leq F(\alpha_1)$  então  $\mathcal{R}^* \cap [\alpha_1, \alpha_2] = \emptyset$ .*

**Lema 5.3** ([34]). *Seja  $\alpha_d = \sup\{\alpha \in (0, 1]; F(\alpha) \leq F(0)\}$ . Então  $\alpha_d \geq \rho_a$  e  $\mathcal{R}_d^* = [0, \alpha_d]$ .*

Agora, juntando os lemas apresentados anteriormente, temos os elementos fundamentais para construirmos o procedimento de busca. A ideia é encontrar um ponto  $x^*$  que pertença ao conjunto Pareto-ótimo do problema 5.3, que domine  $x_0$  restrito para a direção de busca.

---

\*Uma função  $h : \mathbb{R} \rightarrow \mathbb{R}$  é quase-convexa se para todo  $x_1$  e  $x_2$  e  $\forall t \in (0, 1)$  tem-se  $h(tx_1 + (1-t)x_2) \leq \max\{h(x_1), h(x_2)\}$ .

Por comodidade, vamos denotar por  $F_0 = F(0)$ ,  $F_A = F(\alpha_A)$ ,  $F_B = F(\alpha_B)$ . E definiremos uma função vetorial  $C : \mathbb{R}^{m+3} \rightarrow \{0,1\}^6$  por

$$\begin{aligned} C(F_0, F_A, F_B) &= [C_1(F_0, F_A, F_B) \cdots C_6(F_0, F_A, F_B)] \\ &= [(F_0 \geq F_A)(F_0 \geq F_B)(F_A \geq F_B)(F_0 \leq F_A)(F_0 \leq F_B)(F_A \leq F_B)] \end{aligned}$$

com cada  $C_i$  um número binário em que "1" significa "verdadeiro" e "0" significa falso para resultado de cada comparação.

Existem três possíveis operações para contração do atual segmento verdadeiro, ou seja, o segmento no qual é conhecido que existe algum ponto que pertence ao conjunto  $\mathcal{R}^*$ , em uma busca linear baseada na avaliação da função nos pontos  $\alpha = 0$ ,  $\alpha = \alpha_A$  e  $\alpha = \alpha_B$ . Seja  $[\alpha_1, \alpha_2] \subset [0, 1]$  um atual segmento verdadeiro. Estas operações serão denotadas por  $D_1$ ,  $D_2$  e  $D_3$ , como indica na tabela 5.1.

Tabela 5.1: Operações de contração sobre o segmento-verdadeiro  $[\alpha_1, \alpha_2]$ .

	Operação de contração
$D_1$	rejeita $[0, \alpha_A] \cap [\alpha_1, \alpha_2]$
$D_2$	rejeita $[\alpha_A, \alpha_B] \cap [\alpha_1, \alpha_2]$
$D_3$	rejeita $[\alpha_B, 1] \cap [\alpha_1, \alpha_2]$

A tabela de decisões que aciona cada operação de contração é a tabela 5.2. Essa tabela maximiza a contração do segmento verdadeiro, sem permitir perdas de elementos do conjunto  $\mathcal{R}^*$ .

Tabela 5.2: Operações de contração sobre o segmento-verdadeiro  $[\alpha_1, \alpha_2]$ .

Condição	Operação
$C_1 \wedge (\sim C_6)$	$D_1$
$\sim C_1$	$D_2$
$\sim (C_1 \wedge C_2 \wedge C_3 \wedge (\sim C_4) \wedge (\sim C_5) \wedge (\sim C_6))$	$D_3$

O procedimento para descrever essas operações é mostrado na função kill-segmento.

Por fim, exibimos a função seção-áurea-multiobjetivo que apareceu no algoritmo 5 no começo da seção, cujo objetivo é encontrar um ponto  $\alpha^* \in \mathcal{R}_d^*$  ou, equivalentemente, uma solução  $x^* \in \mathbb{R}^n$  do problema 2.1.

Semelhante ao que foi feito na seção 2.2.2, usaremos a constante  $\gamma = 0.618$ , a qual é o inverso da razão áurea, para contração do segmento verdadeiro.

---

**Função kill-segmento**( $F_0, F_A, F_B$ )
 

---

**Saída:**  $D_1, D_2$  e  $D_3$

- 1  $C_1 \leftarrow (F_0 \geq F_A)$ ;
  - 2  $C_2 \leftarrow (F_0 \geq F_B)$ ;
  - 3  $C_3 \leftarrow (F_A \geq F_B)$ ;
  - 4  $C_4 \leftarrow (F_0 \leq F_A)$ ;
  - 5  $C_5 \leftarrow (F_0 \leq F_B)$ ;
  - 6  $C_6 \leftarrow (F_A \leq F_B)$ ;
  - 7  $D_1 \leftarrow C_1 \cdot \overline{C_6}$ ;
  - 8  $D_2 \leftarrow \overline{C_1}$ ;
  - 9  $D_3 \leftarrow \overline{C_1 \cdot C_2 \cdot C_3 \cdot \overline{C_4} \cdot \overline{C_5} \cdot \overline{C_6}}$ ;
- 

Na referência [36], foi provado que o número de iterações dessa função é limitado por  $\frac{\log \epsilon}{\log \gamma}$ , onde  $\epsilon > 0$  é a tolerância sobre a precisão do resultado.

**Observação 9.** *No capítulo 6, usaremos o procedimento de busca local no conjunto solução de cada algoritmo em avaliação.*

*Abusando da notação, manteremos os seus codinomes originais. O valor de proximidade com o conjunto Pareto-exato será  $10^{-3}$ .*

---

**Função** seção-áurea-multiobjetivo( $x \in \mathbb{R}^n$ ,  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $d \in \mathbb{R}^n$ ,  $\epsilon > 0$ )

---

**Saída:**  $x^*$

```

1   $\gamma \leftarrow 0.618;$ 
2   $\alpha_A \leftarrow x + (1 - \gamma) \cdot d;$ 
3   $\alpha_B \leftarrow x + \gamma \cdot d;$ 
4   $\alpha_r \leftarrow x + d;$ 
5   $F_0 \leftarrow f(\alpha_r);$ 
6   $F_A \leftarrow f(\alpha_A);$ 
7   $F_B \leftarrow f(\alpha_B);$ 
8  repita
9     $[D_1 D_2 D_3] \leftarrow \text{kill-segmento}(F_0, F_A, F_B);$ 
10   se  $D_1$  então
11      $x \leftarrow \alpha_A;$ 
12      $F_0 \leftarrow F_A;$ 
13      $d \leftarrow (\alpha_r - x);$ 
14      $\alpha_A \leftarrow \alpha_B;$ 
15      $F_A \leftarrow F_B;$ 
16      $\alpha_B \leftarrow x + \gamma d;$ 
17      $F_B \leftarrow f(\alpha_B);$ 
18   fim
19   se  $D_3 \cdot \overline{D_2}$  então
20      $\alpha_r = \alpha_B;$ 
21      $d \leftarrow (\alpha_r - x);$ 
22      $\alpha_B \leftarrow \alpha_A;$ 
23      $F_B \leftarrow F_A;$ 
24      $\alpha_A \leftarrow x + (1 - \gamma)d;$ 
25      $F_A \leftarrow f(\alpha_A);$ 
26   fim
27   se  $D_3 \cdot D_2$  então
28      $\alpha_r = \alpha_A;$ 
29      $d \leftarrow (\alpha_r - x);$ 
30      $\alpha_A \leftarrow x + (1 - \gamma)d;$ 
31      $F_A \leftarrow f(\alpha_A);$ 
32      $\alpha_B \leftarrow x + \gamma d;$ 
33      $F_B \leftarrow f(\alpha_B);$ 
34   fim
35 até  $\|d\|_2 > \epsilon;$ 
36  $x^* \leftarrow (x + r)/2;$ 

```

---

## 6 *Análise de desempenho*

Um dos objetivos da otimização multiobjetivo é encontrar um conjunto de soluções que melhor representem o conjunto Pareto-ótimo. Nos algoritmos evolutivos obtemos, usualmente, uma aproximação desse conjunto. Daí, é importante sabermos uma maneira de comparar essas soluções aproximadas dadas por algoritmos diferentes. Neste sentido, é interessante citar quando o assunto é avaliação de desempenho de algoritmos:

1. Quantidade de soluções encontradas;
2. Proximidade com o conjunto Pareto-ótimo;
3. Convergência das soluções para o conjunto Pareto-ótimo.
4. Distribuição das soluções ao longo do conjunto Pareto-ótimo;

Difícilmente uma única métrica irá medir vários ou todos os aspectos mencionados acima. Além disso, para avaliar alguns desses aspectos, faz-se necessário o conhecimento da localização do conjunto Pareto-ótimo exato, informação essa normalmente não disponível. Com relação ao item (1), nenhuma métrica será usada pois os algoritmos SSW e SSW-CMA não possuem controle para geração de soluções. Diferente do que ocorre com o NSGA2. Para o item (2), também não será utilizado nenhuma medida pois em todos os algoritmos usados estamos acoplado o procedimento de busca local proposto no capítulo 5.

As medidas que serão apresentadas a seguir se concentrarão nos itens (3) e (4). Mas focadamente no item(4), pois esse foi o principal problema apresentado pelo algoritmo SSW e que estamos propondo uma solução.

## 6.1 Medidas de avaliação

Apresentaremos três medidas com vistas a avaliação o desempenho dos algoritmos SSW, NSGA2 e SSW-CMA na questão espalhamento das soluções não-dominadas. As duas primeiras medidas foram escolhidas pelo baixo custo computacional e simplicidade de execução. A ultima medida foi utilizada por ser um referencial já bem posto na literatura.

### 6.1.1 Maximum Spread

Esta medida foi proposta por Zitzler et al.,[41] e mede quão bem um conjunto Pareto-ótimo exato é coberto por um conjunto de soluções não-dominadas. Isto é feito através do calculo do comprimento da diagonal das hipercaixas formada pelos valores extremos das funções objetivo observado no conjunto das soluções não-dominadas.

Usaremos esta métrica na sua versão normalizada, conforme foi apresentada em [2], definida na equação (6.1).

$$MS(C) = \sqrt{\frac{1}{m} \sum_{i=1}^m \frac{\max_{x \in C} f_i(x) - \min_{x \in C} f_i(x)}{F_i^{max} - F_i^{min}}}, \quad (6.1)$$

onde  $C$  é um conjunto com soluções não-dominadas,  $\max_{x \in C} f_i(x)$  e  $\min_{x \in C} f_i(x)$  são os valores de máximo e mínimo da  $i$ -ésima função objetivo no conjunto  $C$  e os valores  $F_i^{max}$  e  $F_i^{min}$  são os valores de máximo e mínimo de cada função objetivo  $f_i$  no conjunto Pareto-ótimo exato, respectivamente. Nota-se que o custo computacional desta medida é baixo.

O valor para  $MS(\cdot)$  desejado é próximo de 1, ou seja, espalhamento ideal das soluções no espaço de objetivos. Para valores próximos de 0, indica que tal conjunto  $C$  possui fraco espalhamento comparado ao espalhamento pretendido. Já para valores maiores que 1, indica uma dispersão excessiva das soluções. Este tipo de comportamento não é interessante pois provavelmente faz com que haja divergência das soluções das soluções encontradas dificultando o processo de otimização por incluir um comportamento cíclico forçando os algoritmos de otimização multiobjetivo explorar repetidamente regiões já visitadas do espaço de variáveis.

### 6.1.2 Contagem de esferas

Esta métrica foi proposta em [38] e posteriormente generalizado em [33]. Com o mesmo propósito que as outras medidas apresentadas, esta métrica serve para calcular a distribuição das soluções não-dominadas obtidas por um determinado algoritmo. A figura (6.1) mostra essa métrica para o caso bi-objetivo.

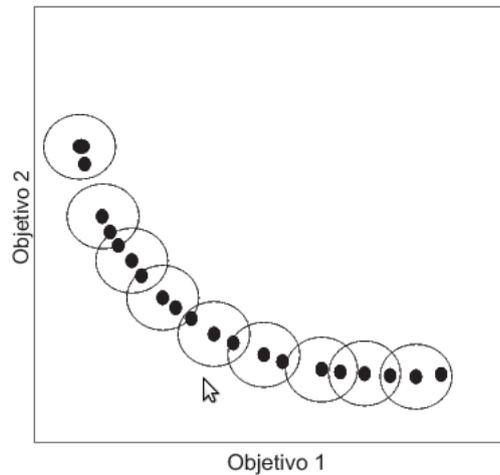


Figura 6.1: Contagem das esferas para o caso bi-objetivo. Note que temos 20 pontos no espaço de objetivos mas somente 9 esferas.

Para descrever essa medida, comecemos fixemos um valor  $r > 0$ . Seja  $m > 1$  a quantidade de objetivos de um determinado problema. Denotaremos por  $S^{m-1}(x, r)$  à esfera  $(m-1)$ -dimensional em  $\mathbb{R}^m$  com centro em  $x \in \mathbb{R}^m$  e raio  $r > 0$ . Dado um conjunto de soluções não-dominadas  $A$  geradas por determinado algoritmo, seja  $x_0$  um elemento escolhido aleatoriamente nesse conjunto. Defina o conjunto  $A_1$  por  $A \setminus S^{m-1}(x_0, r)$ . Seja  $x_1$  o elemento de  $A_1$  que realiza a menor distância em relação a  $x_0$ . Indutivamente, definimos  $x_i$ , para  $i > 1$ , de maneira que  $x_i \in A_i := A_{i-1} \setminus S^{m-1}(x_{i-1}, r)$ , com  $A_0 = A$ . O total de de esferas retiradas representa a contagem de esferas do conjunto  $A$ . Quanto maior for esse total de esferas, melhor será o conjunto.

Note que esta medida não requer de nenhum conhecimento prévio do conjunto Pareto-ótimo e que seu custo computacional de seu cálculo não varia expressivamente com o número de dimensões do espaço de objetivos. Por esse motivo, pode ser aplicada em problemas com quantidades arbitrárias de funções objetivo.

Convém comentar um pouco sobre a escolha do raio  $r > 0$ . Note que se  $r$  for muito grande, uma única esfera conterá todos os elementos. Por outro lado, se for muito pequeno, por exemplo, cada indivíduo será contado como uma esfera, tornando

a quantidade de esferas igual à quantidade de pontos. O ideal é tomar valores de  $r$  intermediários entre tais extremos. Assim, se denotarmos por  $d_{ij}$  a distância entre o indivíduo  $x_i$  e  $x_j$ , com  $i \neq j$ , consideraremos os seguintes valores de  $r$ :

- $r_1 = (\min_{i \neq j} d_{ij} + \max_{i \neq j} d_{ij})/5$ ;
- $r_2 = (\min_{i \neq j} d_{ij} + r_1)/5$ ;
- $r_3 = (r_1 + \max_{i \neq j} d_{ij})/5$ ;

Portanto, a contagem de esferas de um conjunto de soluções não-dominadas será a média aritmética da contagem de esferas com cada um  $r_i$ , para  $1 \leq i \leq 3$ , descrito acima.

### 6.1.3 S-Metric

Às vezes chamada de *indicador de hipervolume*, esta medida foi apresentada pela primeira vez em [42]. Essa métrica calcula o hipervolume de uma região multidimensional formada pela imagem do conjunto de soluções  $A$ , dado por um determinado algoritmo, e por um elemento referência. A figura (6.2), mostra essa medida para o caso bi-objetivo. Quanto maior for este hipervolume, melhor o conjunto de soluções.

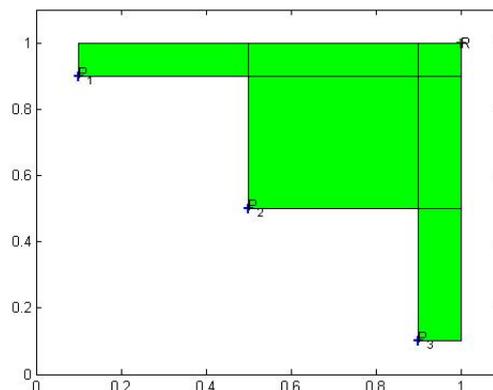


Figura 6.2: S-Metric para o caso bi-objetivo. Neste caso, calculamos a área da região dominada pelos pontos  $p_1$ ,  $p_2$  e  $p_3$  com ponto de referência  $R$  (parte verde da imagem!).

Como esta medida é sensível à escolha desse ponto de referência, alguns estudos já foram feitos sugerindo qual ponto escolher, ver por exemplo [1].

Existem alguns estudos que visam estabelecer cotas sobre a complexidade do cálculo do hipervolume. Por exemplo, em [8] é estabelecida a cota superior para problemas

com  $M > 2$  objetivos e  $n$  pontos de  $O(n^{M-2} \log(n))$ . Assim, para problemas com muitos objetivos o custo computacional é elevado. Para contornar esse problema, em [7] foi apresentado um algoritmo de busca que utiliza simulação Monte Carlos para aproximar o valor exato do hipervolume. Neste trabalho, usaremos tal algoritmo.

## 6.2 Avaliação de desempenho

Agora, uma vez estabelecidas as medidas que utilizaremos, vamos fazer avaliação de desempenho com os algoritmos SSW, SSW-CMA e NSGA2. Faremos isso em duas etapas. Em um primeiro momento, avaliaremos os algoritmos citados em problemas com duas e três funções objetivo. Posteriormente, verificaremos o desempenho dos algoritmos em problemas com muitos objetivos.

Em ambos os casos, os testes foram realizados em um Pc Intel Core i3 com 2gb de RAM em plataforma Linux 64bits. Os códigos foram escritos em Matlab na sua versão 2012a.

Em cada conjunto solução dos algoritmos citados acima, vamos aplicar a seção áurea vista no capítulo 5 para normalizar a aproximação. Deve-se notar que esse algoritmo de busca local irá obter uma solução Pareto-ótima exata (dentro da tolerância numérica estabelecida para a busca) para cada solução aproximada original. A nova solução necessariamente domina a solução aproximada original. Assim procedendo, pretendemos eliminar as diferenças que existem entre os conjuntos de soluções provenientes de diferentes algoritmos que estejam relacionadas com a distância do conjunto em relação ao conjunto Pareto-ótimo exato, preservando apenas as diferenças que dizem respeito à cobertura do conjunto exato pelo conjunto de amostras. Essa manobra torna possível a avaliação da característica de interesse dos conjuntos de soluções, assim transformados, utilizando a métrica da contagem de esferas.

Devido a aleatoriedade dos resultados, vamos executar cada algoritmo 30 vezes e cada execução com um máximo de avaliação de função de 30.000.

Além disso, em cada experimento, vamos apresentar também o gráfico **boxplot** para cada métrica de desempenho que resume as seguintes medidas estatísticas:

- mediana;
- quantis superior e inferior;

- os valores mínimo e máximo.

A caixa propriamente contém 50% das soluções. O limite superior da caixa indica o percentil de 75% dos dados e o limite inferior da caixa indica o percentil de 25%. A distância entre esses dois quantis é conhecida como inter-quartil. Os extremos do gráfico indicam os valores mínimo e máximo, a menos que valores outliers estejam presentes, nesse caso o gráfico se estende ao máximo de 1.5 vezes da distância inter-quartil. Com este modo de exibir as soluções poderemos ver graficamente a posição central dos dados (mediana) e a tendência das variáveis de busca.

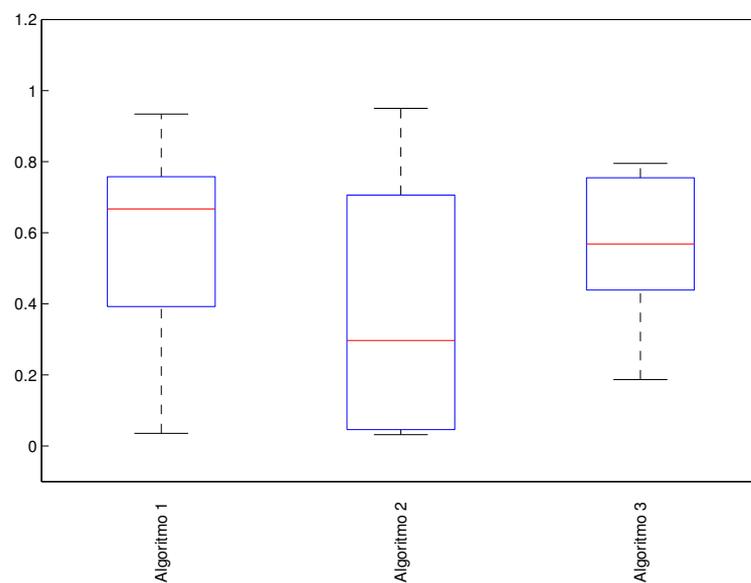


Figura 6.3: Gráfico boxplot com 3 algoritmos.

### 6.2.1 Problemas com poucos objetivos

Nesta seção, faremos uma avaliação dos algoritmos com problemas com duas e três funções objetivos. Como referência, utilizaremos as mesmas funções da seção 3.2. A finalidade dessa seção é indicar que o algoritmo SSW-CMA proposto traz significativas melhorias ao algoritmo SSW.

As figuras (6.4)-(6.5), (6.6)-(6.7) e (6.8)-(6.9) representam os resultados das medidas Maximum Spread, Contagem de Esferas e S-metric, respectivamente. Essas medidas indicam que o algoritmo SSW-CMA tem um bom desempenho para problemas com poucas funções objetivos, com resultados bem próximos aos do algoritmo NSGA-2.

A seguir, faremos testes com problemas em que o número de funções objetivo é maior que 3.

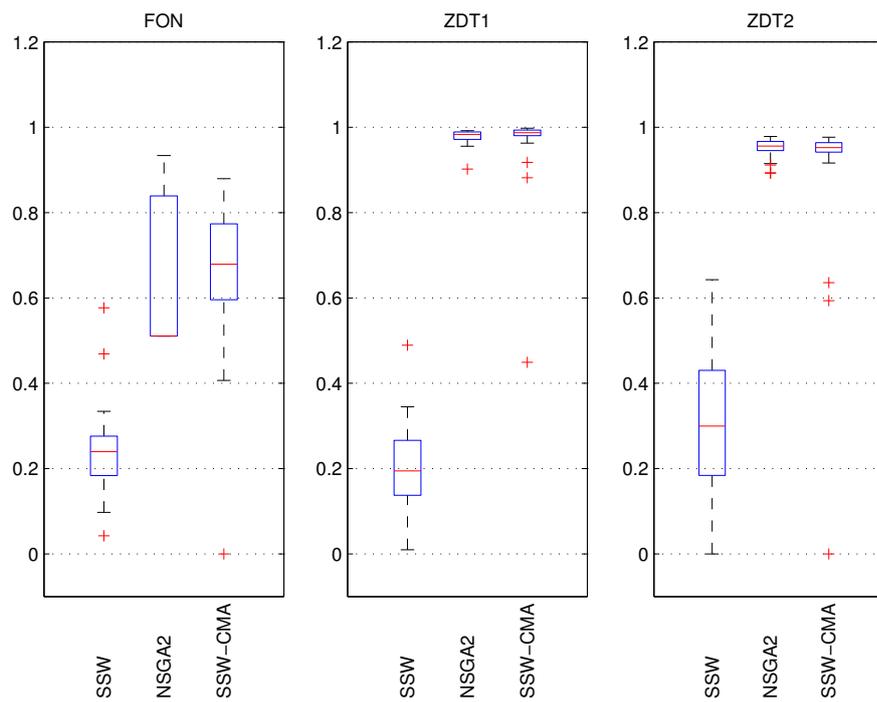


Figura 6.4: Resultados da medida Maximum Spread para problemas bi-objetivo.

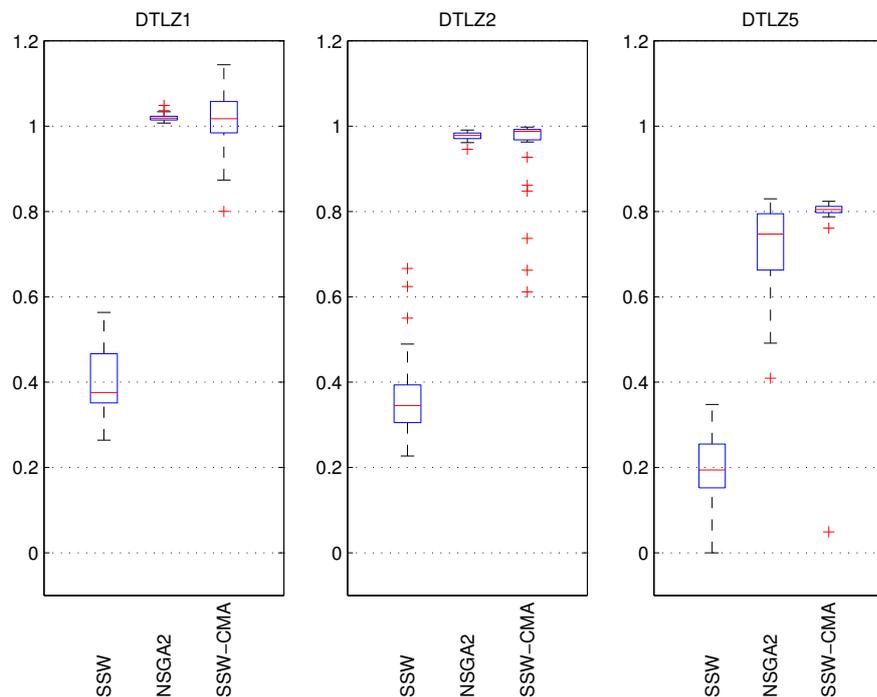


Figura 6.5: Resultados da medida Maximum Spread para problemas com três objetivos.

## 6.2.2 Problemas com muitos objetivos

Os algoritmos evolutivos usualmente têm trabalhado bem para dimensões de espaço de objetivos dois ou três (cf. seção anterior como referência.). Para problemas com

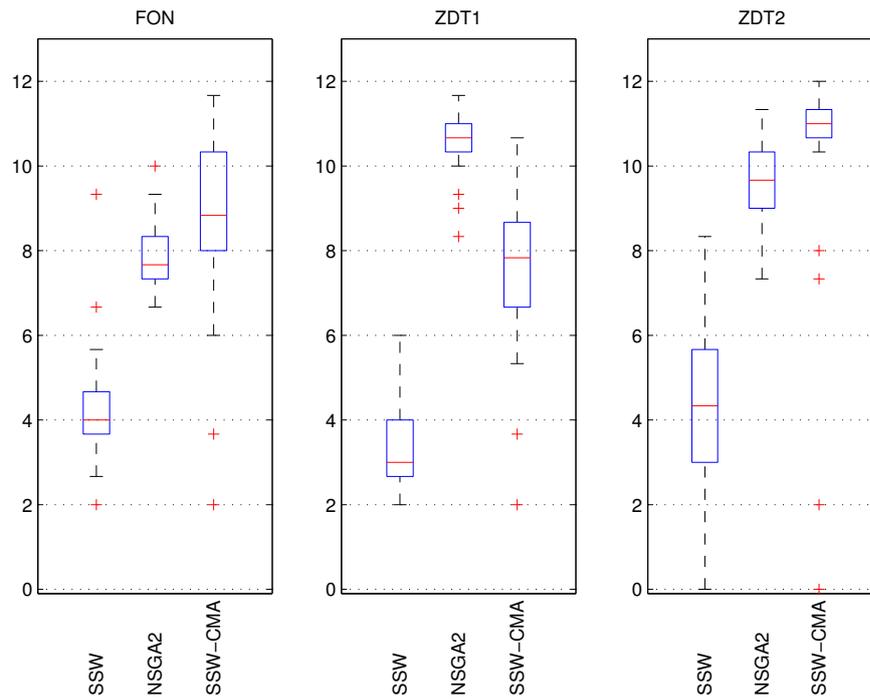


Figura 6.6: Resultados da medida Contagem de Esferas para problemas bi-objetivo.

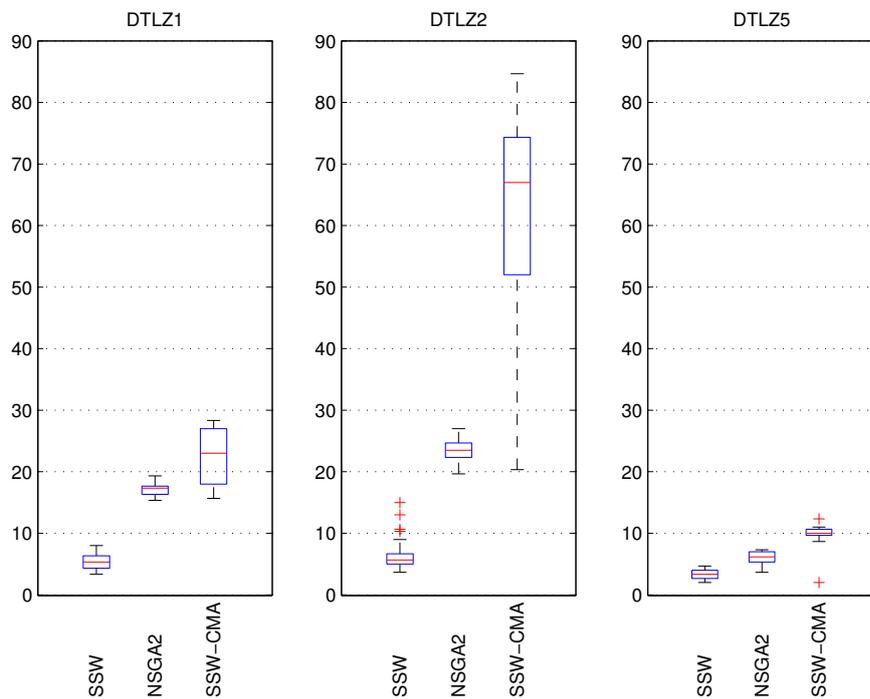


Figura 6.7: Resultados da medida Contagem de Esferas para problemas com três objetivos.

mais de três funções objetivos, é considerado como problemas com muitos objetivos. Esse tipo de problema tem sido alvo de vários estudos nos últimos anos (cf. [20, 29, 25]). Duas situações são interessantes nesse tipo de problema: i) A influência do aumento

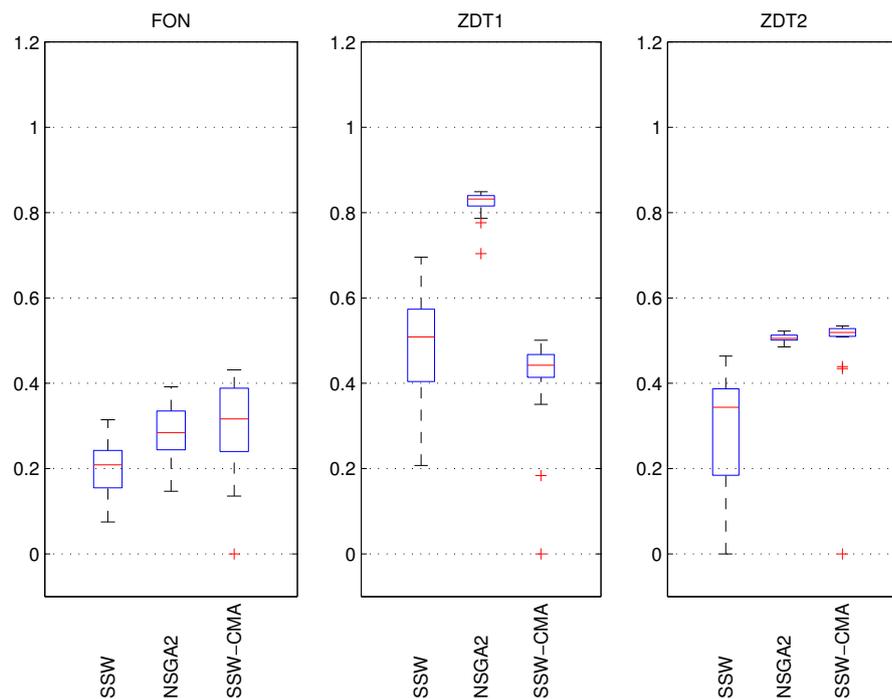


Figura 6.8: Resultados da medida S-Metric para problemas bi-objetivo.

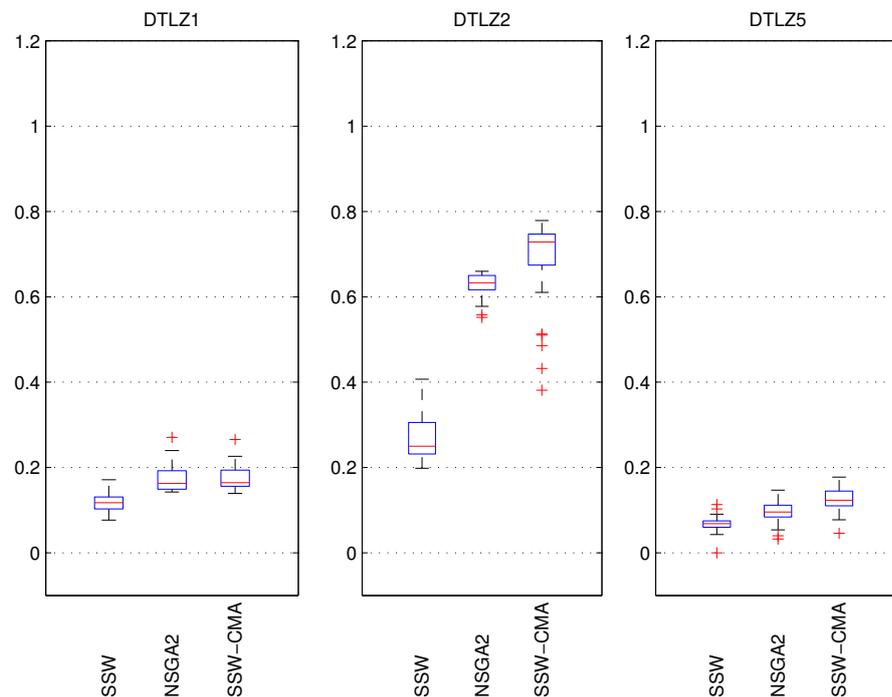


Figura 6.9: Resultados da medida S-Metric para problemas com três objetivos.

da quantidade de funções objetivos (cf. [31]), e ii) Diversidade das soluções (cf. [2]). O aumento da quantidade de funções objetivo prejudica a habilidade de encontrar soluções nos algoritmos de otimização multiobjetivo, e portanto interfere na diversidade das soluções. A deterioração do desempenho dos AEs pode ser explicada da seguinte

maneira: a pressão de seleção para a imagem do conjunto Pareto em algoritmos evolutivos se torna muito fraca sob a existência de muitos objetivos, porque quase todas as soluções são não-dominadas com respeito a muitos objetivos, ou seja, não é provável que exista Pareto dominância entre soluções para valores de  $m$  muito grande.

Em [20], foi feito um experimento para ilustrar o problema com muitos objetivos. O ilustração consiste em sortear aleatoriamente, via uma distribuição normal multivariada, 400 indivíduos e aplicar a relação de Pareto-dominância fazendo variar a quantidade de funções objetivo, em um qualquer problema com muitos objetivos.

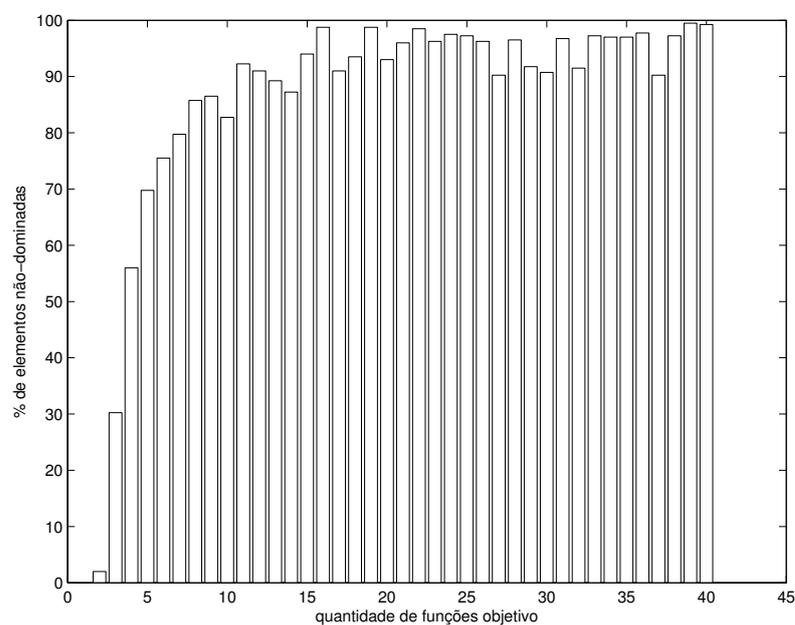


Figura 6.10: Impacto da quantidade de funções objetivo na relação de Pareto-dominância.

Podemos notar na figura (6.10) que para valores de  $m > 10$ , praticamente todos os elementos são considerados não-dominados, tornando a seleção via Pareto-dominância ineficaz. Além disso, quando temos poucas funções objetivo, por exemplo 2 ou 3 funções, temos uma baixa percentagem de elementos não-dominados. Esta dificuldade não será sentida no algoritmo SSW-CMA, pois mesmo usando a relação de Pareto dominância, a estrutura com direções de descida para todas as funções-objetivos conduzem a busca para as regiões factíveis do problema.

Nos experimentos dessa seção, consideraremos três problemas com  $m > 3$  funções objetivos e  $n = m + k - 1^*$  variáveis de decisão:

\*O valor de  $k$  é usualmente igual a 5.

## 1. Problem DTZL1

$$\begin{cases} f_1(x) &= (1/2)x_1x_2 \cdots x_{m-1}(1+g(x)) \\ f_2(x) &= (1/2)x_1x_2 \cdots (1-x_{m-1})(1+g(x)) \\ &\vdots \\ f_{m-1}(x) &= (1/2)x_1(1-x_2)(1+g(x)) \\ f_m(x) &= (1/2)(1-x_1)(1+g(x)) \end{cases}$$

onde cada  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  and

$$g(x) = \sum_{i=m}^n (x_i - 0.5)^2.$$

## 2. Problem DTZL2

$$\begin{cases} f_1(x) &= (1+g(x)) \cos\left(\frac{\pi \cdot x_1}{2}\right) \cdots \cos\left(\frac{\pi \cdot x_{m-2}}{2}\right) \cos\left(\frac{\pi \cdot x_{m-1}}{2}\right) \\ f_2(x) &= (1+g(x)) \cos\left(\frac{\pi \cdot x_1}{2}\right) \cdots \cos\left(\frac{\pi \cdot x_{m-2}}{2}\right) \sin\left(\frac{\pi \cdot x_{m-1}}{2}\right) \\ &\vdots \\ f_m(x) &= (1+g(x)) \sin\left(\frac{\pi \cdot x_1}{2}\right) \end{cases}$$

onde cada  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  and

$$g(x) = \sum_{i=m}^n (x_i - 0.5)^2.$$

## 3. Problem DTZL5

$$\begin{cases} f_1(x) &= (1+g(x)) \cos\left(\frac{\pi \cdot \theta_1}{2}\right) \cdots \cos\left(\frac{\pi \cdot \theta_{m-2}}{2}\right) \cos\left(\frac{\pi \cdot \theta_{m-1}}{2}\right) \\ f_2(x) &= (1+g(x)) \cos\left(\frac{\pi \cdot \theta_1}{2}\right) \cdots \cos\left(\frac{\pi \cdot \theta_{m-2}}{2}\right) \sin\left(\frac{\pi \cdot \theta_{m-1}}{2}\right) \\ &\vdots \\ f_m(x) &= (1+g(x)) \sin\left(\frac{\pi \cdot \theta_1}{2}\right) \end{cases}$$

onde cada  $f_i : \mathbb{R}^{m+1} \rightarrow \mathbb{R}$  e

$$\theta_1 = x_1, \theta_i = \frac{\pi}{4(1+g(x))} (1 + 2g(x)x_i), \text{ para } i = 2, 3, \dots, (m-1)$$

$$g(x) = \sum_{i=m}^n (x_i - 0.5)^2, x_i \in [0, 1], \text{ para } i = 1, 2, \dots, n.$$

Em cada problema, os valores de  $m$  serão  $\{5, 10, 20, 40\}$ . Para cada valor de  $m$ , repetimos a configuração feita com problemas com poucos objetivos: cada algoritmo será executado 30 vezes e cada execução com um máximo de avaliação de função de

30.000.

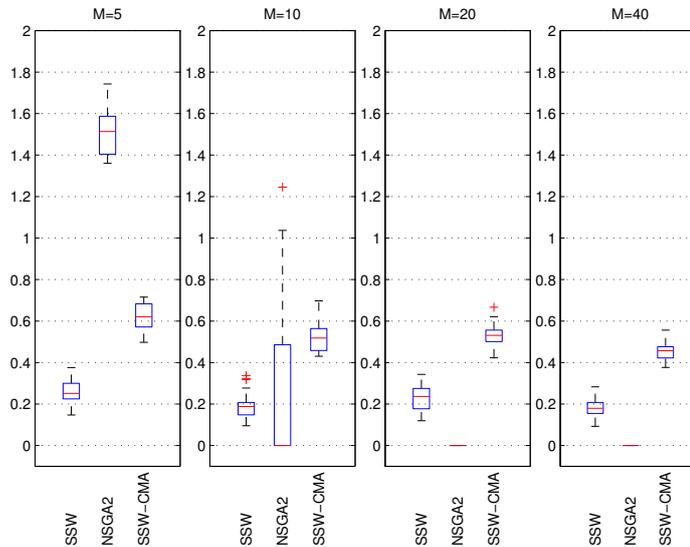


Figura 6.11: Resultados da medida Maximum Spread para problema DTLZ1.

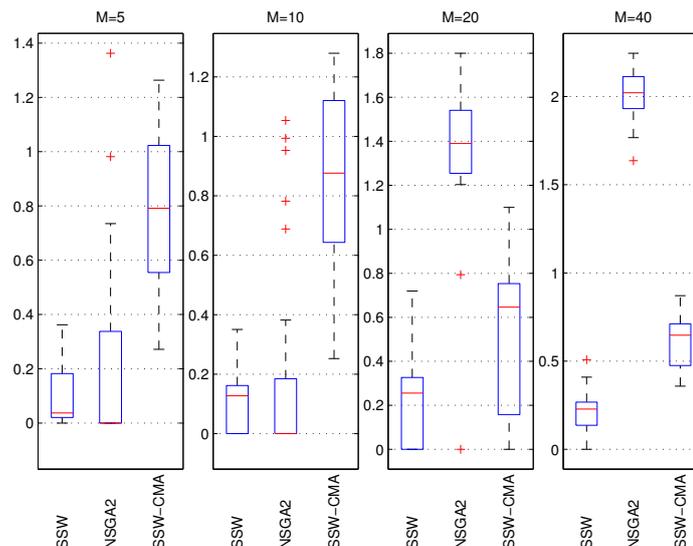


Figura 6.12: Resultados da medida Maximum Spread para problema DTLZ2.

As figuras (6.11)-(6.19) mostram os resultados das medidas estabelecidas na seção 6.1. Em todas as avaliações, quando tomamos o valor de  $m$  elevado (20 ou 40, por exemplo), percebemos a dificuldade do algoritmo NSGA2 de encontrar soluções Pareto-ótimas em cada problema. Dificuldade esta, já esperada para esse tipo de algoritmo. Em contra-partida, o algoritmo SSW-CMA mantém apresenta bons resultados nessas mesmas situações.

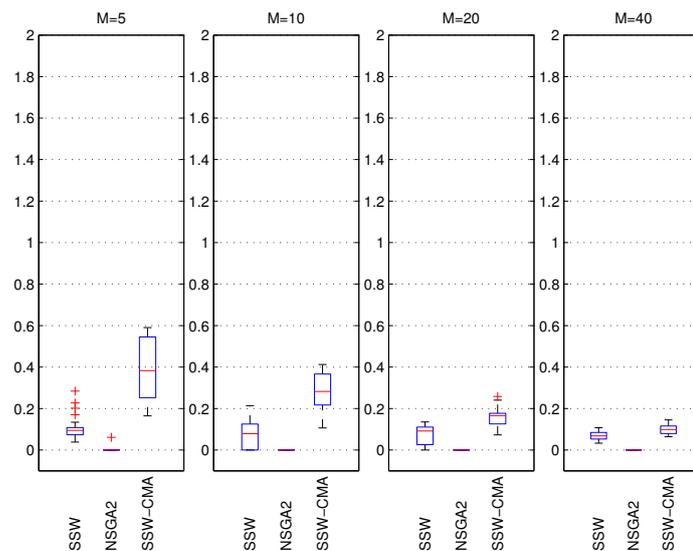


Figura 6.13: Resultados da medida Maximum Spread para problema DTLZ5.

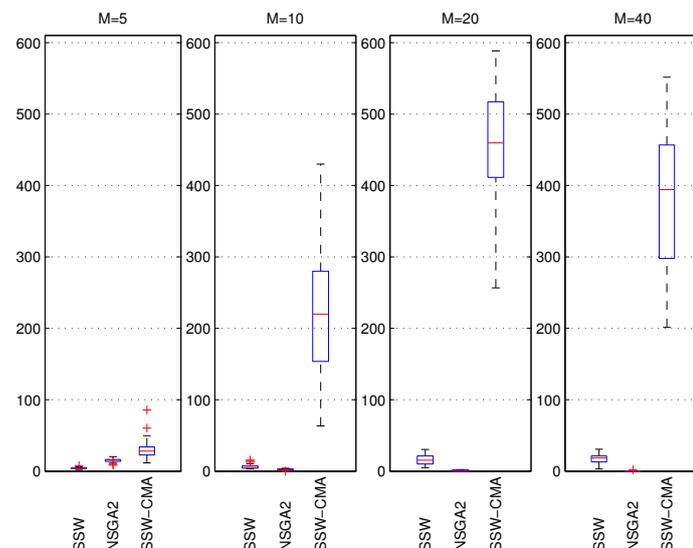


Figura 6.14: Resultados da medida Contagem de Esferas para problema DTLZ1.

### Discussão sobre a proximidade das soluções

Além das medidas que avaliam a distribuição das soluções, vamos analisar o valor  $\|q(\cdot)\|^2$ , onde  $q: \mathbb{R}^n \rightarrow \mathbb{R}^n$  é a função definida pelo teorema 2.2.1. Por meio desse teorema, tal valor nos dá uma noção de proximidade das soluções em relação ao conjunto Pareto-ótimo. Com relação a proximidade de uma solução em relação ao conjunto Pareto-ótimo, vamos definir o seguinte:

**Definição 6.1.** Dizemos que um indivíduo  $\omega \in \mathbb{R}^n$  está  $\epsilon$ -próximo do conjunto Pareto-ótimo se  $\|q(\omega)\|^2 < \epsilon$ .

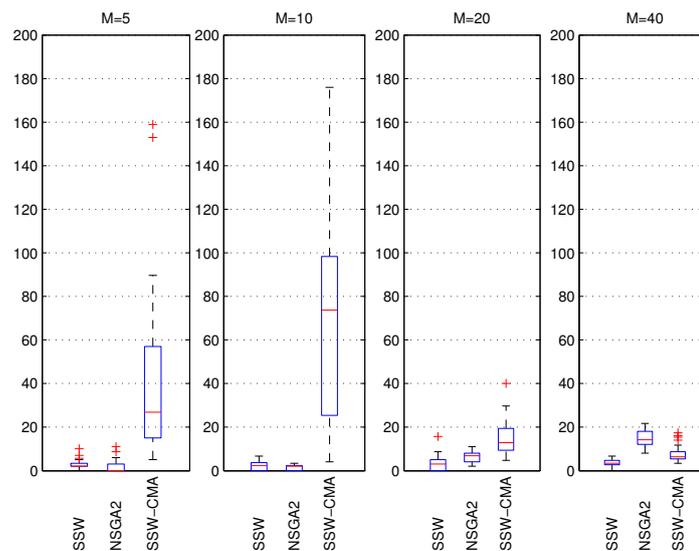


Figura 6.15: Resultados da medida Contagem de Esferas para problema DTLZ2.

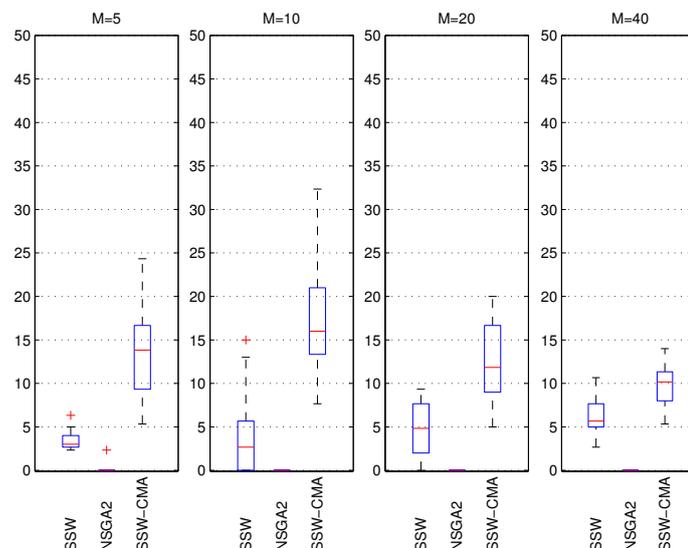


Figura 6.16: Resultados da medida Contagem de Esferas para problema DTLZ5.

Nas tabelas (6.1)-(6.6) mostramos o valor médio de  $\|q(\cdot)\|^2$  considerando todas as 30 execuções dos algoritmos<sup>†</sup> NSGA2 e SSW-CMA, para diferentes quantidades de funções-objetivo, antes e depois de aplicar a busca local (BL) proposta no capítulo 5. Nota-se que quando aumentamos o valor de  $m$ , o algoritmo NSGA2 gera muitas soluções não-dominadas, porém distantes do conjunto Pareto-ótimo. Diferente do que acontece com o algoritmo SSW-CMA, que continua gerando soluções próximas do conjunto Pareto.

Na figura (6.20), detalhamos os valores médio de  $\|q(\cdot)\|^2$ , em escala logarítmica,

<sup>†</sup>Não faremos comparações com o algoritmo SSW nesta seção pois a versão SSW-CMA possui o mesmo mecanismo para se aproximar do conjunto Pareto-ótimo.

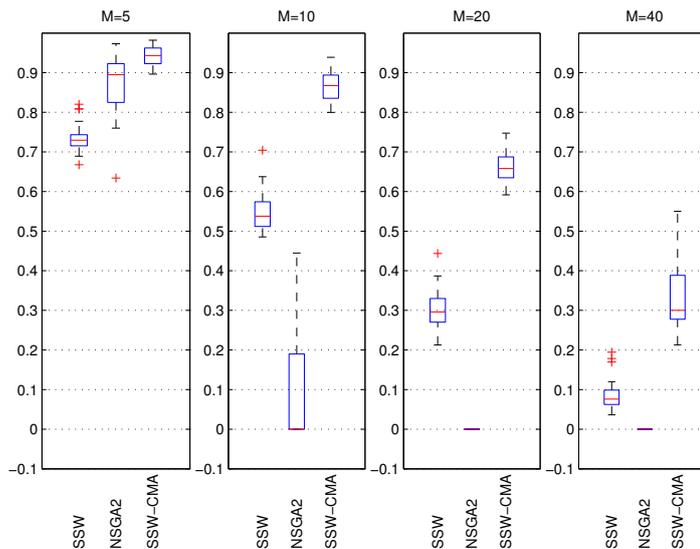


Figura 6.17: Resultados da medida S-metric para problema DTLZ1.

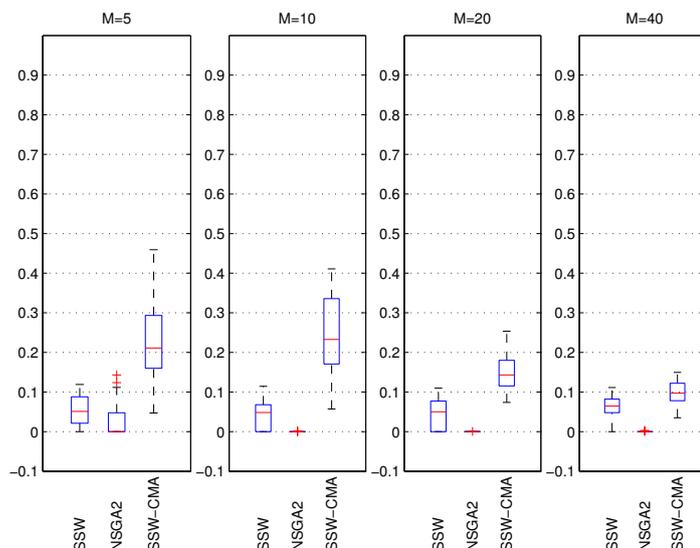


Figura 6.18: Resultados da medida S-metric para problema DTLZ2.

Tabela 6.1: Valor médio de  $\|q(x)\|^2$  nas soluções do algoritmo NSGA-2 para o problema DTLZ1.

m	Antes da BL	Total de Soluções	Depois da BL	Total de Soluções
5	$9.22 \cdot 10^{-2}$	3000	$5.22 \cdot 10^{-3}$	729
10	$7.57 \cdot 10^{-1}$	3000	$4.67 \cdot 10^{-3}$	39
20	$4.67 \cdot 10^{-1}$	3000	$2.68 \cdot 10^{-3}$	9
40	$3.11 \cdot 10^{-1}$	3000	$1.22 \cdot 10^{-3}$	3

antes de aplicar o procedimento de busca local, em cada algoritmo, para  $m = 40$ . Nota-se que, antes de aplicar a busca local, o algoritmo SSW-CMA apresenta resultados bem melhores quando comparado com o NSGA2.

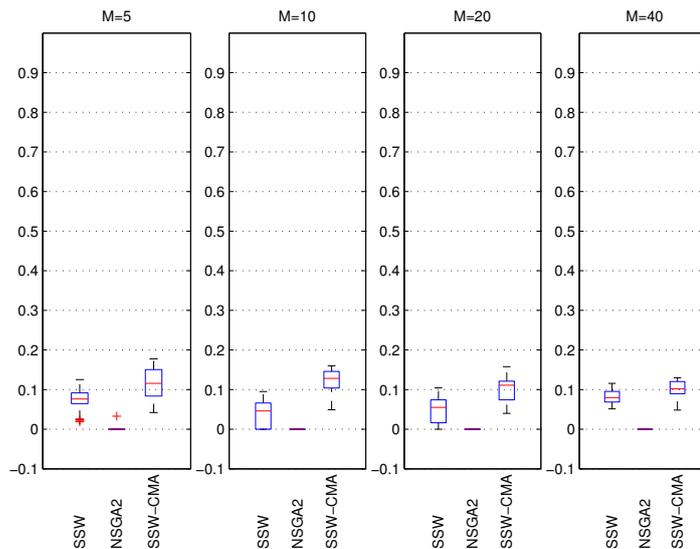


Figura 6.19: Resultados da medida S-metric para problema DTLZ5.

Tabela 6.2: Valor médio de  $\|q(x)\|^2$  nas soluções do algoritmo SSW-CMA para o problema DTLZ1 .

m	Antes da BL	Total de Soluções	Depois da BL	Total de Soluções
5	$1.41 \cdot 10^{-2}$	4883	$3.31 \cdot 10^{-3}$	4833
10	$4.77 \cdot 10^{-3}$	26227	$4.77 \cdot 10^{-3}$	26194
20	$1.27 \cdot 10^{-5}$	41672	$1.27 \cdot 10^{-5}$	41654
40	$6.08 \cdot 10^{-13}$	22011	$6.08 \cdot 10^{-13}$	21996

Tabela 6.3: Valor médio de  $\|q(x)\|^2$  nas soluções do algoritmo NSGA-2 para o problema DTLZ2 .

m	Antes da BL	Total de Soluções	Depois da BL	Total de Soluções
5	1.42	3000	$2.41 \cdot 10^{-3}$	95
10	1.63	3000	$1.08 \cdot 10^{-3}$	39
20	$7.83 \cdot 10^{-1}$	3000	$8.36 \cdot 10^{-3}$	186
40	$1.12 \cdot 10^{-1}$	3000	$2.20 \cdot 10^{-3}$	646

Tabela 6.4: Valor médio de  $\|q(x)\|^2$  nas soluções do algoritmo SSW-CMA para o problema DTLZ2 .

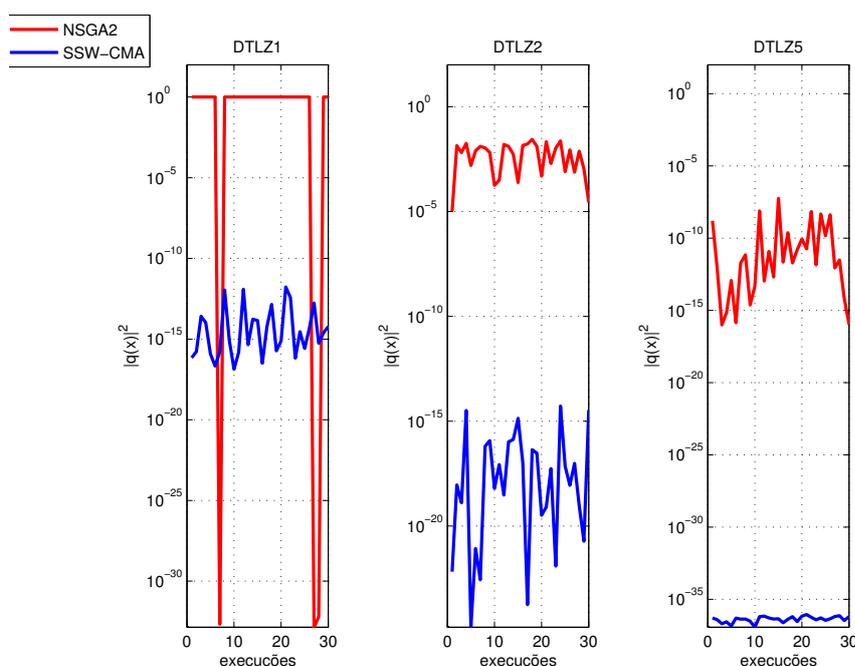
m	Antes da BL	Total de Soluções	Depois da BL	Total de Soluções
5	$1.46 \cdot 10^{-2}$	37169	$1.69 \cdot 10^{-3}$	36716
10	$1.44 \cdot 10^{-3}$	28454	$1.05 \cdot 10^{-3}$	27466
20	$3.11 \cdot 10^{-5}$	6270	$2.92 \cdot 10^{-5}$	5465
40	$4.21 \cdot 10^{-14}$	7350	$5.10 \cdot 10^{-14}$	6267

Tabela 6.5: Valor médio de  $\|q(x)\|^2$  nas soluções do algoritmo NSGA-2 para o problema DTLZ5 .

m	Antes da BL	Total de Soluções	Depois da BL	Total de Soluções
5	1.54	3000	$2.33 \cdot 10^{-3}$	9
10	$2.42 \cdot 10^{-1}$	3000	$1.88 \cdot 10^{-3}$	7
20	$3.97 \cdot 10^{-3}$	3000	$3.97 \cdot 10^{-3}$	15
40	$4.12 \cdot 10^{-10}$	3000	$4.12 \cdot 10^{-10}$	11

Tabela 6.6: Valor médio de  $\|q(x)\|^2$  nas soluções do algoritmo SSW-CMA para o problema DTLZ5 .

m	Antes da BL	Total de Soluções	Depois da BL	Total de Soluções
5	$1.06 \cdot 10^{-3}$	11427	$1.06 \cdot 10^{-3}$	8853
10	$1.97 \cdot 10^{-8}$	19942	$1.97 \cdot 10^{-8}$	15568
20	$8.42 \cdot 10^{-19}$	18642	$8.42 \cdot 10^{-19}$	12667
40	$1.49 \cdot 10^{-35}$	10333	$1.49 \cdot 10^{-35}$	8002

Figura 6.20: Gráfico com os valores de  $\|q(\cdot)\|^2$  médio, em escala logarítmica, para  $m = 40$ .

### 6.3 Resultados

De maneira geral, as medidas de avaliação de desempenho indicam que o algoritmo SSW-CMA é significativamente melhor que seu predecessor, o algoritmo SSW.

Quando comparado com o algoritmo NSGA2, ainda mantém índices razoavelmente melhores.

Dois motivos justificam esse feito. A primeira delas, vindo do algoritmo SSW que o originou, foi usar a direção de descida " $-q$ ", apresentado no teorema 2.2.1, tornando a convergência para o conjunto Pareto-ótimo eficiente. A outra razão vem do fato de usarmos operadores de mutações com adaptação da matriz de covariância para distribuir as soluções. Tais ideias, propostas inicialmente em [15], nos trouxeram resultados relevantes. Nós usamos as mesmas sugestões deste artigo, mudando o seguinte:

1. Aplicamos a mutação em um indivíduo gerado pelo SSW quando ele está  $\epsilon$ -próximo do conjunto Pareto, ou seja,  $\|q(x)\| < \epsilon$ ;
2. Prosseguimos a execução do SSW com o elemento  $x_{mean}$  obtido da operação acima, conforme apresentamos no algoritmo (4).

## 6.4 Conclusões

Este estudo apresenta um novo algoritmo para problemas com muitos objetivos. A proposta é construída como combinação de dois algoritmos antigos para otimização multiobjetivo: 1) um algoritmo baseado na abordagem de equações diferenciais estocásticas (o algoritmo SSW), apresentado em [30], e 2) um algoritmo CMA-ES, proposto em [15]. O novo herda a propriedade de encontrar pontos Pareto-ótimos para problemas com muitos objetivos com baixo custo computacional vindos do algoritmo SSW e faz uso da habilidade do CMA-ES de encontrar um bom espalhamento de soluções pertencendo ao conjunto Pareto-ótimo.

## APÊNDICE A – Alguns lemas técnicos

Escolha  $\bar{t} > 0$ . Seja  $C[0, \bar{t}]$  o conjunto das funções contínuas  $h : [0, \bar{t}] \rightarrow \mathbb{R}^n$ ,  $n \in \mathbb{N}$ .

Usando a norma

$$\|h\|_{(K)} := \max_{0 \leq t \leq \bar{t}} (e^{-Kt} \|h(t)\|_2) \quad (\text{A.1})$$

temos um espaço de Banach  $(C[0, \bar{t}], \|\cdot\|_{(K)})$  para cada  $K \in \mathbb{R}$ .

Seja  $z : \mathbb{R}^n \rightarrow \mathbb{R}^n$  uma função Lipschitz com constante de Lipschitz  $L > 0$ . Para cada  $\omega \in \Omega$ , consideremos o operador  $T_\omega : C[0, \bar{t}] \rightarrow C[0, \bar{t}]$ ,

$$(T_\omega h)(t) := \omega(t) - \int_0^t z(h(\tau)) d\tau, \quad 0 \leq t \leq \bar{t}.$$

**Teorema A.1.** Para cada  $\omega \in \Omega$ ,

1.  $\|T_\omega h - T_\omega \bar{h}\|_{(2L)} \leq \frac{1}{2} \|h - \bar{h}\|_{(2L)}$  para todo  $h, \bar{h} \in C[0, \bar{t}]$ .
2. A equação  $T_\omega h = h$  tem uma única solução  $\hat{h} \in C[0, \bar{t}]$ .
3. Escolha  $h_0 \in C[0, \bar{t}]$ . Seja  $h_{j+1} = T_\omega h_j$ ,  $j \in \mathbb{N}_0$  então

$$\lim_{j \rightarrow \infty} \|h_j - \hat{h}\|_{(2L)} = 0$$

*Demonstração.*

$$\begin{aligned}
 \|(T_\omega h)(t) - (T_\omega \bar{h})(t)\|_2 &= \left\| \int_0^t z(h(\tau))d\tau - \int_0^t z(\bar{h}(\tau))d\tau \right\|_2 \\
 &= \left\| \int_0^t (z(h(\tau)) - z(\bar{h}(\tau)))d\tau \right\|_2 \\
 &\leq \int_0^t \|z(h(\tau)) - z(\bar{h}(\tau))\|_2 e^{-2L\tau} e^{2L\tau} d\tau \\
 &\leq L \int_0^t \|h(\tau) - \bar{h}(\tau)\|_2 e^{-2L\tau} e^{2L\tau} d\tau \\
 &\leq L \int_0^t \max_{0 \leq s \leq \tau} (\|h(s) - \bar{h}(s)\|_2 e^{-2Ls}) e^{2L\tau} d\tau \\
 &\leq L \|h - \bar{h}\|_{(2L)} \int_0^t e^{2L\tau} d\tau \\
 &\leq (1/2) \|h - \bar{h}\|_{(2L)} e^{2Lt} \text{ para todo } t \in [0, \bar{t}], \omega \in \Omega
 \end{aligned}$$

Portanto, para todo  $t \in [0, \bar{t}]$ ,  $\omega \in \Omega$ ,

$$\|(T_\omega h)(t) - (T_\omega \bar{h})(t)\|_2 e^{-2Lt} \leq (1/2) \|h - \bar{h}\|_{(2L)}$$

Daí,

$$\|T_\omega h - T_\omega \bar{h}\|_{(2L)} \leq \frac{1}{2} \|h - \bar{h}\|_{(2L)},$$

para todo  $t \in [0, \bar{t}]$ ,  $\omega \in \Omega$ .

As provas dos itens (2) e (3) decorrem do teorema do ponto fixo de Banach.  $\square$

**Lema A.1.** *Seja  $h : \mathbb{R}^n \rightarrow \mathbb{R}^n$  uma função localmente Lipschitz. Então, para cada  $k \in \mathbb{N}$  e  $x_0 \in \mathbb{R}^n$ , a função  $\hat{h} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  dada por*

$$\hat{h}(x) := \begin{cases} h(x), & \text{se } \|x - x_0\|_2 < k \\ h\left(\frac{k(x - x_0)}{\|x - x_0\|_2}\right), & \text{caso contrário.} \end{cases}$$

*é Lipschitz.*

*Demonstração.* Notemos que como  $h$  é localmente Lipschitz então  $h|_{B(x_0; k)}$  é Lipschitz.

Denotaremos a constante de Lipschitz de  $h$  na bola  $B(x_0; k)$  por  $L_{x_0, k}$ .

Para cada  $k \in \mathbb{N}$  e  $x_0 \in \mathbb{R}^n$ , existe  $p > 0$  tal que a bola  $B(0; p)$  contém as bolas  $B(x_0; k)$

e  $B(0;k)$  e para todo  $x \in B(x_0;k)$  e  $y \notin B(0;p)$  tem-se  $\|x_0\|_2 \leq \|x - y\|_2$ .

Seja  $L = 2 \max(L_{0,k}, L_{x_0,k}, L_{0,p})$ . Vamos mostrar que para todo  $x, y \in \mathbb{R}^n$  temos que

$$\|\hat{h}(x) - \hat{h}(y)\|_2 \leq L\|x - y\|_2$$

Existem 3 possibilidades:

i)  $x, y \in B(x_0;k)$ ;

ii)  $x, y \notin B(x_0;k)$ ;

iii)  $x \in B(x_0;k)$  e  $y \notin B(x_0;k)$ .

O caso i) é trivial. No segundo caso, observemos que  $\frac{k(x - x_0)}{\|x - x_0\|_2}, \frac{k(y - x_0)}{\|y - x_0\|_2} \in B(0;k)$  e como  $k < \|x - x_0\|_2$  e  $k < \|y - x_0\|_2$  segue que

$$\begin{aligned} \|\hat{h}(x) - \hat{h}(y)\|_2 &= \left\| h\left(\frac{k(x - x_0)}{\|x - x_0\|_2}\right) - h\left(\frac{k(y - x_0)}{\|y - x_0\|_2}\right) \right\|_2 \\ &\leq L_{0,p} \left\| \frac{k(x - x_0)}{\|x - x_0\|_2} - \frac{k(y - x_0)}{\|y - x_0\|_2} \right\|_2 \\ &\leq L_{0,p} \|(x - x_0) - (y - x_0)\|_2 = L_{0,p} \|x - y\|_2 \\ &\leq L \|x - y\|_2 \end{aligned}$$

No ultimo caso, se  $y \in B(0;p) \setminus B(x_0;k)$  ainda teremos  $x, y \in B(0;p)$  e a condição se verifica trivialmente. Se  $y \notin B(0;p)$  então

$$\|\hat{h}(x) - \hat{h}(y)\|_2 = \left\| h(x) - h\left(\frac{k(y - x_0)}{\|y - x_0\|_2}\right) \right\|_2$$

Desde que  $x, \frac{k(y - x_0)}{\|y - x_0\|_2} \in B(0;p)$  temos que

$$\|\hat{h}(x) - \hat{h}(y)\|_2 \leq L_{0,p} \left\| x - \frac{k(y - x_0)}{\|y - x_0\|_2} \right\|_2$$

Como  $\frac{k}{\|y - x_0\|_2} < 1$  tem-se

$$\begin{aligned} \|\hat{h}(x) - \hat{h}(y)\|_2 &\leq L_{0,p}(\|x - (y - x_0)\|_2) \\ &\leq L_{0,p}(\|x - y\|_2 + \|x_0\|_2) \\ &\leq L_{0,p}(\|x - y\|_2 + \|x - y\|_2) \\ &\leq L\|x - y\|_2 \end{aligned}$$

□

**Lema A.2.** *Seja  $\{r_k\}_{k \in \mathbb{N}}$  uma sequência de variáveis aleatórias  $n$ -dimensional definidas em  $\Omega$ . Assuma que  $\lim_{k \rightarrow \infty} r_k(w)$  existe em  $\mathbb{R}^n$ , para cada  $w \in \Omega$ , então  $r : \Omega \rightarrow \mathbb{R}^n$  dado por  $r(w) = \lim_{k \rightarrow \infty} r_k(w)$  é uma variável aleatória  $n$ -dimensional.*

*Demonstração.* Como  $\limsup r_k$  e o  $\liminf r_k$  são funções mensuráveis e  $\lim_{k \rightarrow \infty} r_k(w)$  existe para cada  $w \in \Omega$  então  $r(w) = \lim_{k \rightarrow \infty} r_k(w)$  é mensurável e portanto variável aleatória. □

**Teorema A.2.** *Seja  $\{X_t\}$  um processo estocástico  $n$ -dimensional então  $Y_t : \Omega \rightarrow \mathbb{R}^n$  dado por  $Y_t(w) = \int_0^t f(X_\tau(w))d\tau$  é um processo estocástico contínuo  $n$ -dimensional para cada função contínua  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ .*

*Demonstração.* Considere  $t_k := k\frac{t}{m}$ ,  $k = 0, 1, \dots, m$  e  $Y_t^m : \Omega \rightarrow \mathbb{R}^n$  dado por

$$Y_t^m(w) = \sum_{k=1}^m f(X_{t_k}(w))(t_k - t_{k-1})$$

Claro que  $Y_t^m$  é uma variável aleatória para cada  $m \in \mathbb{N}$ ,  $t \geq 0$ . Para cada  $w \in \Omega$  fixo,  $t \geq 0$ , temos que

$$\lim_{m \rightarrow \infty} Y_t^m(w) = \int_0^t f(X_\tau(w))d\tau = Y_t(w).$$

Pelo lema (A.2),  $Y_t$  é uma variável aleatória para cada  $t \geq 0$ . A continuidade dos caminhos de  $\{Y_t\}$  é dada pela definição de  $Y_t$ . □

## *APÊNDICE B – Distribuição normal multivariada $\mathcal{N}(\mu, C)$*

Faremos uma pequena abordagem sobre distribuição normal multivariada. Para este fim, vamos primeiro pensar no caso unidimensional.

Seja  $x \in \mathbb{R}$  uma variável aleatória contínua. A esperança de  $x$ , denotada por  $E(x)$ , é dada por:

$$E(x) = \int xp(x)dx, \quad (\text{B.1})$$

onde  $p(x)$  é a função densidade de probabilidade (FDP) de  $x$ . A variância de  $x$ , denotada por  $var(x)$  é dada por

$$var(x) = E((x - E(x))^2)$$

**Lema B.1.** *Para uma variável aleatória contínua  $x \in \mathbb{R}$ , temos que  $var(x) = E(x^2) - E(x)^2$ .*

*Demonstração.* Como  $E(\cdot)$  é um operador linear, segue da definição que:

$$\begin{aligned} var(x) &= E((x - E(x))^2) \\ &= E((x^2 - 2xE(x) + E(x)^2)) \\ &= E(x^2) - 2E(x)E(x) + E(x)^2 \\ &= E(x^2) - E(x)^2. \end{aligned}$$

□

O desvio padrão de  $x$  é dado por  $\sigma(x) = \sqrt{var(x)}$ . Dada outra variável aleatória

$y \in \mathbb{R}$ , a covariância de  $x$  e  $y$  é dada por

$$\text{cov}(x, y) = E((x - E(x))(y - E(y))) \quad (\text{B.2})$$

**Lema B.2.** Para uma variável aleatória contínua  $x \in \mathbb{R}$ , temos que  $\text{cov}(x, y) = E(xy) - E(x)E(y)$ .

*Demonstração.* Usando o fato que  $E(\cdot)$  é um operador linear, segue da definição que:

$$\begin{aligned} \text{cov}(x, y) &= E((x - E(x))(y - E(y))) \\ &= E(xy - xE(y) - yE(x) + E(x)E(y)) \\ &= E(xy) - E(x)E(y) - E(y)E(x) + E(x)E(y) \\ &= E(xy) - E(x)E(y). \end{aligned}$$

□

Com os lemas provados anteriormente, podemos encontrar uma expressão simples para  $\text{var}(x + y)$ , a saber

$$\text{var}(x + y) = \text{var}(x) + \text{var}(y) + 2\text{cov}(x, y).$$

**Definição B.1.** Uma variável aleatória  $x \in \mathbb{R}$  é dada pela distribuição normal com parâmetros  $\mu$  e  $\sigma^2$  se a FDP de  $x$  é dada por

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right), \quad (\text{B.3})$$

onde  $\mu$  é a esperança de  $x$  e  $\sigma^2$  é a variância de  $x$ .

Denotaremos por  $\mathcal{N}(\mu, \sigma^2)$  à todas as variáveis aleatórias dada pela distribuição normal com média  $\mu$  e variância  $\sigma^2$ .

Agora, vamos considerar variáveis aleatórias  $n$ -dimensionais. Seja  $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$  uma variável aleatória contínua. A esperança de  $x$  é definida de maneira análoga:

$$E(x) = (E(x_1), E(x_2), \dots, E(x_n)) \in \mathbb{R}^n \quad (\text{B.4})$$

A matriz de covariâncias é dada por  $\text{Cov}(x) = E((x - \mu)(x - \mu)^T) \in \mathbb{R}^{n \times n}$ , onde  $\mu = E(x) \in \mathbb{R}^n$ .

É fácil ver que as entradas da matriz  $Cov(x)$  são os elementos  $cov(x_i, x_j)$ , onde  $i$  e  $j$  representam a linha e coluna dessa matriz, respectivamente. Outro fato simples de perceber é que a matriz de covariância é simétrica e definida positiva. Além disso,

$$Cov(x) = E(xx^T) - E(x)E(x^T) \quad (\text{B.5})$$

**Definição B.2.** Uma variável aleatória contínua  $x \in \mathbb{R}^n$  é dita normalmente distribuída com parâmetros  $\mu$  e  $C$ , o qual denotaremos por  $x \sim \mathcal{N}(\mu, C)$ , se a FDP de  $x$  é dada por

$$p(x) = \frac{1}{(2\pi)^{n/2}|C|^{1/2}} \exp\left(-\frac{(x-\mu)^T C^{-1}(x-\mu)}{2}\right), \quad (\text{B.6})$$

onde  $\mu \in \mathbb{R}^n$  é média e  $C \in \mathbb{R}^{n \times n}$  é a matriz de covariâncias de  $x$ .

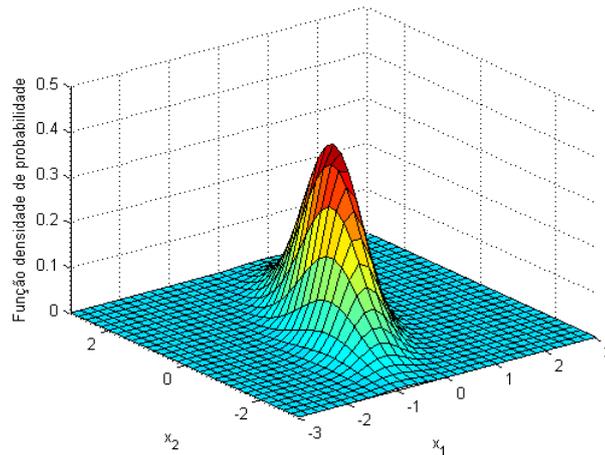


Figura B.1: Exemplo mostra a função densidade de probabilidade para uma distribuição normal bivariada com desvio padrão desigual.

Decorre da definição que distribuição  $\mathcal{N}(\mu, C)$  é unicamente determinada por sua média  $\mu \in \mathbb{R}^n$  e sua matriz de covariância definida positiva e simétrica  $C \in \mathbb{R}^{n \times n}$ .

A autodecomposição de  $C$  é denotada por  $C = B \cdot D^2 \cdot B^T$ , onde  $B$  é uma base ortonormal de autovetores, e os elementos da diagonal da matriz  $D$  são raízes quadradas dos correspondentes dos autovalores positivos. Podemos escrever a distribuição  $\mathcal{N}(\mu, C)$

de várias formas diferentes. Logo abaixo temos algumas:

$$\begin{aligned}
 \mathcal{N}(\mu, C) &\sim \mu + \mathcal{N}(0, C) \\
 &\sim \mu + C^{\frac{1}{2}} \mathcal{N}(0, I) \\
 &\sim \mu + B \cdot D \cdot (B^T \mathcal{N}(0, I)) \\
 &\sim \mu + B \cdot D \cdot \mathcal{N}(0, I)
 \end{aligned}$$

onde  $\sim$  é igualdade em distribuição e  $C^{\frac{1}{2}} = B \cdot D \cdot B^T$ .

## B.1 Relação da hessiana com a matriz de covariância

Considere um vetor aleatório  $x \in \mathbb{R}^n$  dado por uma distribuição normal multivariada  $\mathcal{N}(\mu, C)$ . Dada a FDP de  $x$ ,  $p(x)$ , definamos a função quadrática  $f(x) = -\log(p(x))$ . Vamos verificar que a hessiana da função  $f$  em  $x$  é a inversa da matriz de covariâncias de  $x$ . Ora, é fácil ver que

$$f(x) = \frac{n}{2} + \frac{\log|C|}{2} + \frac{(x - \mu)^T C^{-1} (x - \mu)}{2} \quad (\text{B.7})$$

Daí, efetuando as derivadas parciais, obtemos que

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x) = (C^{-1})_{ij}. \quad (\text{B.8})$$

Como a hessiana  $H$  de  $f$  em  $x$  é uma matriz cujas entradas  $H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$ , temos que

$$H = C^{-1} \quad (\text{B.9})$$

Para outras informações a respeito a distribuição normal multivariada, ver [21, 40].

## APÊNDICE C – Método Numérico para EDE

Uma equação diferencial não tem necessariamente uma solução, e quando existe nem sempre é possível exibi-la. Neste último caso, temos que nos contentar apenas uma aproximação da solução. Existem vários métodos para se determinar tal aproximação. Os métodos de Euler, Taylor ou Runge-Kutta já fazem parte da literatura atual, ver por exemplo [13, 32, 11, 6]. Faremos alguns comentários sobre o método de Euler.

Este método, às vezes chamado de método das tangentes, constitui uma das técnicas mais simples para se aproximar soluções de equações diferenciais.

Suponha que queremos encontrar a solução do seguinte problema:

$$x'(t) = f(t, x), \quad x(t_0) = x_0 \quad (\text{C.1})$$

Seja  $\sigma > 0$  um incremento na variável  $t$  de modo que seja possível achar um ponto  $(t_1, x_1) = (t_0 + h, x_1)$  na tangente à curva solução desconhecida em  $(x_0, t_0)$ . Daí,

$$\frac{x_1 - x_0}{t_1 - t_0} = x'(t_0) \quad (\text{C.2})$$

Como,  $x'(t_0) = f(t_0, x_0)$  temos que

$$x_1 = x_0 + \sigma f(t_0, x_0) \quad (\text{C.3})$$

O ponto  $(t_1, x_1)$  na tangente é uma aproximação do ponto  $(t_1, x(t_1))$  na curva solução. Em outras palavras,  $x_1 \approx x(t_1)$ . Essa aproximação depende do valor de  $\sigma$  escolhido. Em geral, tomamos suficientemente pequeno.

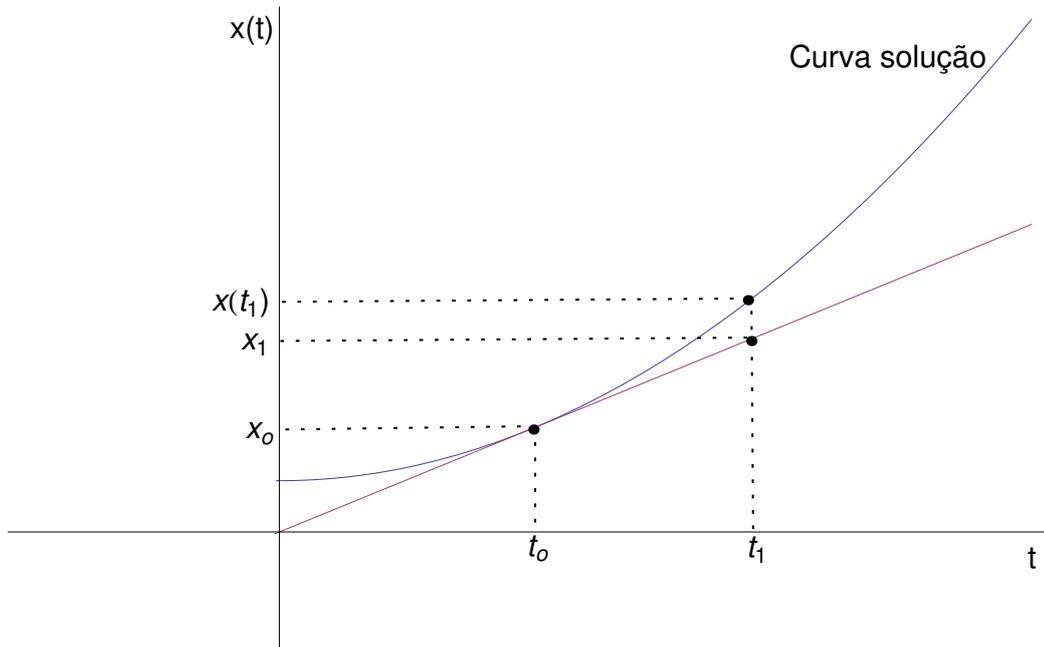


Figura C.1: Ilustração de uma iteração do método de Euler.

Indutivamente, definimos os demais termos, a saber,

$$x_{n+1} = x_n + \sigma f(t_n, x_n), \tag{C.4}$$

onde  $t_n = t_0 + n\sigma$ .

Agora, consideremos a seguinte equação diferencial estocástica

$$dX_t = a(t, X_t)dt + b(t, X_t)dB_t, \quad X_0 = x_0 \tag{C.5}$$

Vamos supor que as funções  $a(\cdot)$  e  $b(\cdot)$  possuem condições (cf. [4]) tais que exista solução para a equação C.5.

Para obter soluções numéricas para a equação C.5, primeiro consideramos o método de Euler para a equação diferencial  $x' = a(t, x)$  com  $x(0) = x_0$ , conforme visto acima. Depois, para parte " $dB_t$ ", nós podemos usar distribuições idênticas e independente (cf. C.5) para as realizamos  $B(t_n)$ . Assim, uma forma de encontrar soluções para C.5 é dada por

$$x_{n+1} = x_n + \sigma a(t_n, x_n) + b(t_n, x_n)\Delta_n^\sigma B, \tag{C.6}$$

onde  $\Delta_n^\sigma B := B(t_n) - B(t_{n-1})$  é a distribuição idêntica e independente  $\mathcal{N}(0, \sigma)$  para um

valor de  $\sigma > 0$  fixado.

## *Referências Bibliográficas*

- [1] D. Brockhoff E. Zitzler A. Auger, J. Bader. Theory of the hypervolume indicator: Optimal  $\mu$ -distributions and the choice of the reference point. *ACM*, pages 87–102, 2009.
- [2] Salem F. Adra and Peter J. Fleming. Diversity management in evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2011. to appear.
- [3] G. E. Andrews, R. , Askey, and R. Roy. *Special Functions (Encyclopedia of Mathematics and its Applications)*. Cambridge University Press, 2011.
- [4] L. Arnold. *Stochastic Differential Equations: Theory and Applications*. Dover Publications, 2012.
- [5] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.
- [6] S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Springer Science+Business, 2007.
- [7] J. Bader and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19:45–76, 2011.
- [8] N. Beume, C.M. Fonseca, M. Lopez-Ibanez, L. Paquete, and J. Vahrenhold. On the complexity of computing the hypervolume indicator. *IEEE Trans. Evol. Comp.*, 13(5):1075–1082, 2009.
- [9] P. A. N. Bosman and E. D. de Jong. Exploiting gradient information in numerical multi-objective evolutionary optimization. In *Proceedings of the 2005 Genetic and Evolutionary Computation Conference (GECCO'05)*, pages 755–762, Washington, DC, USA, June 2005. ACM.
- [10] D. Brockhoff and E. Zitzler. Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization,. *Lecture Notes in Computer Science 4193: Parallel Problem Solving from Nature - PPSN IX*., pages 533–542, September 2006.
- [11] C. A. Coello Coello, D.A. Van Veldhuizen, and G.B. Lamont. *Evolutionary Algorithms for Solving Multi-objective Problems*. Kluwer Academic Publishers, 2001.

- [12] K. Deb and K. Saxena. Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. *Proc. of 2006 IEEE Congress on Evolutionary Computation*, pages 3353–3360, July 16-21, 2006.
- [13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA II. *IEEE Trans. Evol. Comp.*, 6(2):182–197, 2002.
- [14] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. *Congress on Evolutionary Computation*, 1:825–830, 2002.
- [15] N. Hansen. The cma evolution strategy: A comparing review. towards a new evolutionary computation. 192 of *Studies in Fuzziness and Soft Computing*:75–102, 2006.
- [16] N. Hansen and S. Kern. Evaluating the cma evolution strategy on multimodal test functions. *Proceedings In Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII*, pages 282–291, 2004.
- [17] N. Hansen, S. D. Muller, and P. K. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 11(1):1–18, 2003.
- [18] C. Igel, H. Nikolaus, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evol. Comput.*, 15:1–28, March 2007.
- [19] C. Igel, T. Suttrop, and N. Hansen. Steady-state selection and efficient covariance matrix update in the multi-objective cma-es. pages 171–185, 2007.
- [20] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization. In *Genetic and Evolving Systems, 2008. GEFS 2008. 3rd International Workshop on*, pages 47–52, march 2008.
- [21] B. R. James. *Probalidade: Um curso em nível intermediário*. Coleção Projeto Euclides, 2010.
- [22] A. Jaszkiwicz. On the computational efficiency of multiple objective metaheuristics: The knapsack problem case study. *European Journal of Operational Research*, 158(2):418–433, 2004.
- [23] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer; Corrected edition, 1992.
- [24] H. Kuhn and A. Tucker. Nonlinear programming. In *Proceedings of the Berkeley Symposium on Mathematical and Statistical Probability*, pages 481–492, 1951.
- [25] S. Kukkonen and J. Lampinen. Ranking-dominance and many-objective optimization. *Proc. of 2007 IEEE Congress on Evolutionary Computation*, pages 3983–3990, September 25-28, 2007.
- [26] S. Li. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistic*, 4 (1):pp. 66–70, 2011.

- [27] P. Loskot and N. C. Beaulieu. On monotonicity of the hypersphere volume and area. *Journal of Geometry*, 87:pp. 96–98, 2007.
- [28] Vilfredo Pareto. *Manual of political economy*. Augustus M. Kelley, 1906.
- [29] Robin C. Purshouse and Peter J. Fleming. On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 11(6):770–784, 2007.
- [30] S. Schaffler, R. Schultz, and K. Weinzierl. Stochastic method for the solution of unconstrained vector optimization problems. *J. Optim. Theor. Appl.*, 114(1):209–222, 2002.
- [31] O. Schutze, A. Lara, and C.A.C. Coello. On the influence of the number of objectives on the hardness of a multiobjective optimization problem. *IEEE Transactions on Evolutionary Computation*, 2011. to appear.
- [32] P. K. Shukla and K. Deb. On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods. *European Journal of Operational Research*, 181:1630–1652, 2007.
- [33] V. L. S. Silva, E. F. Wanner, S. A. A. G. Cerqueira, and R. H. C. Takahashi. A new performance metric for multiobjective optimization: the integrated sphere counting. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 3625–3630, Singapore, 2007.
- [34] R. H. C. Takahashi, E. G. Carrano, and E. F. Wanner. On a stochastic differential equation approach for multiobjective optimization up to pareto-criticality. In *Proceedings of the 6th international conference on Evolutionary multi-criterion optimization, EMO'11*, pages 61–75, Berlin, Heidelberg, 2011. Springer-Verlag.
- [35] G. Timmel. Ein stochastisches suchverfahren zur bestim- mung der optimalen kompromisungen bei statischen pol- zkriteriellen optimierungsaufgaben. *Wissenschaftliche Zeitschrift der Technischen Hochschule Ilmenau*, 26 (5)(159-174), 1980.
- [36] D. A. G. Vieira, R. H. C. Takahashi, and R. R. Saldanha. Multicriteria optimization with a multiobjective golden section line search. *Mathematical Programming*, 2011. to appear (doi: 10.1007/s10107-010-0347-9).
- [37] T. VoB, N. Hansen, and C. Igel. Improved step size adaptation for the mo-cma-es. *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 2010.
- [38] E. F. Wanner. *Operators for genetic algorithms based on quadratic approximations of functions of continuous variables*. PhD thesis, 2006.
- [39] E. F. Wanner, F. G. Guimaraes, R. H. C. Takahashi, and P. J. Fleming. Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria. *Evolutionary Computation*, 16(2):185–224, 2008.
- [40] K. V. Yuen. *Bayesian methods for structural dynamics and civil engineering*. John Wiley & Sons (Asia) Pte Ltd, 2010.

- 
- [41] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
- [42] E. Zitzler and C.M. Fonseca V.G. Fonseca L. Thiele, M. Laumanns. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comp.*, 7(2):117–132, 2003.