

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

RONALDO ALVES DE ABREU

**Método de Otimização Não Convexa
por Análise Secante (MONCAS)**

Profa. Dra. Luziane Ferreira Mendonça
Orientador

Rio de Janeiro, Julho de 2009

*Dedico esse trabalho aos 60 anos de pesquisa em otimização e seus pesquisadores.
Seus esforços levaram a um novo ramo de pesquisa em Matemática e áreas afins.*

AGRADECIMENTOS

Gostaria de agradecer aos responsáveis pela minha introdução a esse grande campo de pesquisa: a Otimização.

Iniciando com Programação Linear e continuando por Métodos de Otimização Não-Linear (para funções suaves), passando por Otimização Inteira e Estocástica, fui introduzido a um extremamente rico campo de pesquisa com aplicação nos mais diversos campos da ciência, indústria e negócios.

Em uma época onde o uso racional de recursos é tão mencionada, seja por questões econômicas ou ambientais, a pesquisa de Métodos e Linhas de Otimização se tornam fundamentais para o melhor desempenho de qualquer empreendimento.

Em um mundo onde os recursos são escassos, somente os que os usarem da melhor forma possível podem prevalecer. Com um intuito menos competitivo do que foi exposto na frase anterior, a pesquisa nos vários campos da Otimização prossegue com cada vez mais importância no mundo da ciência.

Depois deste discurso romântico, gostaria de explicitamente agradecer à minha orientadora Profa. Dra. Luziane Ferreira de Mendonça, que me ensinou Programação Linear, Não Linear e Estocástica, e à Profa. Dra. Márcia Helena Fampa, que me aceitou como aluno ouvinte em um curso de Programação Inteira e Combinatória para a Graduação em Informática da UFRJ.

Agradeço ao PPGI e ao NCE. Seus recursos humanos e de infra-estrutura foram de grande ajuda, senão essenciais, para a conclusão deste trabalho. Gostaria de agradecer à CAPES, que forneceu uma bolsa de estudos para que pudesse cursar este Mestrado.

Gostaria de dar um agradecimento especial a meu amigo Hugo Cesar de Castro Carneiro que me ajudou em um ponto muito importante do desenvolvimento. Um outro agradecimento se faz necessário a Rogério Theodoro de Brito que me forneceu as traduções para o ambiente L^AT_EX **algorithm**.

Agradeço a meus pais e avós que me deram condições de chegar até aqui.

Agradeço ainda aos Profs. Drs. Fábio Protti, Ivo Lopes, Marcello Goulart Teixeira e Haroldo Clark que com sua paciência e interesse ensinaram esse aluno

vários aspectos que ligam as Ciências de Matemática e Informática.

RESUMO

Métodos de Otimização Sem Derivadas são utilizados em trabalhos científicos desde meados dos Século XX. Menos custosos em termos de programação das rotinas envolvidas nos cálculos, eles continuam sendo largamente utilizados nos mais diversos trabalhos científicos ou técnicos. Neste trabalho apresentamos uma breve explicação sobre dois métodos muito importantes, e os utilizamos como inspiração para formar uma nova abordagem para busca de um minimizador global para um problema irrestrito. Esta abordagem serve de fundamentação para a apresentação do método MONCAS (Método de Otimização Não-Convexa por Análise Secante) cujo desempenho é analisado por meio de simulações numéricas para minimização de problemas com estruturas não suaves e/ou com muitos minimizadores locais.

Palavras-chave: Otimização, Derivative-free, Busca Direta, Busca Padrão, Nelder Mead.

Secant Analysis on NonConvex Optimization Method (MONCAS)

RESUMO

Derivative-free Optimization Methods are used in scientific research since the middle of the XX Century. Less expensive in terms of programming time, they have been widely used in scientific and technical works. On this work, we present a brief explanation at two very important methods, and we use them as inspiration to create a new approach for searching the global minimizer for an unconstrained problem. This approach works as a fundament for the MONCAS method (portuguese anacronism for Secant Analysis on Non-Convex Optimization Method). We tested its performance by numerical simulations over functions either with non-smooth structures or which have several local minima.

Palavras-chave: Optimization, Derivative-free, Direct Search, Pattern Search, Nelder-Mead.

LISTA DE FIGURAS

Figura 2.1: Um hipercone do \mathbb{R}^2	23
Figura 2.2: Exemplo de bases minimal (esquerda) e maximal (direita).	24
Figura 2.3: Primeiro vértice do Simplex.	29
Figura 2.4: Segmento que forma uma versão degenerada de Simplex.	29
Figura 2.5: Simplex de um espaço com dimensão 2.	30
Figura 2.6: Simplex de um espaço tridimensional.	30
Figura 3.1: Centróide do Simplex tomando x_3 como pior vértice.	40
Figura 3.2: Reflexão do Simplex em relação ao vértice x_3	41
Figura 3.3: Expansão do Simplex em relação ao vértice x_3	42
Figura 3.4: Contração Externa do vértice x_3	43
Figura 3.5: Contração Interna do vértice x_3	43
Figura 3.6: Redução dos vértices x_1 , x_2 e x_3 em x_1' , x_2' e x_3' , respectivamente.	45
Figura 4.1: Busca Coordenada. Avaliações seguem nas direções dos eixos, transladados pelo ponto x (base).	47
Figura 4.2: Busca Padrão, as avaliações seguem em várias direções a partir do ponto x (base). À direita, Busca Padrão com outro Padrão (linhas em cinza representam uma base positiva minimal).	49
Figura 4.3: Padrão de Busca e simétrico do vetor gradiente (traço-e-ponto). Direção de maior projeção sobre o vetor simétrico é tomada para cálculo de novo iterando.	51
Figura 5.1: Exemplos de Vale. No primeiro, o mínimo é indicado pela reta cinza. No segundo, é apenas um ponto.	61
Figura 5.2: Ilustração de função não Convexa.	62
Figura 5.3: Secantes adjacentes em um gráfico de função	65
Figura 5.4: Esboço das secantes implicando em um critério $t < 1$	67
Figura 5.5: Esboço das secantes implicando em um critério $t = 1$	67
Figura 5.6: Esboço das secantes implicando em um critério $t > 1$	68

Figura 5.7: Esboço das secantes implicando em um critério $\tau < 1$	68
Figura 5.8: Esboço das secantes implicando em um critério $\tau = 1$	69
Figura 5.9: Esboço das secantes implicando em um critério $\tau > 1$	69
Figura 5.10: Caso $v_1 = 0$ com inclinação $v_2 < 0$	69
Figura 5.11: Caso $v_1 = 0$ com inclinação $v_2 = 0$	70
Figura 5.12: Caso $v_1 = 0$ com inclinação $v_2 > 0$	70
Figura 5.13: Ganho em Passo Interno com alerta para degeração no Passo Externo.	75
Figura 5.14: Fluxograma do Algoritmo MONCAS. Setas em cinza são negativos nos blocos de decisão	84
Figura 6.1: Esboço da função Griewank para $n = 2$	94
Figura 6.2: Esboço da função de Ackley para $n = 2$	95
Figura 6.3: Esboço da função Swefel Generalizada para $n = 2$	96
Figura 6.4: Esboço da função Hyper-Elipsóide Rotacionado para $n = 2$	96
Figura 6.5: Esboço da função Rastrigin para $n = 2$	97
Figura 6.6: Esboço da função de Schwefel para $n = 2$	98
Figura 6.7: Esboço da função Trigonométrica para $n = 2$	99
Figura 6.8: Perfil de Desempenho Baseado em Número de Iterações para dimensões n_1 (esquerda) e n_2 (direita).	113
Figura 6.9: Perfil de Desempenho Baseado em Número de Iterações para dimensões n_3 (esquerda) e n_4 (direita).	113
Figura 6.10: Perfil de Desempenho Baseado em Número de Iterações para dimensões n_5 (esquerda) e n_6 (direita).	114
Figura 6.11: Perfil de Desempenho Baseado em Tempo para dimensões n_1 (esquerda) e n_2 (direita).	114
Figura 6.12: Perfil de Desempenho Baseado em Tempo para dimensões n_3 (esquerda) e n_4 (direita).	115
Figura 6.13: Perfil de Desempenho Baseado em Tempo para dimensões n_5 (esquerda) e n_6 (direita).	115
Figura 6.14: Perfil de Desempenho Baseado em Número de Avaliações de Função para dimensões n_1 (esquerda) e n_2 (direita).	116
Figura 6.15: Perfil de Desempenho Baseado em Número de Avaliações de Função para dimensões n_3 (esquerda) e n_4 (direita).	116
Figura 6.16: Perfil de Desempenho Baseado em Número de Avaliações de Função para dimensões n_5 (esquerda) e n_6 (direita).	117
Figura 6.17: Gráfico de média de número de avaliações de função por dimensão do espaço considerando casos de salvaguarda.	121
Figura 6.18: Gráfico de média de número de avaliações de função por dimensão do espaço desconsiderando casos de salvaguarda.	122

LISTA DE TABELAS

Tabela 2.1: Tabela de resumo das coordenadas para geração de politopos e bases minimais.	38
Tabela 6.1: Solvers Aplicados na Solução dos Problemas	91
Tabela 6.2: Lista de Problemas para Avaliação de Métodos	92
Tabela 6.3: Variação das dimensões dos problemas usados na análise	100
Tabela 6.4: Dados sobre o limite de iterações e critério de parada	103
Tabela 6.5: Discriminação das Colunas dos resultados das simulações numéricas.104	
Tabela 6.6: Resultados das simulações numéricas.	105
Tabela 6.7: Resultados das simulações numéricas.	106
Tabela 6.8: Resultados das simulações numéricas.	107
Tabela 6.9: Resultados das simulações numéricas.	108
Tabela 6.10: Resultados das simulações numéricas.	109
Tabela 6.11: Resultados das simulações numéricas.	110
Tabela 6.12: Resultados das simulações numéricas.	111
Tabela 6.13: Número de términos bem sucedido por Passo Externo e dimensão. 120	
Tabela 6.14: Média de número de avaliações de função por Passo Externo e dimensão.	120
Tabela 6.15: Média de número de avaliações de função por Passo Externo e dimensão.	120

LISTA DE ALGORITMOS

1	Algoritmo para geração de politopo minimal	30
2	Algoritmo Nelder-Mead	44
3	Algoritmo de busca coordenada	48
4	Algoritmo de busca padrão	52
5	Algoritmo de análise de possibilidade de novo vale	72
6	Algoritmo de cálculo de nível de certeza de vale médio	79
7	Algoritmo de avaliação de parada de análise de vale	79
8	Algoritmo de movimentos exploratórios analítico	81
9	Algoritmo de análise de mudança de curvatura	82
10	Algoritmo de decisão de próximo iterando	83
11	Algoritmo MONCAS	85

LISTA DE ABREVIATURAS E SIGLAS

\mathbb{R}	Conjunto dos números reais ou corpo dos reais.
\mathbb{R}^n	Espaço vetorial de dimensão finita sobre o corpo dos reais.
E	Espaço Euclidiano, ou seja, um espaço do tipo \mathbb{R}^n .
QED	Quod erat demonstrandum, traduzindo do latim, como queríamos demonstrar.
LI	Linearmente independente.
LD	Linearmente dependente.
$\mathcal{B}_\rho(x)$	Vizinhança do ponto x com raio ρ é o conjunto dos pontos $y \in E$ tais que $\ x - y\ < \rho$.
$A \geq 0$	Matriz semipositiva definida é uma matriz $n \times n$ tal que $x^T A x \geq 0, \forall x \in E$.

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Sobre a utilidade e finalidade dos capítulos	17
1.2	Definições matemáticas relevantes	19
2	DEPENDÊNCIA LINEAR POSITIVA	21
2.1	A álgebra	22
2.2	Proposições e teoremas	24
2.3	algoritmo	28
2.3.1	Algoritmo ótimo para a construção de um simplex não degenerado	28
2.3.2	Aplicação do algoritmo	35
3	O MÉTODO NELDER-MEAD	39
3.1	Operações com vetores	40
3.2	Algoritmo	43
3.3	Convergência	45
4	BUSCA COORDENADA E BUSCA PADRÃO	46
4.1	Busca coordenada	47
4.1.1	Algoritmo	48
4.1.2	Critério de parada	48
4.2	Busca padrão	48
4.2.1	Critério de parada	51
4.3	Convergência de métodos de busca padrão	51
4.3.1	Convergência global	52
4.3.2	Convergência local	54
5	MÉTODO DE OTIMIZAÇÃO NÃO CONVEXA POR ANÁLISE SECANTE	59
5.1	Formulação dos princípios	60

5.2	Passo externo e secante no vale	62
5.3	Estrutura do capítulo	63
5.4	Análise secante	64
5.5	Análise de casos com secantes e fundamentos de decisão	70
5.5.1	Casos de possíveis vales	71
5.5.2	Decaimento melhorado	74
5.5.3	Degeneração	76
5.5.4	Critério de parada por análise de certeza de vale	77
5.5.5	Algoritmo de movimentos exploratórios analítico	80
5.6	Critério de decisão para novo iterando	81
5.7	Método MONCAS	84
5.8	Passo Externo	86
6	SIMULAÇÕES, METODOLOGIA DE COMPARAÇÃO E ANÁLISE	88
6.1	Métodos numéricos e parâmetros iniciais	89
6.1.1	Padrão de busca	91
6.2	Problemas	92
6.2.1	Parabolóide	93
6.2.2	Extended Powell singular function	93
6.2.3	Griewank function	93
6.2.4	Ackley's function	94
6.2.5	Generalized Swefel function	94
6.2.6	Rotated Hyper-Ellipsoid function	95
6.2.7	Rastrigin function	97
6.2.8	Schwefel's function	97
6.2.9	Trigonometric function	98
6.3	Dimensionamento dos problemas	99
6.4	Perfil de desempenho	100
6.5	Referencial técnico e simulações numéricas	102
6.6	Desenvolvimento e análise sobre perfis de desempenho	104
6.6.1	Comentários técnicos	112
6.6.2	Gráficos de perfis	112
6.6.3	Alta variação do perfil gerado por medição de tempo	117
6.7	Análise dos resultados	118
6.8	Passo Externo e dimensão do espaço E	119
7	CONCLUSÕES	124
	REFERÊNCIAS	127

1 INTRODUÇÃO

Há 50 anos, Métodos de Otimização Sem Derivada vêm sendo usados para encontrar elementos que minimizem, ou maximizem, certas funções de interesse nos mais variados campos da ciência.

Logo nos primeiros anos da Era dos Computadores Eletrônicos, métodos que aproximam o valor de mínimo, ou de máximo, de funções já estavam sendo criados para uso em Computadores Eletrônicos, mas apenas como uma necessidade dentro de pesquisas, sem o propósito de desenvolvê-los.

Devido às limitações dos computadores e tecnologias de computação à época, métodos que envolvessem derivadas eram muito custosos em termos de tempo. Seja devido ao número de avaliações necessárias, ou devido às linguagens e às técnicas de programação presentes à época para estes computadores, métodos que não fizessem uso de derivada, explicitamente, acabaram por se tornar populares. Tal classe de métodos acabou por receber o nome de Métodos de Otimização Sem Derivada (*Derivative-free Methods*, em inglês).

Nesta dissertação trataremos apenas de problemas de minimização irrestrita de uma

função de imagem real e domínio E . Matematicamente representamos isto por

$$\min f(x), \text{ onde } x \in E. \quad (1.1)$$

Apesar de ainda haver alguma discussão sobre o encapsulamento completo desta classe de métodos, isto é, ainda há argumentos a serem consolidados para que a inclusão de um método na categoria de *Método Sem Derivada* esteja fora de questionamento, um ponto fica explícito: “O Método não precisa ter uma forma analítica para o gradiente”.

Para funções altamente regulares, existem métodos de complexidade linear, a citar, o Método dos Gradientes Conjugados. Estes métodos têm suas próprias peculiaridades e outras questões a serem tratadas. Seguindo o exemplo do Método dos Gradientes Conjugados, ele necessita que um conjunto de direções conjugadas seja calculado ao longo do processo.

Métodos Sem Derivada podem fazer uso de outras estratégias para contornar a falta do gradiente (explicitamente calculado). Entre algumas muito usadas estão a Interpolação por uma função quadrática e o uso de Mínimos Quadrados para aproximar a solução a partir da amostragem dos pontos da função.

Uma subclasse dos Métodos de Otimização sem Derivada muito utilizada é chamada de Métodos de Busca Direta ou Métodos de Busca. A definição formal para os atributos de pertinência nesta classe também sofre de uma pequena controvérsia por parte de alguns autores. Seguindo o critério proposto em (LEWIS; TORCZON; TROSSET, 2000), consideramos que em Métodos de Busca Direta, além de não haver um gradiente avaliado, um gradiente aproximado por diferenças finitas ou uma função de estimação aproximada do mesmo, não há modelo a ser considerado sobre a função.

Com isso, se excluem modelos que fazem qualquer tipo de aproximação por Interpolação ou Mínimos Quadrados da Classe dos Métodos de Busca.

Com o objetivo de apresentar métodos que sejam cunhados nas características dos métodos de busca direta, produzimos este trabalho. Nele foram apresentados dois Métodos de Busca Direta com características diferentes, e uma modificação de minha própria autoria. O objetivo é formular a modificação, testá-la com problemas clássicos e executar comparação com os métodos bem estabelecidos.

Para melhor entendimento da disposição dos capítulos e suas utilidades, na seção a seguir apresentamos uma breve explicação sobre a disposição dos capítulos. Ao final inserimos algumas definições que são utilizadas.

1.1 Sobre a utilidade e finalidade dos capítulos

Os capítulos foram dispostos com a finalidade de que o leitor pudesse ter uma visão geral da finalidade do trabalho, sendo que os últimos capítulos fazem um papel chave para entender completamente a dissertação.

O Capítulo 2 tem uma importância mais matemática no contexto dos métodos. Nele, várias noções sobre Álgebra Linear, aplicadas no contexto de otimização, são expostas. Alguns teoremas fundamentais são demonstrados de forma original e um algoritmo ótimo para geração da base geradora mínima é enunciado e demonstrado.

No Capítulo 3, os conceitos básicos sobre o Método Nelder-Mead são expostos. A importância deste método é alta devida a sua vasta utilização, com sucesso, nos mais variados problemas. Ele será usado para base de comparação. O Método Nelder-Mead também serviu de inspiração para uma subrotina formulada pra o Método

MONCAS.

No Capítulo 4, é apresentado o Método de Busca Padrão, que tem um papel fundamental para o método introduzido nesta dissertação. São apresentados o Método de Busca Coordenada e o Método de Busca Padrão, sendo o segundo uma generalização do primeiro. Há provas de convergência para o Método de Busca Padrão, as quais também são apresentadas neste capítulo. O Método de Busca Padrão serve como segunda comparação para o método a ser desenvolvido.

No Capítulo 5, o método MONCAS, o principal da dissertação, é apresentado. Nele a idéia que originou o método é introduzida e desenvolvida. São também desenvolvidas e enunciadas as subrotinas de que o Método MONCAS faz uso, podendo ser consideradas seus “módulos”.

No Capítulo 6, são formulados os princípios da metodologia de simulação entre os Métodos de Busca Padrão, Nelder-Mead e MONCAS, como também são enunciados os problemas-teste e mostrados os resultados destas simulações. A análise do desempenho dos métodos, apresentada na parte final deste capítulo, é feita com o uso de uma ferramenta chamada Perfil de Desempenho (DOLAN; MORÉ, 2002).

No Capítulo 7, revisamos as principais características dos métodos apresentados no presente trabalho e das experiências numéricas desenvolvidas no Capítulo 6, culminando com a nossa conclusão sobre os resultados obtidos.

Nesta dissertação, serão denotadas por código as instruções programadas em linguagem de computador. Denominaremos de **pseudo-código** um algoritmo escrito em linguagem humana, ou informal.

1.2 Definições matemáticas relevantes

Apresentamos aqui algumas definições que apesar de básicas são muito relevantes para o contexto da dissertação.

Definição 1.2.1 *Em topologia dos Espaços Reais (\mathbb{R}^n), um **politopo** é uma generalização, para um espaço de dimensão n , do conceito de poliedro e polígono.*

Definição 1.2.2 *Levando em conta a topologia dos Espaços Reais, um **simplex** é uma generalização do conceito de triângulo e tetraedro, isto é, um simplex do \mathbb{R}^n é um politopo de $n + 1$ vértices.*

Em Otimização Não-Linear, quando há referência a um simplex, ela significa o termo topológico e não o método de Otimização Linear, exceto quando for explicitado o contrário. Quando um método faz uso de um simplex, ele é chamado de *simplético*.

Definição 1.2.3 *Para a topologia dos Espaços Reais, um **n -intervalo fechado** é um conjunto dado por*

$$[a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n] = \{x \in \mathbb{R} \mid a_i \leq x_i \leq b_i, i = 1, 2, \dots, n\}.$$

Definição 1.2.4 *Para a topologia dos Espaços Reais, um **n -intervalo aberto** é um conjunto dado por*

$$(a_1, b_1) \times (a_2, b_2) \times \dots \times (a_n, b_n) = \{x \in \mathbb{R} \mid a_i < x_i < b_i, i = 1, 2, \dots, n\}.$$

Qualquer *n -intervalo* que não é definido acima, não recebe nomenclatura especial. Apesar de alguém poder dizer que ele se trata de um **n -intervalo misto**.

Definição 1.2.5 Dizemos que um n -intervalo é **não-degenerado** quando $a_i < b_i$, $i = 1, \dots, n$, caso contrário é dito **degenerado**.

Definição 1.2.6 Dizemos que um **politopo** é **não-degenerado** se há um n -intervalo não-degenerado totalmente contido dentro do politopo. Desta forma, um simplex é um **politopo não-degenerado mínimo**.

Definição 1.2.7 Uma **sequência real**, é uma função $x : \mathbb{N} \rightarrow \mathbb{R}$ e representamos por $\{x_i\}_{i=0}^n$ sua imagem.

Definição 1.2.8 Um **minimizador global** da função $f(x)$ é um elemento $y \in E$ tal que $f(y) = \min \{f(x) \mid x \in E\}$.

Definição 1.2.9 Um **minimizador local** da função $f(x)$ é um elemento $y \in A \subset E$ tal que $f(y) = \min \{f(x) \mid x \in A\}$, para algum subconjunto $A \subset E$.

Definição 1.2.10 Define-se um **produto interno** do \mathbb{R}^n como sendo a forma bilinear

$$(a; b) = a_1 b_1 + \dots + a_i b_i + \dots + a_n b_n, \quad a, b \in \mathbb{R}^n.$$

Definição 1.2.11 Define-se um **ponto fixo**, ou ainda um **ponto estacionário**, como sendo o ponto $x \in E$ tal que, se $f(x) \in C^1(E)$, $\nabla f(x) = 0$.

2 DEPENDÊNCIA LINEAR POSITIVA

Desde o começo da segunda metade do século XX, uma aplicação da Teoria de Álgebra Linear vem despertando um interesse particular aos estudiosos de Matemática Aplicada. A Teoria da Dependência Linear Positiva vem sendo usada em vários campos de pesquisa da Matemática, como em Programação Linear, criada por George B. Dantzig, em Programação Não-Linear e na Teoria dos Jogos, criada por John von Neumann.

A Teoria da Dependência Linear Positiva tem uma interpretação geométrica muito forte, constantemente usada por autores e pesquisadores para ilustrar os princípios e resultados. Um de seus elementos mais importantes recebe o nome de um elemento da Geometria, um cone (ou hipercone, conforme se admita mais dimensões).

Em 1954, Chandler Davis publicou um trabalho (DAVIS, 1954) onde desenvolveu bem as características dos espaços vetoriais envolvidos nesta teoria, como também demonstrou vários teoremas e proposições, muito usados em otimização e outros campos da ciência.

Este capítulo se dedica à exposição de alguns de seus aspectos, mais especificamente

os elementos úteis para as demonstrações de alguns dos métodos abordados. A referência mencionada apresenta mais detalhes sobre o assunto.

Em especial, utilizamos em Métodos da Família de Busca Padrão, um conjunto de **Bases Positivas** para a formulação do **Padrão de Busca**.

2.1 A álgebra

Definiremos nesta seção a nomenclatura específica e todas as relações tomadas como hipóteses para os teoremas.

Consideraremos que todos os espaços vetoriais são Euclidianos, sobre o corpo dos números reais \mathbb{R} . Tomaremos como padrão que a dimensão do espaço Euclidiano é dada pela letra n . Para deixar explícito que se trata de um espaço euclidiano e para que possíveis concessões possam ser feitas sobre o corpo do Espaço Vetorial, denominaremos o espaço por E .

Definição 2.1.1 *Uma combinação linear positiva, ou simplesmente uma combinação positiva, é uma combinação linear tal que*

$$\lambda_1 a_1 + \lambda_2 a_2 + \dots + \lambda_r a_r \in E, \quad (2.1)$$

onde $\lambda_i \geq 0$, $i = 1, \dots, r$ e $a_i \in E$, $i = 1, \dots, r$

Igualmente como na Álgebra Linear, onde a combinação linear toma o papel de maior importância, aqui a combinação positiva toma a maior importância na teoria. Definiremos agora os equivalentes a subespaços vetoriais.

Definição 2.1.2 *Um cone, ou um hipercone, gerado pelos vetores $\{a_1, a_2, \dots, a_r\}$ é o conjunto $\{x \in E \mid x = \lambda_1 a_1 + \lambda_2 a_2 + \dots + \lambda_r a_r, \lambda_i \geq 0, i = 1, \dots, r\}$.*

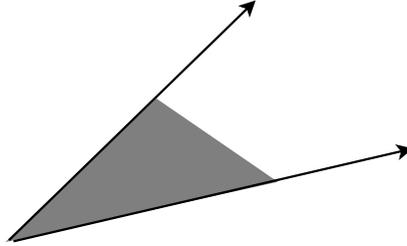


Figura 2.1: Um hipercone do \mathbb{R}^2 .

Chamamos de gerador, o conjunto $\{a_1, a_2, \dots, a_r\}$ e denotamos o cone gerado por eles como $span_+\{a_1, a_2, \dots, a_r\}$. Se é possível obter um subconjunto, digamos $\{a_{i_1}, \dots, a_{i_s}\} \subset \{a_1, a_2, \dots, a_r\}$ tal que

$$span_+\{a_{i_1}, \dots, a_{i_s}\} = span_+\{a_1, a_2, \dots, a_r\}$$

mas nenhum subconjunto próprio de $\{a_{i_1}, \dots, a_{i_s}\}$ satisfaz esta condição, diz-se que $\{a_{i_1}, \dots, a_{i_s}\}$ é uma base positiva de $span_+\{a_1, a_2, \dots, a_r\}$. Para o contexto desta dissertação, usaremos o termo *base* para nos referirmos a uma base positiva do hipercone. Caso haja necessidade para se fazer referência a uma base de um subespaço vetorial, usaremos o termo *base linear*.

Seja $B_a = \{a_{i_1}, \dots, a_{i_{s_a}}\}$ uma base de $C_a = span_+\{a_1, a_2, \dots, a_{r_a}\}$. Se, para um conjunto de vetores $\{b_1, b_2, \dots, b_{r_b}\} \subset C_a$ tal que $C_a = span_+\{b_1, b_2, \dots, b_{r_b}\}$ podemos extrair uma base $B_b = \{b_{j_1}, \dots, b_{j_{s_b}}\}$, com $s_b < s_a$, então dizemos que B_b é uma redução da base B_a . Em contraposição, dizemos que B_a é uma ampliação da base B_b . Se B é uma base da qual não se pode obter uma redução, ou seja, uma base irreduzível, dizemos que B é uma base minimal. Se B é uma base que não pode ser ampliada, dizemos que B é uma base maximal. Ilustramos isto na Figura 2.2.

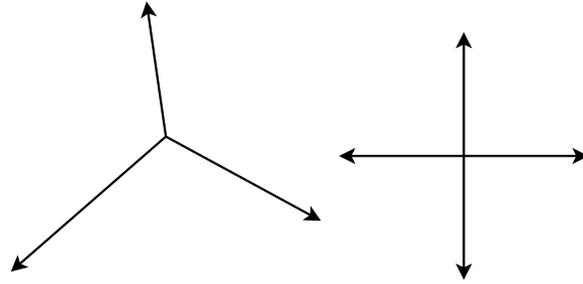


Figura 2.2: Exemplo de bases minimal (esquerda) e maximal (direita).

As relações de dependência ou independência são igualmente definidas neste contexto.

Definição 2.1.3 Dizemos que um conjunto $\{a_1, a_2, \dots, a_r\}$ é **positivamente linearmente dependente**, ou simplesmente **positivamente dependente**, se $\exists i$ tal que a_i pode ser escrito como combinação positiva dos demais. Caso não exista tal i , o conjunto é dito **positivamente linearmente independente**, ou simplesmente **positivamente independente**.

Com isso, abrimos a possibilidade de um conjunto LD, com coeficientes negativos, possa ser positivamente independente.

No restante do texto, ao fazermos uso do termo dependente, ou independente, estaremos nos referindo ao caso de dependência ou independência linear positiva. Ao nos referirmos a dependência ou independência linear, o termo completo será usado.

2.2 Proposições e teoremas

Como mencionado anteriormente, a Proposição 2.2.1, bem como o Teorema 2.2.2 foram enunciados e demonstrados em (DAVIS, 1954). O Teorema 2.2.4 é também

demonstrado no texto, porém não explicitamente. O Corolário 2.2.5, apesar de bem conhecido pelos pesquisadores do campo foi demonstrado sem ajuda de outros textos.

Proposição 2.2.1 *O conjunto $B = \{a_1, a_2, \dots, a_r\}$ é positivamente independente se, e somente se, forma uma base positiva para $C = \text{span}_+ B$.*

Demonstração :

Suponha que B é positivamente independente. Daí, nenhum dos vetores a_j , $j = 1, \dots, r$ pode ser combinação positiva dos demais. Logo, hipercones gerados por subconjuntos próprios não podem gerar positivamente os vetores excluídos em seus respectivos hipercones gerados, isto é, qualquer subconjunto próprio de B não pode gerar o mesmo hipercone gerado por B . Logo, B é uma base positiva para C .

*Suponha que B não seja positivamente independente. Daí, existe um subconjunto próprio de B , digamos, $G(j) = B - \{a_j\} = \{a_{i_1}, \dots, a_{i_s}\}$, $s < r$, tal que $a_j = \lambda_{i_1} a_{i_1} + \dots + \lambda_{i_s} a_{i_s}$, $\lambda_{i_j} \geq 0, \forall j$ e, $a_j \in \text{span}_+ G(j)$. Com isso, obtivemos um subconjunto próprio de B , $G(j)$ tal que $\text{span}_+ B = \text{span}_+ G(j)$ e B não é uma base para C . Uma prova por contrapositiva. **QED**.*

Teorema 2.2.2 *$G = \{a_1, a_2, \dots, a_r\}$ gera o espaço E , positivamente, se, e somente se $\forall x \in E$, $x \neq 0$, existe pelo menos um i tal que $(x; a_i) = \lambda > 0$*

Demonstração :

Suponha que G gera positivamente o espaço E . Daí, cada vetor x do espaço, pode ser decomposto em uma combinação positiva dos elementos do conjunto gerador G . Como, $x = 0 \iff \lambda_i = 0, \forall i = 1, \dots, r$, temos que pelo menos um destes λ_i deve ser maior que zero.

Agora, suponha que $\forall x \in E$ existe pelo menos um vetor a_i do conjunto G tal que $(x; a_i) = \lambda_i \|a_i\| \geq 0$. Obviamente, se $x = 0$ então $(x; a_i) = 0, \forall i = 1, \dots, r$. Ora, mas a afirmação anterior implica que se $x \neq 0$ então para o dito x temos um conjunto $D(x) \subset G$ tal que $(x; d_j) = \delta_j \|d_j\|^2 > 0, \forall d \in D(x)$. Considere o conjunto $R(x) = G - D(x)$, com isso estabelecemos uma forma de escrever x como uma combinação positiva dos elementos de G , onde associamos a cada vetor de $R(x)$ o peso nulo. **QED**.

Corolário 2.2.3 É válida a seguinte relação $(x; a_i) \geq c \|x\| \|a_i\|$ onde $0 < c \leq 1, \forall x \in \mathbb{R}^n$ desde que $x \neq 0$ e $\text{span}_+\{a_1, a_2, \dots, a_r\}$.

Demonstração :

Da Álgebra Linear, temos válida a relação

$$-1 \leq \left(\frac{x}{\|x\|}; \frac{a_i}{\|a_i\|} \right) \leq 1$$

Mas como a_i faz parte de um conjunto de geração positiva de \mathbb{R}^n , temos $(x; a_i) > 0, \forall x \in \mathbb{R}^n, x \neq 0$. Utilizando o resultado do Teorema 2.2.2, temos que

$$\left(\frac{x}{\|x\|}; \frac{a_i}{\|a_i\|} \right) \geq c > 0$$

e portanto,

$$(x; a_i) \geq c \|x\| \|a_i\|.$$

Teorema 2.2.4 Seja $B = \{a_1, a_2, \dots, a_r\}$ uma base positiva de E . Então, $n + 1 \leq r \leq 2n$.

Demonstração :

Obviamente, $r < n$ implica que E não pode ser gerado linearmente, e com $r = n$

ele é, no máximo, gerado linearmente. Agora, tome uma base linear de E dada por $\{a_1, a_2, \dots, a_n\}$. Construa um vetor

$$a_{n+1} = \alpha_1 a_1 + \alpha_2 a_2 + \dots + \alpha_n a_n, \quad \alpha_i < 0, \quad i = 1, \dots, n$$

assim, para um $x \in E$ arbitrário, sempre podemos decompô-lo em uma soma não negativa de vetores do conjunto $\{a_1, a_2, \dots, a_{n+1}\}$.

Agora, observemos que se um conjunto de vetores, arbitrários, $\{b_1, b_2, \dots, b_{2n+1}\}$ gera E , então existem pelo menos n vetores LI. Estes vetores geram, positivamente, um hipercone, que divide em 2, o espaço E . De fato, o hipercone gerado por eles geram positivamente um semi-espaço, pois caso contrário, eles não seriam LI. Denominemos R o semi-espaço gerado pelos vetores restantes. Dos $n + 1$ vetores restantes, no máximo n podem ser LI e gerar o semi-espaço R , e, por tanto, ser uma base para ele. Daí, podemos concluir que $2n$ é um número máximo de vetores para uma base positiva de E . **QED.**

Notemos que nem toda base positiva de E é dado por um conjunto $\{a_1, a_2, \dots, a_n\}$ unido a seus simétricos $\{-a_1, -a_2, \dots, -a_n\}$. De fato, é suficiente que haja uma divisão dos $2n$ vetores tal que n permaneçam em um semi-espaço e os demais em outro.

Corolário 2.2.5 *Seja $\{a_1, a_2, \dots, a_{n+1}\}$ um conjunto de vetores. Se $(a_i; a_j) < 0, \forall i \neq j$ então $\{a_1, a_2, \dots, a_{n+1}\}$ é uma base de E .*

Demonstração :

É de imediato que se os elementos satisfazem essa condição então eles são positivamente independentes. Conforme o Teorema 2.2.2, visto anteriormente, isto confere a $\{a_1, a_2, \dots, a_{n+1}\}$ as características de uma base positiva. **QED.**

2.3 algoritmo

Esta seção apresenta um algoritmo particularmente útil para esta dissertação. Ele foi desenvolvido especialmente para duas finalidades:

- Montar um simplex regular e;
- Gerar uma base positiva minimal.

A seguir mostramos os fundamentos que nos levaram a desenvolvê-lo e seu desenvolvimento matemático. Igualmente, demonstramos que este algoritmo resolve o problema proposto.

2.3.1 Algoritmo ótimo para a construção de um simplex não degenerado

Iniciaremos o desenvolvimento de princípios que nos levarão a obter um politopo minimal regular no espaço \mathbb{R}^n , isto é, um simplex. Este procedimento pode ser usado para gerar um conjunto de direções positivas que geram uma base positiva minimal do espaço.

Para isso, teremos que fazer uso de um conjunto ortonormal \mathcal{A} arbitrário. Seja $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ tais que $(a_i; a_j) = \delta_{ij}$, o delta de Kronecker. Faremos, também, uso da representação em componentes para um vetor $v \in \mathbb{R}^n$, isto é,

$$v = v^1 a_1 + v^2 a_2 + \dots + v^n a_n = (v^1, v^2, \dots, v^n)_{\mathcal{A}} = (v^1, v^2, \dots, v^n).$$

Caso o vetor esteja indexado por um índice i , então representaremos por

$$v_i = v_i^1 a_1 + v_i^2 a_2 + \dots + v_i^n a_n = (v_i^1, v_i^2, \dots, v_i^n)_{\mathcal{A}} = (v_i^1, v_i^2, \dots, v_i^n).$$

$$\begin{array}{c} v_0 \\ * \end{array}$$

Figura 2.3: Primeiro vértice do Simplex.

$$v_0 * \text{-----} * v_1$$

Figura 2.4: Segmento que forma uma versão degenerada de Simplex.

O algoritmo, aqui proposto, visa encontrar um polítopo minimal de lado l sempre executando uma expansão ortogonal do espaço vetorial considerado, isto é, a cada iteração k estamos considerando vetores expressos por:

$$v = v^1 a_1 + v^2 a_2 + \dots + v^k a_k + 0 + \dots + 0 = (v^1, v^2, \dots, v^k, 0, \dots, 0) = (v^1, v^2, \dots, v^k).$$

Começamos considerando um simplex do espaço degenerado D de dimensão nula. Um simplex neste espaço é dado por um único ponto, que consideraremos a origem de um Espaço Euclidiano E . Nomearemos este ponto por v_0 .

O próximo passo, na iteração onde o contador $k = 1$, expande nosso simplex para um segmento de reta de lado l . Como o espaço D é representado pela origem, qualquer direção gera um espaço ortogonal a ele. Tomamos então o vetor a_1 . Como os vetores são ortonormais, o nosso simplex deverá ser composto dos vetores v_0 e v_1 , com $v_1 = l a_1$.

A partir deste ponto, iteração onde $k \geq 2$, tomaremos o ponto médio do simplex obtido na iteração anterior e sempre executaremos uma expansão ortogonal do espaço acrescentando o vetor a_k à base.

Formularemos o seguinte algoritmo:

Demonstraremos que tal algoritmo forma um polítopo regular do espaço \mathbb{R}^n com

Algoritmo 1 Algoritmo para geração de politopo minimal

Entrada: Obtenha uma base ortonormal $\{a_1, a_2, \dots, a_n\}$ e um lado l ;

Saída: Um simplex $\{v_0, v_1, \dots, v_n\}$;

1: Tome $v_0 = 0$;

2: **para** $k = 1, \dots, n$ **faça**

3: Calcule o baricentro do simplex atual e nomeie-o ponto $\bar{v} \in \mathbb{R}^{k-1}$;

4: Calcule o próximo vetor, tomando um ponto cujas coordenadas são $v_k = \bar{v}^1 a_1 + \bar{v}^2 a_2 + \dots + \bar{v}^{k-1} a_{k-1} + v^k a_k$, utilizando a seguinte fórmula para v^k :

$$v^k = \sqrt{l^2 - \sum_{j=1}^{k-1} ((\bar{v}^j - v_i^j)^2)}, i < k$$

5: **fim para**

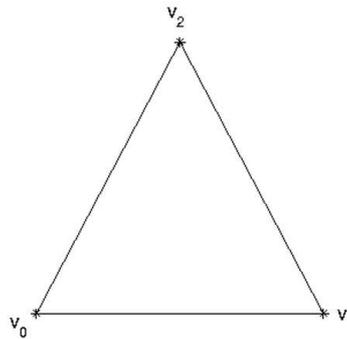


Figura 2.5: Simplex de um espaço com dimensão 2.

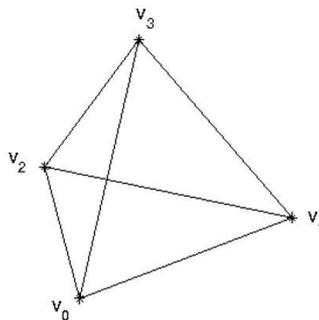


Figura 2.6: Simplex de um espaço tridimensional.

$n + 1$ vértices. Porém, antes disso definiremos algumas notações que utilizaremos para facilitar nosso raciocínio.

Definição 2.3.1 *Seja $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ um conjunto LI. Chamaremos de $S^{(k)}$ o conjunto dos primeiros k vetores de \mathcal{A} , isto é, $S^{(k)} = \{a_1, a_2, \dots, a_k\}$.*

Definição 2.3.2 *Seja $\{v_0, v_1, \dots, v_n\}$ um simplex regular do \mathbb{R}^n , definiremos por $Sx^{(k)}$ a lista dos primeiros $k+1$ vetores de $\{v_0, v_1, \dots, v_n\}$, isto é, $Sx^{(k)} = \{v_0, v_1, \dots, v_k\}$.*

O seguinte teorema será enunciado, mas não será demonstrado (uma demonstração pode ser obtida de (COXETER, 1973)).

Teorema 2.3.1 *O ponto médio de todo politopo regular dista a mesma distância de cada um dos seus vértices.*

Seguem os teoremas de demonstração sobre o algoritmo:

Teorema 2.3.2 *Sob a validade do Teorema 2.3.1, o algoritmo 1 gera um simplex regular.*

Demonstração :

Primeiramente, notemos que para o caso de $n = 2$ o algoritmo gera um simplex regular. De fato, na iteração onde $k = 1$ geramos um vértice $v_1 = l a_1 = (l, 0)$. O ponto médio do simplex $Sx^{(1)}$ é o ponto $\bar{v} = (0.5l, 0)$. Então geramos o próximo vértice $v_2 = (0.5l, v^2)$, onde $v^2 = \sqrt{l^2 - (0.5l - l)^2} = l\sqrt{\frac{3}{4}}$.

Neste caso, temos que v^2 foi obtido a partir da distância do ponto v_2 ao ponto v_1 . Para que esse valor seja correto deve ser igualmente possível obtê-lo pela distância entre v_2 e v_0 . Isso é verdade pois o triângulo $v_1 - \bar{v} - v_2$ é semelhante ao triângulo $v_0 - \bar{v} - v_2$, devido à ortonormalidade dos vetores em \mathcal{A} . Isso garante que o ponto v^2 dista a mesma distância l de ambos os outros vértices.

Passaremos à demonstração do passo de indução neste instante. Suponha que na iteração onde $k = n - 1$ tenhamos obtido um simplex regular. Sob a validade do Teorema 2.3.1, podemos obter um ponto \bar{v} que dista a mesma distância de cada um dos pontos do conjunto $Sx^{(n-1)}$. Assim, construímos um ponto $v_n = (\bar{v}^1, \bar{v}^2, \dots, \bar{v}^{n-1}, v^n)$, onde v^n é obtido por $v^n = \sqrt{l^2 - \sum_{j=1}^{n-1} ((\bar{v}^j - v_{n-1}^j)^2)}$.

Para esse passo vale o mesmo argumento sobre a semelhança de triângulos que foi aplicado no caso onde $n = 2$. De fato, sob a validade do Teorema 2.3.1, temos que os triângulos formados por $v_n - \bar{v} - v_i, i = 1, \dots, n - 1$, quando planarizados, são todos semelhantes devido à ortonormalidade dos pontos do conjunto \mathcal{A} . Assim, o ponto v^n dista a uma distância l de todos os outros vértices e esse politopo é regular, finalizando a indução. **QED.**

Suponha que na iteração k tenhamos obtido um vetor $\bar{v}_k = (\bar{v}_k^1, \dots, \bar{v}_k^{k-1})$, na iteração $k + 1$ obteremos um vetor $\bar{v}_{k+1} = (\bar{v}_{k+1}^1, \dots, \bar{v}_{k+1}^{k-1}, \bar{v}_{k+1}^k)$. As componentes $\bar{v}_{k+1}^1, \dots, \bar{v}_{k+1}^{k-1}$ apresentam uma certa regularidade, que iremos mostrar com um pequeno cálculo.

Cada componente $i = 1, \dots, k - 1$ tem a seguinte forma:

$$\bar{v}_{k+1}^i = \frac{\sum_{j=0}^k (\bar{v}_j^i)}{k+1} = \frac{\sum_{j=0}^{k-1} (\bar{v}_j^i) + \bar{v}_k^i}{k+1} = \frac{\binom{k}{i} \frac{\sum_{j=0}^{k-1} (\bar{v}_j^i)}{k} + \bar{v}_k^i}{k+1}, \quad \bar{v}_k^i = \frac{\sum_{j=0}^{k-1} (\bar{v}_j^i)}{k}$$

$$\bar{v}_{k+1}^i = \frac{\binom{k}{i} \bar{v}_k^i + \bar{v}_k^i}{k+1} = \bar{v}_k^i$$

Devido à expansão ortogonal sobre a qual o algoritmo foi formado, a componente k do vetor \bar{v}_{k+1} é imediatamente obtida pela divisão da componente k do vetor $v_k \in Sx^{(k)}$ por $k+1$, isto é,

$$\bar{v}_{k+1}^k = \frac{\overbrace{0 + \dots + 0}^{k-1} + v_k^k}{k+1} = \frac{v_k^k}{k+1} \quad (2.2)$$

Usaremos a notação $\bar{v}_{(p)}^i$ para indicar a componente i do vetor $\bar{v}_{(p)} \in S^{(p)}$. Assim, nosso vetor \bar{v}_{k+1} é composto por

$$\bar{v}_{k+1} = \left(\bar{v}_{(1)}^1, \bar{v}_{(2)}^2, \dots, \bar{v}_{(k)}^k \right)$$

onde cada componente $\bar{v}_{(i)}^i, i = 1, \dots, k$ foi calculada na iteração $i+1$.

Dada essa relação, temos que cada vértice v_i do simplex $Sx^{(k)}$ tem a seguinte forma

$$v_i = \left(\bar{v}_{(1)}^1, \bar{v}_{(2)}^2, \dots, \bar{v}_{(i-1)}^{i-1}, v_i^i, 0, \dots, 0 \right), 1 \leq i \leq k \leq n$$

onde a componente v_i^i é obtida por

$$v_i^i = \sqrt{l^2 - \left(v^b - \bar{v}_{(b)}^b \right)^2 - \sum_{j=b+1}^{i-1} \left(\bar{v}_{(j)}^j \right)^2}, b < i,$$

com $v_b \in Sx^{(i-1)}$.

Esta fórmula, aliada ao Teorema 2.3.2, nos leva a tomar v_b como sendo v_{i-1} sempre, a fim de evitar cálculos desnecessários. E, deste modo, temos

$$v_i^i = \sqrt{l^2 - \left(v^{i-1} - \bar{v}_{(i-1)}^{i-1} \right)^2} \quad (2.3)$$

Ao calcularmos tais valores, de v_i^i e $\bar{v}_{(i)}^i$, podemos encontrar certas fórmulas, enunciadas a seguir:

Teorema 2.3.3 *São válidas as seguintes fórmulas:*

$$v_i^i = v^i = l\sqrt{\frac{i+1}{2i}} \quad (2.4)$$

e

$$\bar{v}_{(i)}^i = \bar{v}^i = l\sqrt{\frac{1}{2i(i+1)}}. \quad (2.5)$$

Demonstração :

Vamos iniciar com a Equação (2.5). Da relação demonstrada em (2.2), temos que $\bar{v}_{(n)}^n = \frac{v_n^n}{n+1}$. A fórmula (2.5) é uma consequência direta da validade da fórmula (2.4).

Vamos nos focar na Equação (2.4). Notemos que para $n = 2$ já calculamos o valor de $v_2^2 = l\sqrt{\frac{3}{4}} = l\sqrt{\frac{2+1}{2(2)}}$. Vamos passar ao passo de indução e usar a fórmula (2.3)

para comprová-la. Suponhamos que $v_n^n = l\sqrt{\frac{n+1}{2n}}$ então temos que

$$\begin{aligned} v_{n+1}^{n+1} &= \sqrt{l^2 - (v_n^n - \bar{v}_{(n)}^n)^2} = \sqrt{l^2 - \left(\left(1 - \frac{1}{n+1}\right) v_n^n \right)^2} = \\ &= \sqrt{l^2 - l^2 \frac{n+1}{2n} \left(\frac{n}{n+1} \right)^2} = l\sqrt{1 - \frac{n}{2(n+1)}} = l\sqrt{\frac{2(n+1) - n}{2(n+1)}} = \\ &= l\sqrt{\frac{n+2}{2(n+1)}} \end{aligned}$$

o que conclui da indução e prova a fórmula (2.4).

Da relação (2.2), temos

$$\bar{v}_{(n)}^n = \frac{v_n^n}{n+1} = \frac{l}{n+1} \sqrt{\frac{n+1}{2n}} = l \sqrt{\frac{1}{2n(n+1)}}$$

o que prova a fórmula (2.5). **QED.**

Assim, a forma geral do simplex $\{v_0, v_1, \dots, v_n\}$ com cada vetor da forma $v_i = (x_i^1, x_i^2, \dots, x_i^n)$ é

$$x_i^j = \begin{cases} \bar{v}_{(j)}^j, & j < i \\ v_{(j)}^j, & j = i, \quad 0 \leq i \leq n, \quad 1 \leq j \leq n \\ 0, & j > i \end{cases}$$

2.3.2 Aplicação do algoritmo

Durante a exposição de aplicações, nos valem do seguinte lema,

Lema 2.3.4 *É válida a igualdade*

$$\sum_{j=a}^b \left(\frac{1}{j(j+1)} \right) = \frac{1}{a} - \frac{1}{b+1}.$$

Demonstração :

Decompondo o termo do somatório em frações parciais, temos:

$$\frac{1}{j(j+1)} = \frac{c}{j} + \frac{d}{j+1} = \frac{(c+d)j+c}{j(j+1)} \implies c=1, d=-1;$$

então,

$$\sum_{j=a}^b \left(\frac{1}{j(j+1)} \right) = \frac{1}{a} - \frac{1}{a+1} + \frac{1}{a+1} - \frac{1}{a+2} + \dots + \frac{1}{b} - \frac{1}{b+1} = \frac{1}{a} - \frac{1}{b+1}$$

QED.

2.3.2.1 Geração de uma base positiva minimal

Seja P_k um conjunto de $n + 1$ vetores do \mathbb{R}^n . Nossa intenção é que os vetores de P_k representem direções positivas do Espaço Euclidiano. Para isso, tomamos um simplex do \mathbb{R}^n , o seu centro e as subtrações dos vértices v_i para o centro.

Assim, dados vértices do tipo

$$v_i = (x_i^1, x_i^2, \dots, x_i^n)$$

com

$$x_i^j = \begin{cases} \bar{v}^j, & j < i \\ v^j, & j = i, \quad 0 \leq i \leq n, \quad 1 \leq j \leq n \\ 0, & j > i \end{cases}$$

e um centro do tipo

$$\bar{v} = (\bar{v}^1, \bar{v}^2, \dots, \bar{v}^n)$$

temos que as direções $d_i = v_i - \bar{v}$, $0 \leq i \leq n$ são dadas por

$$d_i = (y_i^1, y_i^2, \dots, y_i^n)$$

$$y_i^j = x_i^j - \bar{v} = \begin{cases} 0, & j < i \\ v^j - \bar{v}^j, & j = i \\ -\bar{v}^j, & j > i \end{cases}$$

Aplicando as fórmulas (2.4) e (2.5), temos:

$$y_i^j = \begin{cases} 0, & j < i \\ l \sqrt{\frac{j}{2(j+1)}}, & j = i, \quad 0 \leq i \leq n, \quad 1 \leq j \leq n \\ -\frac{l}{\sqrt{2j(j+1)}}, & j > i \end{cases} \quad (2.6)$$

Vamos demonstrar que as direções d_i satisfazem o critério de Base Minimal, isto é, $(d_i; d_k) < 0, i \neq k$.

Vamos supor que $i < k$, sem perda de generalidade. Procedendo diretamente para o cálculo, temos

$$(d_i; d_k) = \sum_{j=1}^n (y_i^j y_k^j)$$

Mas, $j < k \implies y_k^j = 0$ e assim, utilizando o Lema 2.3.4, temos:

$$\begin{aligned} (d_i; d_k) &= \sum_{j=k}^n (y_i^j y_k^j) = \frac{-l}{\sqrt{2k(k+1)}} l \sqrt{\frac{k}{2(k+1)}} + \sum_{j=k+1}^n \left(\left(\frac{-l}{\sqrt{2j(j+1)}} \right)^2 \right) = \\ &= \frac{-l^2}{2(k+1)} + \sum_{j=k+1}^n \left(\frac{l^2}{2j(j+1)} \right) = \frac{l^2}{2} \left(\frac{-1}{k+1} + \frac{1}{k+1} - \frac{1}{n+1} \right) = -\frac{l^2}{2(n+1)} < 0, \end{aligned}$$

e com isso, temos que as direções obtidas por 2.6 estão de acordo com o colorário 2.2.5.

2.3.2.2 Base minimal com vetores de módulo unitário

Consideraremos uma Base Minimal Positiva formada por vetores $u_i = (z_i^1, z_i^2, \dots, z_i^n)$ tais que $\|u_i\| = 1 \iff \|u_i\|^2 = 1$, o que nos dá

$$\begin{aligned} \|u_i\|^2 &= (u_i; u_i) = \sum_{j=1}^n (z_i^j z_i^j) = (z_i^i)^2 + \sum_{j=i+1}^n \left((z_i^j)^2 \right) = \\ &= \frac{l^2 i}{2(i+1)} + \sum_{j=i+1}^n \left(\frac{l^2}{2j(j+1)} \right) = \frac{l^2}{2} \left(\frac{i}{i+1} + \frac{1}{i+1} - \frac{1}{n+1} \right) = \\ &= \frac{l^2}{2} \left(1 - \frac{1}{n+1} \right) = \frac{nl^2}{2(n+1)} = 1 \implies l = \sqrt{\frac{2(n+1)}{n}}. \end{aligned}$$

	x_i^j	y_i^j	z_i^j
$j < i$	$\frac{l}{\sqrt{2j(j+1)}}$	0	0
$j = i$	$l\sqrt{\frac{j+1}{2j}}$	$\frac{lj}{\sqrt{2j(j+1)}}$	$\sqrt{\frac{j(n+1)}{n(j+1)}}$
$j > i$	0	$\frac{-l}{\sqrt{2j(j+1)}}$	$-\sqrt{\frac{n+1}{nj(j+1)}}$

Tabela 2.1: Tabela de resumo das coordenadas para geração de politopos e bases minimais.

Com isso a Equação (2.6) é alterada para

$$z_i^j = \begin{cases} 0, & j < i \\ \sqrt{\frac{j(n+1)}{n(j+1)}}, & j = i, \\ -\sqrt{\frac{n+1}{nj(j+1)}}, & j > i \end{cases}, \quad 0 \leq i \leq n, \quad 1 \leq j \leq n. \quad (2.7)$$

2.3.2.3 Resumo

Para facilitar a consulta vamos organizar as coordenadas obtidas por cálculos na tabela 2.1. Onde consideramos $0 \leq i \leq n$ e $1 \leq j \leq n$.

3 O MÉTODO NELDER-MEAD

Em 1965, no trabalho (NELDER; MEAD, 1965), dois estatísticos ingleses, chamados John Ashworth Nelder e Roger Mead, desenvolveram um método para resolver o Problema (1.1).

O método consiste em delimitar uma região com um politopo de $n + 1$ vértices, o que é chamado de um simplex, e, a partir de avaliações nos seus vértices, modificar suas posições no espaço \mathbb{R}^n seguindo alguns critérios. Este procedimento se repete até que o simplex tenha convergido ao interior de uma região pequena o suficiente. Esta heurística tem sido amplamente usada com bastante sucesso em vários casos.

O seu uso tão amplo pode ser explicado pela simplicidade de sua abordagem. Ela contém quatro grandes operações principais, todas elas baseadas em operações lineares simples, como soma de um vetor ponderado por um escalar e avaliações de centróides para os politopos.

Além destas operações, são tomadas algumas outras auxiliares, tais como a ordenação dos índices (crescente de acordo com a imagem dos pontos a que estão associados), e a avaliação e a comparação dos valores da função.

3.1 Operações com vetores

Antes das operações com os vértices, há algumas considerações de ordem de programação que merecem ser mencionadas. Durante o processo trabalhamos com os índices segundo os quais os vértices do simplex estão numerados, assim é necessário manter uma lista contendo os índices de tais vértices. Para finalidades do método, esses índices devem estar ordenados crescentemente segundo o valor da função objetivo no vértice correspondente. Então, tomamos o primeiro e o último índice desta lista e os nomeamos l e h , respectivamente.

A primeira operação a se considerar neste método é a de se obter o centróide da região determinada pelos n vértices com as menores imagens (como o simplex de pontos do \mathbb{R}^n tem $n + 1$ pontos, esse *centróide compensado* não considera o ponto de maior imagem). Como o politopo é uma figura geométrica convexa, a retirada de qualquer um dos vértices mantém a região restante uma região convexa, logo o seu centróide é dado pela média dos vértices x_l, \dots, x_h . É comum designar essa média por \bar{x} , como indicado na Figura 3.1. Então

$$\bar{x} = \sum_{i=1}^{n+1} \frac{x_i}{n} - \frac{x_h}{n}. \quad (3.1)$$

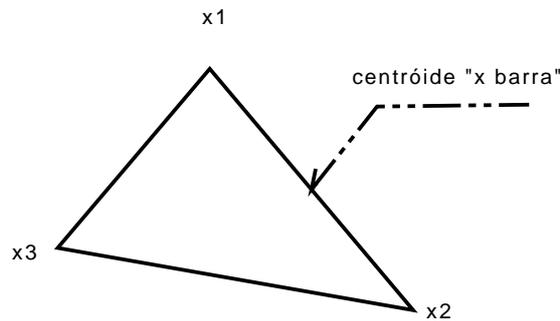


Figura 3.1: Centróide do Simplex tomando x_3 como pior vértice.

A segunda operação, denominada reflexão (podendo ser considerada, ainda, uma reflexão ponderada), pode ser considerada como a principal operação de cada iteração do método, pois ela é a responsável por perturbar a posição do “pior vértice do simplex”. Ela consiste em transladar o vértice de maior valor da função objetivo, geralmente mencionado como o pior vértice, em contra posição ao seu centróide, isto é, soma-se o vetor do vértice com o vetor de direção vértice-centróide, como indicado na Figura 3.2. Em termos matemáticos queremos dizer que se

$$d = \bar{x} - x_h \quad (3.2)$$

é a direção vértice-centróide, então a operação de reflexão é dada por

$$x_r = \bar{x} + \alpha (\bar{x} - x_h), \quad (3.3)$$

onde α é um parâmetro positivo que controla a distância de deslocamento que o vértice irá sofrer para que se possa obter um melhor simplex, ou seja um conjunto de vértices onde os valores da função objetivo tenha um valor menor nos vértices.

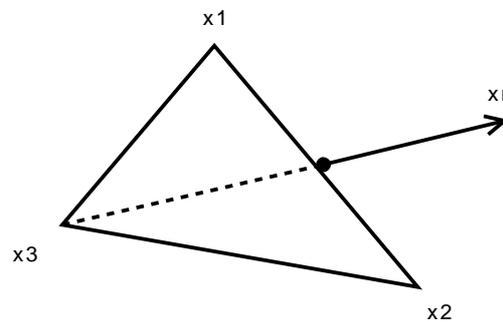


Figura 3.2: Reflexão do Simplex em relação ao vértice x_3 .

Outra operação é a de expansão por um coeficiente $\gamma > 1$. O resultado é representado por x_e , e é obtido de forma semelhante a fórmula de x_r , apenas substituindo x_h e α por x_r e $-\gamma$, respectivamente (indicado pela Figura 3.3). Matematicamente, temos

$$x_e = \bar{x} + \gamma (x_r - \bar{x}). \quad (3.4)$$

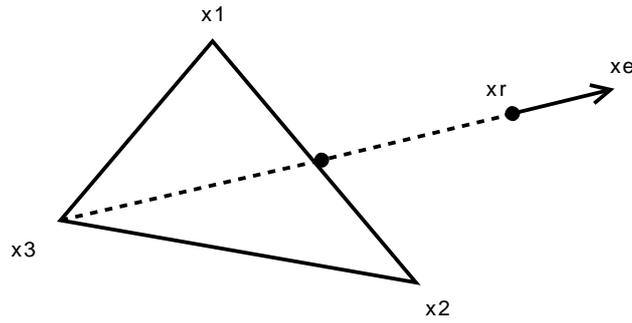


Figura 3.3: Expansão do Simplex em relação ao vértice x_3 .

Uma outra operação é a de contração por um parâmetro $0 < \beta < 1$. Aqui, denominamos o ponto contraído por x_c e utilizamos a fórmula da reflexão, onde substituímos α e x_h por β e $\min\{x_h, x_r\}$, respectivamente. Nomeamos

$$\tilde{x} = \arg \min_{x \in \{x_h, x_r\}} f(x)$$

e daí,

$$x_c = \bar{x} + \beta(\tilde{x} - \bar{x}). \quad (3.5)$$

Seguindo a mesma nomenclatura que a adotada em (CIAPPINA, 2008), se $\tilde{x} = x_r$ então a contração é dita Contração Externa, caso contrário é dita Contração Interna, indicadas pelas Figuras 3.4 e 3.5, respectivamente.

A última operação é a de redução, onde todos os vértices são substituídos pelos pontos médios dos segmentos de reta que os ligam a x_l (indicado na Figura 3.6). Matematicamente, temos

$$x_i \leftarrow \frac{(x_i + x_l)}{2}, \forall i \in 1, \dots, n+1. \quad (3.6)$$

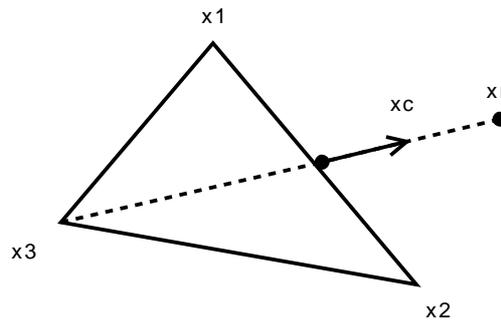


Figura 3.4: Contração Externa do vértice x_3 .

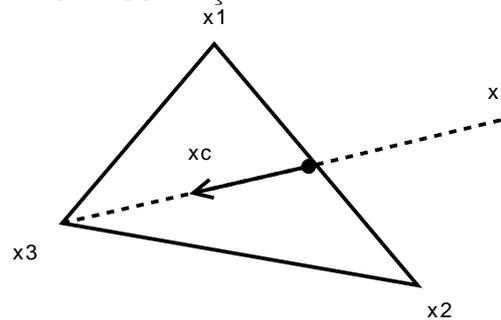


Figura 3.5: Contração Interna do vértice x_3 .

3.2 Algoritmo

O método de Nelder-Mead é uma heurística, isto é, não existe uma demonstração que garanta a solução de um problema de minimização a partir de sua aplicação. Apresentamos o pseudo-código do algoritmo. O algoritmo faz uso de uma função $g(x)$ como critério de parada.

O trabalho (NELDER; MEAD, 1965), e também (CIAPPINA, 2008), utilizam como medida de “erro para o mínimo” uma função de avaliação $g(S)$, $S = \{x_1, x_2, \dots, x_{n+1}\}$, da forma

$$g(S) = \sqrt{\sum_{i=1}^{n+1} \left(\frac{(f(x_i) - f(\bar{x}))^2}{n+1} \right)},$$

onde $f(x)$ é a função objetivo, e \bar{x} é o nosso “centróide compensado”.

Algoritmo 2 Algoritmo Nelder-Mead

Entrada: Um simplex $\{x_1, x_2, \dots, x_{n+1}\}$, parâmetros $\alpha > 0$, $0 < \beta < 1$ e $\gamma > 1$, um limite de tolerância $\epsilon > 0$ e um número máximo de iterações k_{max} ;

Saída: O ponto de ótimo, aproximado, x^* ;

- 1: $k = 0$;
 - 2: **enquanto** $g(\{x_1, x_2, \dots, x_{n+1}\}) > \epsilon$ e $k < k_{max}$ **faça**
 - 3: Construir uma lista de índices i_1, i_2, \dots, i_{n+1} tal que $f(x_{i_1}) \leq f(x_{i_2}) \leq \dots \leq f(x_{i_{n+1}})$;
 - 4: Nomear $i_1 = l$ e $i_{n+1} = h$;
 - 5: $\bar{x} \leftarrow \sum_{i=1}^{n+1} \frac{x_i}{n} - \frac{x_h}{n}$; // Atribuição de Centróide Compensado
 - 6: $x_r \leftarrow \bar{x} + \alpha(\bar{x} - x_h)$; // Reflexão do vértice x_h do Politopo
 - 7: **se** $f(x_r) < f(x_l)$ **então**
 - 8: $x_e \leftarrow \bar{x} + \gamma(x_r - \bar{x})$; // Operação de Expansão
 - 9: $x_h \leftarrow \arg \min_{x \in \{x_r, x_e\}} f(x)$;
 - 10: **senão**
 - 11: **se** $\exists j \neq h$ tal que $f(x_r) \leq f(x_j)$ **então**
 - 12: $x_h \leftarrow x_r$;
 - 13: **senão**
 - 14: $\tilde{x} \leftarrow \arg \min_{x \in \{x_h, x_r\}} f(x)$;
 - 15: $x_c \leftarrow \bar{x} + \beta(\tilde{x} - \bar{x})$; // Operação de Contração
 - 16: **se** $f(x_c) < f(\tilde{x})$ **então**
 - 17: $x_h \leftarrow x_c$;
 - 18: **senão**
 - 19: **para** $i = 1, \dots, n + 1$ **faça**
 - 20: $x_i \leftarrow \frac{(x_i + x_l)}{2}$; // Operação de Redução
 - 21: **fim para**
 - 22: **fim se**
 - 23: **fim se**
 - 24: **fim se**
 - 25: $k \leftarrow k + 1$;
 - 26: **fim enquanto**
-

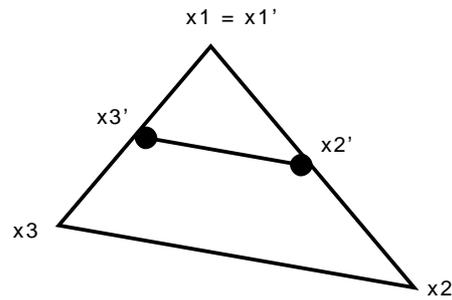


Figura 3.6: Redução dos vértices x_1 , x_2 e x_3 em x_1' , x_2' e x_3' , respectivamente.

3.3 Convergência

O Método de Nelder-Mead é uma Meta-Heurística, sendo assim, é apenas um procedimento de cálculo, e não representa uma solução, teoricamente comprovada, para algum problema. Desta forma, não há demonstração teórica sobre sua convergência, seja local ou global. Apesar do esforço aplicado no passar dos anos para se obter essa comprovação, foram poucos os resultados que obtiveram notoriedade na comunidade acadêmica. Um trabalho que obteve grande atenção da comunidade acadêmica foi (MCKINNON, 1998), onde foi demonstrado que a função

$$f(x_1, x_2) = \begin{cases} \theta\phi|x_1|^\tau + x_2 + x_2^2, & x_1 \leq 0 \\ \theta x_1^\tau + x_2 + x_2^2, & x_1 \geq 0 \end{cases}, \quad (3.7)$$

onde θ , ϕ são constantes positivas e $\tau > 1$ (para garantias que a derivada 1ª é contínua), poderia convergir para um ponto não-estacionário, ao se aplicar o Método Nelder-Mead.

Contudo, não há relatos publicados de falha na convergência para funções mais simples. Portanto, esse método se mantém muito popular para minimização irrestrita sem avaliação de gradiente.

4 BUSCA COORDENADA E BUSCA PADRÃO

Como visto anteriormente, Métodos de Busca constituem uma classe de alta relevância dentro do contexto de Métodos de Otimização Sem Derivada. Aqui, apresentaremos dois Métodos, em ordem crescente de generalização, seus algoritmos e suas características e convergência.

Para que não haja confusão com o termo “ordem crescente de generalização”, queremos deixar claro que o termo significa que o método subsequente engloba os anteriores por meio do conceito de contenção da classe a qual o método se encaixa.

As demonstrações de convergência foram tiradas do trabalho (TORCZON, 1997), que constitui um material muito utilizado por pesquisadores da área, e encerra muito bem esta questão. O uso de Teoria de Dependência Positiva, exposta brevemente no Capítulo 2, ficará claro na seção de demonstração de convergência.

Antes de expormos os métodos vamos fazer uma ressalva sobre a notação utilizada. Para o contexto a seguir é conveniente fazermos a equivalência das colunas de uma matriz, digamos P_k , com um conjunto finito de vetores do espaço E . Daí, ao dizermos que o vetor c pertence a uma matriz P_k (denotado por $c \in P_k$) estamos dizendo que

c pertence a um conjunto formado por vetores extraídos das colunas da matriz P_k .

No mesmo sentido, se nos referimos a união de duas matrizes A e B , estamos nos referindo à união dos conjuntos formados por elementos do espaço E extraídos das colunas de A e B , respectivamente.

4.1 Busca coordenada

Um dos primeiros métodos de busca desenvolvido foi o de Busca Coordenada, onde o conjunto de pontos na vizinhança de x_k é dado em direção aos eixos cartesianos das coordenadas, de forma que o ponto base é somado a um dos vetores canônicos e_i , ou um de seus simétricos $-e_i$, ponderados por um fator Δ_k , como mostrado na Figura 4.1.

O fator Δ_k é, algumas vezes, chamado de tamanho da malha, uma vez que ele define uma malha cartesiana de pontos, onde x_0 , ponto inicial, é o parâmetro de deslocamento da origem.

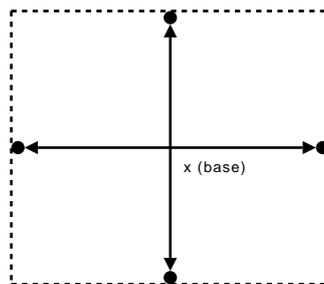


Figura 4.1: Busca Coordenada. Avaliações seguem nas direções dos eixos, transladados pelo ponto $x(\text{base})$.

4.1.1 Algoritmo

Algoritmo 3 Algoritmo de busca coordenada

Entrada: Ponto inicial x_0 , um parâmetro $\Delta_0 > 0$, um limite de tolerância $\epsilon > 0$ e um número máximo de iterações k_{max} ;

Saída: O ponto de ótimo, aproximado, \bar{x} ;

1: $k = 0$;

2: **enquanto** $\Delta_k > \epsilon$ e $k < k_{max}$ **faça**

3: Gerar $P_k = E_+ \cup E_-$, onde

$E_+ = \{y \in \mathbb{R}^n \mid y = x_k + \Delta_k e_i, \text{ para algum } i = 1, \dots, n\}$ e

$E_- = \{y \in \mathbb{R}^n \mid y = x_k - \Delta_k e_i, \text{ para algum } i = 1, \dots, n\}$

4: **para todo** $j = 1, \dots, 2n$ **faça**

5: **se** $f(y_j) < f(x_k)$, onde $y_j \in P_k$ **então**

6: $x_k = y_j$;

7: **fim se**

8: **fim para**

9: **fim enquanto**

4.1.2 Critério de parada

Toma-se o Critério de Parada como sendo o comprimento de refinamento da malha. Se a malha for mais fina que um limite de tolerância $\epsilon > 0$, então o erro entre a resposta e o mínimo local pode ser considerado tolerável. Estando a resposta dentro da região com centro em x_k e pontos determinados pela combinação convexa dos vetores do Padrão, ponderados pelo passo Δ_k .

4.2 Busca padrão

Este método é uma generalização do anterior. Sua generalização não está apenas no fato de haver uma mudança na base de busca, mas também a de inserir o conceito de padrão, para o conjunto de pontos de amostragem da função.

Um Padrão constitui um conjunto finito de pontos contidos na vizinhança do ponto base x_k . Estes pontos são dados sobre um conjunto de direções, as quais constituem um conjunto gerador positivo do espaço euclidiano E , mostrada na Figura 4.2.

Seguindo uma forma utilizada por Elizabeth Dolan e Virginia Torczon em (DOLAN; LEWIS; TORCZON, 2003), o padrão é gerado a partir de uma matriz B não-singular de tamanho n , e uma matriz, $n \times p$, de valores inteiros C_k . Artigos mais antigos fazem menção a uma partição da matriz $P_k = B C_k$ da seguinte forma:

$$C_k = [M_k \quad -M_k \quad L_k \quad 0]$$

onde $M_k \subset M \subset \mathbb{Z}^{n \times n}$ e M é um conjunto limitado de matrizes não-singulares, no sentido que cada componente da matriz é limitada por seus próprios parâmetros, não necessariamente independentes. Isto garante a finitude do conjunto e que as colunas de cada elemento de M formam uma base inteira do espaço E .

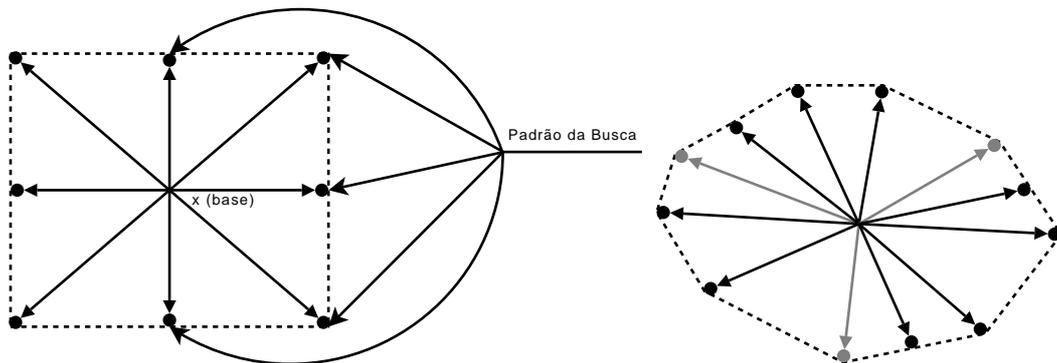


Figura 4.2: Busca Padrão, as avaliações seguem em várias direções a partir do ponto $x(\text{base})$. À direita, Busca Padrão com outro Padrão (linhas em cinza representam uma base positiva minimal).

O conjunto L_k é arbitrário, uma vez que as colunas da matriz $B M_k$ unidas às colunas da matriz $B (-M_k)$ geram positivamente o espaço E , deixando a ele um papel de valores extras para a amostragem. O papel da coluna 0 está relacionado com a necessidade de se comparar o valor da função objetivo de cada elemento do Padrão com o valor do ponto base.

Contudo, a idealização inicial do método estava relacionada a um conjunto gerador suficiente para o espaço E . Com isto em mente, nota-se que o uso das colunas de BM_k e $B(-M_k)$ é desnecessário, pois é possível encontrar um conjunto gerador positivo com $n + 1$ elementos, uma base positiva Γ_k . De fato, o conjunto formado pelas colunas de BM_k e $B(-M_k)$ é uma base positiva maximal, pois as colunas da matriz BM_k formam uma base linear. O conjunto de todas as bases positivas que o algoritmo pode gerar é denotado por Γ .

Desta forma, é mais comum utilizar uma notação na qual se expresse essa condição da base positiva, redefinindo a matriz C_k da seguinte forma:

$$C_k = [\Gamma_k \quad L_k \quad 0]$$

Denotaremos por $c \in C_k$ o vetor c tal que suas componentes se igualam aos valores de uma das colunas da matriz C_k , isto é, $\exists j$ tal que $c_i = (c_k)_{ij}, \forall i = 1, \dots, n$. Assim a notação C_k será usada tanto para nos referirmos à matriz C_k quanto ao conjunto de vetores C_k .

Para fins de teste de Algoritmos da Família de Busca Padrão, pode ser mais interessante utilizar um Padrão Minimal. Visto que quanto maior o número de vetores em um Padrão, maior a chance de algum vetor $d \in P_k$ apresentar um valor positivo alto ao ser projetado ortogonalmente sobre o vetor gradiente $-\nabla f(x)$, isto é, $(d; -\nabla f(x)) = k$, um valor positivo significativamente alto. Na Figura 4.3 ilustramos a como o algoritmo de movimentos exploratórios escolhe a direção do padrão com a maior projeção ortogonal sobre o vetor $-\nabla f(x)$.

Devido a esse aumento de possibilidades, um método tem maior dificuldade de obter um decaimento mais significativo se tomando um Padrão Minimal, devido a redução do tamanho da amostra.

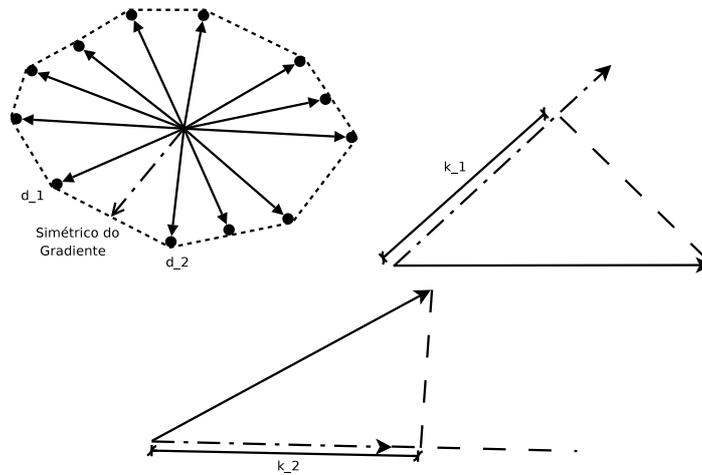


Figura 4.3: Padrão de Busca e simétrico do vetor gradiente (traço-e-ponto). Direção de maior projeção sobre o vetor simétrico é tomada para cálculo de novo iterando.

4.2.1 Critério de parada

O critério de parada do Algoritmo de Busca Padrão se baseia no mesmo critério empregado no Método de Busca Coordenada. Contudo, como o padrão pode apresentar uma geometria qualquer, espera-se que a maneira como o padrão P_k é gerado, permita uma forma bem controlada de se estimar a vizinhança na qual todos os pontos são amostrados para as avaliações.

4.3 Convergência de métodos de busca padrão

A Convergência deste Método, e de Métodos de Busca Padrão em geral, é demonstrada em dois trabalhos: (DOLAN; LEWIS; TORCZON, 2003) e (TORCZON, 1997) garantido por uma propriedade de espaços gerados por direções positivas e por um teorema, apresentado em (DOLAN; LEWIS; TORCZON, 2003), os quais serão listados e demonstrados a seguir.

Algoritmo 4 Algoritmo de busca padrão

Entrada: Ponto inicial x_0 , um parâmetro $\Delta_0 > 0$, um limite de tolerância $\epsilon > 0$ e um número máximo de iterações k_{max} ;

Saída: O ponto de ótimo, aproximado, \bar{x} ;

- 1: $k = 0$;
 - 2: **enquanto** $\Delta_k > \epsilon$ e $k < k_{max}$ **faça**
 - 3: Gerar P_k , um gerador positivo do espaço E
 - 4: **para todo** $j = 1, \dots, m$ onde $m \geq n + 1$ **faça**
 - 5: **se** $f(y_j) < f(x_k)$, onde $y_j \in P_k$ **então**
 - 6: $x_k = y_j$;
 - 7: **fim se**
 - 8: **fim para**
 - 9: **fim enquanto**
-

Definição 4.3.1 Diz-se que o ponto base de amostragem x_k tem uma iteração falha, se a amostragem em todos os pontos do padrão ao seu redor não consegue encontrar um ponto que reduza sua imagem.

4.3.1 Convergência global

Definição 4.3.2 Definimos como $\mathcal{L}(x_0) = \{x \mid f(x) \leq f(x_0)\}$ o conjunto de nível de x_0 .

Teorema 4.3.1 Suponha $f(x) \in C^1(E)$ com $\nabla f(x)$ Lipschitz de constante \mathcal{K} em uma vizinhança aberta $\Omega = \bigcup_{x \in \mathcal{L}(x_0)} \mathcal{B}_\rho(x)$ de $\mathcal{L}(x_0)$, com $\rho > 0$. Então existem um $\delta > 0$ e um $c > 0$ tais que se x_k tem uma iteração falha e $\Delta_k < \delta$ então

$$\|\nabla f(x_k)\| \leq c \Delta_k.$$

Demonstração :

Seja $r = \frac{\min\{1, \rho\}}{2}$. Se $x \in \mathcal{L}(x_0)$, então o aberto $\mathcal{B}_r(x)$ está contido em Ω .

Tomemos apenas os vetores $s = \Delta_k B c_k^i$, $c_k^i \in \Gamma_k \subset \Gamma$. Da finitude de Γ temos que $\|\Gamma_k\|$ é limitado superiormente. Seja Γ^* uma cota superior para $\|\Gamma_k\|$, $k \geq 0$. Assim, $\|s\| \leq \Delta_k \|B\| \Gamma^*$.

$$\text{Tome } \delta = \frac{r}{\|B\| \Gamma^*}.$$

Como requisito, $\Gamma_k \subset \Gamma$ é uma base para o espaço E , isto é, o conjunto $G_k = \{s \mid s \in \Delta_k B \Gamma_k\}$ é uma base para E . Assim, o Colorário 2.2.3 nos dá

$$(-\nabla f(x_k); s) \geq \bar{c} \|\nabla f(x_k)\| \|s\|. \quad (4.1)$$

Já que o algoritmo de movimentos exploratórios falhou, temos que $f(x_k + s) - f(x_k) \geq 0$, $\forall s \in G_k$.

Por hipótese, temos que $\Delta_k < \delta$ e $x_k + s \in \mathcal{B}_r(x_k) \subset \Omega$ - onde s satisfaz a inequação 4.1. Daí podemos utilizar a desigualdade de Cauchy-Schwarz da seguinte forma:

$$0 \leq f(x_k + s) - f(x_k) = (\nabla f(x_k); s) + (\nabla f(\xi) - \nabla f(x_k); s) \leq$$

$$\leq -\bar{c} \|\nabla f(x_k)\| \|s\| + \|\nabla f(\xi) - \nabla f(x_k)\| \|s\|, \text{ onde}$$

$$\xi \in \{x \in \mathbb{R}^n \mid x = ax_k + (1-a)(x_k + s), 0 \leq a \leq 1\} \text{ e daí}$$

$$\bar{c} \|\nabla f(x_k)\| \leq \|\nabla f(\xi) - \nabla f(x_k)\|.$$

Como $\nabla f(x_k)$ é Lipschitz temos

$$\bar{c} \|\nabla f(x_k)\| \leq \mathcal{K} \|\xi - x_k\| \leq \mathcal{K} \|s\| \leq \mathcal{K} \Delta_k \|B\| \Gamma^*.$$

Tomamos, então, $c = \frac{\mathcal{K} \|B\| \Gamma^*}{\bar{c}}$. **QED.**

Para a validade desta demonstração é levado em conta que as iterações foram tais que não foi possível encontrar um decréscimo no valor da função objetivo pelo uso do algoritmo de movimentos exploratórios. É mencionado em (DOLAN; LEWIS; TORCZON, 2003) que Audet em (AUDET, 1998) obteve uma sequência divergente de pontos $\{x_k\}$, porém em todas as iterações foi obtido um decréscimo pelo uso do algoritmo de movimentos exploratórios.

4.3.2 Convergência local

Para a demonstração de convergência local, é necessário supor as seguintes hipóteses sobre a função objetivo $f(x)$.

Hipóteses sobre $f(x)$:

- Hipótese 0 : Os vetores c_k^i obtidos de C_k estão sempre contidos em $\mathcal{B}_{c_0}(0)$, $\forall i$ para um $c_0 > 0$, $\forall k$. E com isso definimos uma constante positiva c_0 que sempre satisfaz $\|s_k\| \leq c_0 \Delta_k$.
- Hipótese 1 : Existe um natural N tal que $\forall k \geq N$ temos Δ_k não é crescente.
- Hipótese 2 : Existe um $r > 0$ tal que se $x \in \mathcal{B}_r(x_*)$ então $f(x) \in C^2(\mathcal{B}_r(x_*))$ e $\nabla^2 f(x) \geq 0$ em $\mathcal{B}_r(x_*)$. Mais ainda, os valores singulares de $\nabla^2 f(x)$ estão contidos em $[\sigma_{min}, \sigma_{max}]$, $\sigma_{min} > 0$.

Para os próximos resultados, utilizaremos as seguintes constantes,

$$\kappa = \frac{\sigma_{max}}{\sigma_{min}}, \quad (4.2)$$

$$\eta = \frac{r}{\|B\| \Gamma^* + 1}. \quad (4.3)$$

Por fim, essa última equação é particularmente útil, pois se $\|x_k - x_*\| < \eta$ e $\Delta_k < \eta$ então para cada $s \in \Delta_k B \Gamma_k$ podemos afirmar que $\|(x_k + s) - x_*\| < r$ pois

$$\|x_k + s - x_*\| - \|s\| \leq \|x_k + s - s - x_*\| = \|x_k - x_*\| < \eta$$

$$\|x_k + s - x_*\| < \eta + \|s\| = \eta + \Delta_k \|B\| \Gamma^* < \eta + \eta \|B\| \Gamma^* = (\|B\| \Gamma^* + 1) \eta = r.$$

Seguem os resultados mencionados.

Proposição 4.3.2 *Se a Hipótese 2 é verdadeira, então existe uma constante $c > 0$ tal que quando um iterando não obtém um decréscimo no valor da função objetivo, com $\Delta_k < \eta$ e $\|x_k - x_*\| < \eta$ então temos que $\|x_k - x_*\| < c\Delta_k$.*

Demonstração :

Utilizando o Colorário 2.2.3, o fato que $(x_k + s) \in \mathcal{B}_r(x_*)$, $s = \Delta_k B c_k^i$ e $c_k^i \in \Gamma_k$ e a desigualdade de Cauchy-Schwarz temos

$$0 \leq f(x_k + s) - f(x_k) \leq -\bar{c} \|\nabla f(x_k)\| \|s\| + \|\nabla f(\xi) - \nabla f(x_k)\| \|s\|$$

$$\bar{c} \|\nabla f(x_k)\| \leq \|\nabla f(\xi) - \nabla f(x_k)\|, \xi \in \{x \in \mathbb{R}^n \mid x = ax_k + (1-a)(x_k + s), 0 \leq a \leq 1\}.$$

Utilizando a forma integral do teorema do valor médio, temos a relações

$$\begin{aligned} \bar{c} \|\nabla f(x_k)\| &\leq \|\nabla f(\xi) - \nabla f(x_k)\| = \left\| \int_0^1 (\nabla^2 f(x_k + t(\xi - x_k)))(\xi - x_k) dt \right\| \leq \\ &\leq \sigma_{max} \|\xi - x_k\| \leq \sigma_{max} \Delta_k \|B\| \Gamma^* \end{aligned}$$

e

$$\|\nabla f(x_k)\| = \|\nabla f(x_k) - \nabla f(x_*)\| = \left\| \int_0^1 (\nabla^2 f(x_k + t(x_k - x_*)))(x_k - x_*) dt \right\| \geq$$

$$\geq \sigma_{\min} \|x_k - x_*\|.$$

Consequentemente,

$$\bar{c}\sigma_{\min} \|x_k - x_*\| \leq \bar{c} \|\nabla f(x_k)\| \leq \sigma_{\max} \|B\| \Gamma^* \Delta_k$$

e assim,

$$c = \frac{\sigma_{\max} \|B\| \Gamma^*}{\sigma_{\min} \bar{c}}.$$

QED.

Proposição 4.3.3 *Se é válida a Hipótese 2 e se existem $x, y \in \mathcal{B}_\eta(x_*)$ tais que $f(x) \leq f(y)$ então $\|x - x_*\| \leq \kappa^{\frac{1}{2}} \|y - x_*\|$.*

Demonstração :

Sob as hipóteses da proposição, e como $\nabla f(x_) = 0$, temos da forma da derivada segunda que*

$$f(y) = f(x_*) + \frac{1}{2} (y - x_*)^T \nabla^2 f(\xi) (y - x_*)$$

$$f(x) = f(x_*) + \frac{1}{2} (x - x_*)^T \nabla^2 f(\omega) (x - x_*)$$

onde

$$\xi \in \{z \in \mathbb{R}^n \mid z = ay + (1-a)x_*, 0 \leq a \leq 1\}$$

e

$$\omega \in \{z \in \mathbb{R}^n \mid z = ax + (1-a)x_*, 0 \leq a \leq 1\}.$$

Segue então, como consequência da Hipótese 2, que

$$0 \leq f(y) - f(x) = \frac{1}{2} (y - x_*)^T \nabla^2 f(\xi) (y - x_*) - \frac{1}{2} (x - x_*)^T \nabla^2 f(\xi) (y - x_*)$$

$$0 \leq \sigma_{\max} \|y - x_*\|^2 - \sigma_{\min} \|x - x_*\|^2$$

$$\sqrt{\frac{\sigma_{max}}{\sigma_{min}}} \|y - x_*\| \geq \|x - x_*\|$$

lembrando que $\kappa = \frac{\sigma_{max}}{\sigma_{min}}$. **QED.**

Proposição 4.3.4 *Se são válidas todas as hipóteses, então existem $\delta > 0$ e $\epsilon > 0$ tais que se $k \geq N$ (definido em Hipótese 1) e se $\Delta_k < \delta$ e $\|x_k - x_*\| < \epsilon$ então, para $l \geq k$, tem-se $\|x_l - x_*\| < \eta$.*

Demonstração :

Tome $\delta < \frac{\eta}{2c_0}$ e $\epsilon < \frac{1}{2}\kappa^{-\frac{1}{2}}\eta \leq \frac{\eta}{2}$, onde c_0 vem da Hipótese 0.

Por indução, considere $x_{k+1} = x_k + s_k$. A Hipótese 0 nos dá a relação $\|x_{k+1} - x_k\| = \|s_k\| \leq c_0\Delta_k$. Assim,

$$\|x_{k+1} - x_*\| \leq \|x_{k+1} - x_k\| + \|x_k - x_*\| < c_0\Delta_k + \epsilon < \eta.$$

Agora, considere um $l \geq k + 1$, e suponha

$$\|x_l - x_*\| < \eta.$$

Então

$$\|x_{l+1} - x_*\| \leq \|x_{l+1} - x_l\| + \|x_l - x_*\|.$$

Pela Hipótese 1, temos que $\Delta_l \leq \Delta_k$ para $l \geq k$. Com isso, $\|x_{l+1} - x_l\| \leq c_0\Delta_l \leq c_0\Delta_k$. Mas, pela hipótese de indução, temos que $x_l \in \mathcal{B}_\eta(x_*)$ e $f(x_l) \leq f(x_k)$, em conjunto com a Proposição 4.3.3 e a hipótese de que $\|x_k - x_*\| < \epsilon$, podemos aferir que

$$\|x_l - x_*\| \leq \kappa^{\frac{1}{2}} \|x_k - x_*\| < \kappa^{\frac{1}{2}}\epsilon$$

e logo,

$$\|x_{l+1} - x_*\| < c_0 \Delta_k + \kappa^{\frac{1}{2}} \epsilon < \eta.$$

QED.

A Proposição 4.3.5 e o Teorema 4.3.6 serão apenas enunciados. As demonstrações podem ser encontradas em (DOLAN; LEWIS; TORCZON, 2003) e/ou suas referências bibliográficas. Eles são fundamentais para a garantia dos iterandos, tenha o Algoritmo de Movimentos Exploratórios falhado ou não. Neles, pensamos nos iterandos como sequências de pontos x_k e de passos Δ_k , onde se considera o ponto crítico x_* como o minimizador.

Proposição 4.3.5 *Suponha que são válidas todas as Hipóteses e que existem $\delta > 0$ e $\epsilon > 0$ dados pela Proposição 4.3.4. Para $k \geq N$ (definido na Hipótese 1), temos que se $\Delta_k < \delta$ e $\|x_k - x_*\| < \epsilon$, então a seqüência $\{\Delta_j\}_{j \in \mathbb{N}}$ converge para 0, isto é, $\lim_{j \rightarrow \infty} \Delta_j = 0$.*

Teorema 4.3.6 *Considere um algoritmo de busca padrão satisfazendo as Hipóteses 0 e 1, com N dado pela Hipótese 1. Suponha que a Hipótese 2 é válida para um ponto x_* em particular. Então existem $\rho > 0$ e $c > 0$ tais que, se para uma iteração K , $K \geq N$, temos o conjunto $A_K = \{x \in \mathbb{R}^n \mid x = x_K + \Delta_K s, s \in BC_k\} \subset \mathcal{B}_\rho(x_*)$ e para a primeira iteração sem decaimento $\bar{K} \geq K$ então para $k \geq \bar{K}$ é válido $\|x_k - x_*\| \leq c \Delta_m(k)$ onde $m(k)$ é a maior iteração até k , inclusive, em que não foi obtido um decaimento. Consequentemente, temos $\lim_{k \rightarrow \infty} x_k = x_*$.*

5 MÉTODO DE OTIMIZAÇÃO NÃO CONVEXA POR ANÁLISE SECANTE

Os métodos vistos anteriormente têm por função buscar o ponto de menor imagem possível, dentro de uma certa tolerância, seguindo critérios baseados no decaimento que a função provê em uma vizinhança do ponto analisado.

Se uma função é convexa, isso garante a convergência do método para um minimizador global. Contudo, caso a função não seja convexa, ou mesmo se as características da função não estão completamente claras, os métodos podem convergir para um minimizador estritamente local ou mesmo para um outro ponto não estacionário.

Neste capítulo, propomos um método baseado em um conjunto de diretrizes de análise e busca por um ponto de mínimo mais próximo do mínimo global. Nele utilizamos metodologias próprias para uma análise das características da função no entorno de uma região de busca. Contudo, ele não se limita a uma região especificamente, buscando por uma outra, mais promissora, sempre visando a proximidade do mínimo global.

Utilizamos uma metodologia de análise já bem consolidada para aproximarmos nossas soluções de um minimizador local e nos inspiramos nesta metodologia para iniciar os procedimentos de nossa análise.

5.1 Formulação dos princípios

O Método de Busca Padrão tem por princípio uma busca por um minimizador em uma região próxima a um ponto, tomado como deslocamento para a definição de uma malha gerada por um Padrão. Essa região não tem, necessariamente, uma forma geométrica regular que se assemelhe a uma bola aberta da Análise Real em espaços \mathbb{R}^n . Isso se deve ao fato de que o conjunto gerador positivo P_k , associado a um ponto base x_k , pode ser formado por vetores de módulos diferentes, levando a um polítopo irregular.

Como já foi provado em (TORCZON, 1997) e em (DOLAN; LEWIS; TORCZON, 2003), o método pode convergir globalmente para um minimizador, sob certas hipóteses de regularidade.

O desenvolvimento do método será baseado em problemas que, por terem uma $f(x)$ essencialmente não convexa, podem levar à dificuldade de convergência para o minimizador global. Não é objetivo produzir exemplos que comprovem a não convergência do método, mas sim formular princípios que ajudem com uma maior flexibilidade no uso da heurística em problemas onde não tenhamos conhecimento de que o seu ponto inicial x_0 , ou seu passo inicial Δ_0 , vão ser adequados para a convergência à solução¹.

Antes do desenvolvimento, vamos colocar uma pequena definição de termos que

¹Para mais referências sobre heurísticas veja (OSMAN; KELLY, 1996).

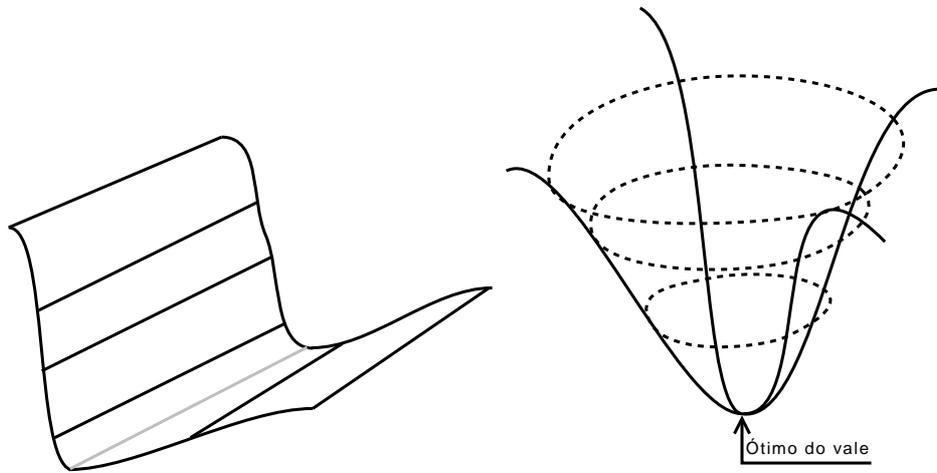


Figura 5.1: Exemplos de Vale. No primeiro, o mínimo é indicado pela reta cinza. No segundo, é apenas um ponto.

usaremos ao longo do capítulo.

Definição 5.1.1 *Associamos o termo **vale** a uma região onde podemos obter um ou mais minimizadores locais tais que o conjunto de minimizadores é finito ou que este conjunto tem dimensão 1, isto é, que nesta região haja apenas alguns pontos minimizadores, ou que os pontos minimizadores estejam em uma reta. Uma ilustração pode ser encontrada na Figura 5.1.*

Para auxiliar no processo de construção, considere o seguinte problema: Uma função que apresente um pequeno vale local, mas com um minimizador global em um vale mais profundo (ilustramos o caso na Figura 5.2). É possível escolher um ponto inicial no interior deste vale e um passo inicial tal que todos os pontos do Padrão sejam mantidos em seu interior. Pelo Método de Busca Padrão, o minimizador local será o atrator da solução do problema e, portanto, será a resposta obtida. Em muitas aplicações, esta resposta é mais que satisfatória. Porém em outras, a necessidade de um minimizador global é real.

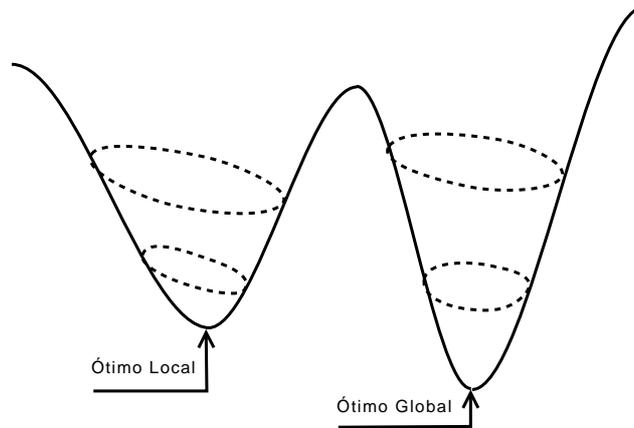


Figura 5.2: Ilustração de função não Convexa.

Problemas que apresentam uma função objetivo muito complexa, tendem a ter uma estimativa de ponto inicial e passo inicial imprecisa para convergir a um minimizador global. Neste sentido, torna-se interessante criar dispositivos que tentem evitar esta convergência estritamente local.

O método que desejamos formular se apresenta como uma análise para tais casos. Tal análise obteve inspiração nos Métodos de Busca Padrão e Nelder-Mead. Ao longo do capítulo, exporemos nossa abordagem analítica e, ao final dele, ficará claro o relacionamento entre os dois.

5.2 Passo externo e secante no vale

Para evitar que o método fique preso a um vale local, sugerimos a adoção de um parâmetro adicional que chamamos de **Passo Externo** e denotamos por Δ_k^e . Seguindo essa nomenclatura, o parâmetro Δ_k é chamado de **Passo Interno**. Denominamos, ainda, o Padrão gerado pela multiplicação por Δ_k^e de **Padrão Externo**. Em contraposição o Padrão multiplicado por Δ_k é chamado de **Padrão Interno**,

deixando assim o nome de **Padrão Total** à união dos dois Padrões.

A fim que o Passo Externo tenha um efeito bem significativo, é mais natural que ele seja escolhido tal que $\Delta_k \ll \Delta_k^e$. Porém, só aumentar o tamanho do Passo de Busca no método é uma maneira insuficiente de evitar problemas como o dado pelo vale local. Poderíamos modificar o Passo Externo a fim que ele cresça indefinidamente até que a exploração nesse Padrão Externo leve a um ponto mais próximo do minimizador global.

Contudo, caso o vale onde estamos localizados seja o vale que nos leva a um minimizador global, estaríamos apenas aumentando desnecessariamente o tamanho do Padrão Total de Busca. Para que esta modificação na Heurística fique mais focalizada neste princípio de saída de um vale estritamente local, vamos focar nossa abordagem em uma análise de malha polar com dois raios, isto é, analisaremos pontos em uma vizinhança próxima e na fronteira de uma vizinhança um tanto quanto distante.

5.3 Estrutura do capítulo

Trataremos da estrutura de organização deste capítulo nesta seção. Aqui queremos organizar as idéias do leitor com um breve resumo de alguns pontos que virão a seguir.

Na seção **análise secante**, formulamos os dados de análise através do uso de segmentos de reta adjacentes e secantes à função objetivo, nos pontos da nossa malha polar. Levantamos os casos que podemos ter, apresentamos ilustrações para melhorar o entendimento, e enunciamos funções que utilizaremos adiante.

Na seção de **análise de casos com secantes e fundamentos de decisão** abordamos todos os casos que podem ocorrer ao se utilizar a Análise Secante sobre a nova malha proposta. Levantamos os principais pontos de análise de Casos de Possíveis Vales, Decaimento Melhorado, os casos de Degeneração e o Critério de Parada por Análise de Certeza de Vale. É também formulada uma variação do Algoritmo de Movimentos Exploratórios, que chamamos de Algoritmo de Movimentos Exploratórios Analítico.

A seção seguinte trata do **critério de decisão de novo iterando**, onde o Método de Nelder-Mead é uma fonte de inspiração.

A seção **Método MONCAS** trata apenas de apresentar o algoritmo e um fluxograma para facilitar o entendimento do método.

A seção de **Passo Externo** é uma parte do texto onde discutimos as principais questões deste parâmetro, a ferramenta inicial que este método usa. Nela discutimos muitas questões que podem ser abordadas em pesquisas futuras.

5.4 Análise secante

Levando em consideração a forma do vale e lembrando que estamos sempre trabalhando com figuras do espaço $\mathbb{R}^n \times \mathbb{R}$, fomos estimulados a utilizar o conceito de curvatura para estipular um critério de parada nessa exploração com o Passo Externo. Para tanto, deve-se lembrar que um Método de Busca Direta não pode ter aproximações para o gradiente ou aplicar explicitamente um modelo de regularidade, como Aproximação de Mínimos Quadrados ou Interpolação. Assim, utilizaremos não uma curvatura explícita, mas uma estrutura que a faz lembrar.

Considere dois segmentos de reta adjacentes no \mathbb{R}^2 . Se os extremos destes segmentos pertencem ao gráfico de uma função, então cada um destes segmentos forma uma secante deste gráfico. Como a derivada é um limite para a reta secante, temos na secante um instrumento muito útil para estimar se em algum ponto houve mudança de inclinação do gráfico desta função.

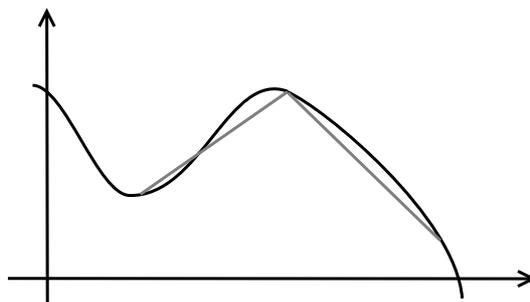


Figura 5.3: Secantes adjacentes em um gráfico de função

Na Figura 5.3, a inclinação do primeiro segmento de reta é uma inclinação positiva e a inclinação do segundo é uma inclinação negativa. Assim, dadas duas secantes adjacentes, podemos ter as seguintes possibilidades para analisar:

- (positivo e positivo) - a primeira secante e a segunda apresentam inclinação positiva;
- (positivo e negativo) - a primeira secante tem inclinação positiva e a segunda negativa;
- (negativo e positivo) - o inverso da anterior;
- (negativo e negativo) - ambas com a inclinação negativa.

Sob o contexto do método, as secantes devem ser tomadas nas direções dos vetores do padrão. Sendo assim, devemos tomar um vetor $d \in P_k$ do padrão para que possamos

definir os seguimentos de reta adjacentes de modo a ligar os pontos $(x_k, f(x_k))$, $(x_k + \Delta_k d, f(x_k + \Delta_k d))$ e $(x_k + \Delta_k^e d, f(x_k + \Delta_k^e d))$. Denominaremos o primeiro segmento de S_1 e o segundo de S_2 .

Cada um dos segmentos tem um coeficiente de inclinação associado. Para S_1 temos

$$v_1 = \frac{f(x_k + \Delta_k d) - f(x_k)}{\|\Delta_k d\|} = \frac{f(x_k + \Delta_k d) - f(x_k)}{\Delta_k \|d\|}$$

e para S_2 temos

$$v_2 = \frac{f(x_k + \Delta_k^e d) - f(x_k + \Delta_k d)}{\|(\Delta_k^e - \Delta_k) d\|} = \frac{f(x_k + \Delta_k^e d) - f(x_k + \Delta_k d)}{(\Delta_k^e - \Delta_k) \|d\|}.$$

Considerando todo o padrão de busca, nos vemos obrigados a formalizar as definições de v_1 e v_2 como funções de x_k , $d \in P_k$, Δ_k e Δ_k^e . Assim,

$$v_1 = v_1(x_k, d, \Delta_k) = \frac{f(x_k + \Delta_k d) - f(x_k)}{\Delta_k \|d\|} \text{ e}$$

$$v_2 = v_2(x_k, d, \Delta_k, \Delta_k^e) = \frac{f(x_k + \Delta_k^e d) - f(x_k + \Delta_k d)}{(\Delta_k^e - \Delta_k) \|d\|}.$$

Como v_1 e v_2 representam inclinações de segmentos de reta adjacentes, e estamos interessados em suas relações, vamos considerar seus casos relativos.

Para casos onde $v_1 > \mathbf{0}$ definimos um coeficiente que denotaremos por t que mede a razão de v_2 por v_1 , isto é,

$$\frac{v_2(x_k, d, \Delta_k, \Delta_k^e)}{v_1(x_k, d, \Delta_k)} = \frac{v_2}{v_1} = t.$$

Temos as seguinte possibilidades para distinção de valores do coeficiente t :

- $t < 1$

- $t = 1$
- $t > 1$

Ilustramos essas possibilidades nas Figuras 5.4, 5.5 e 5.6, respectivamente.

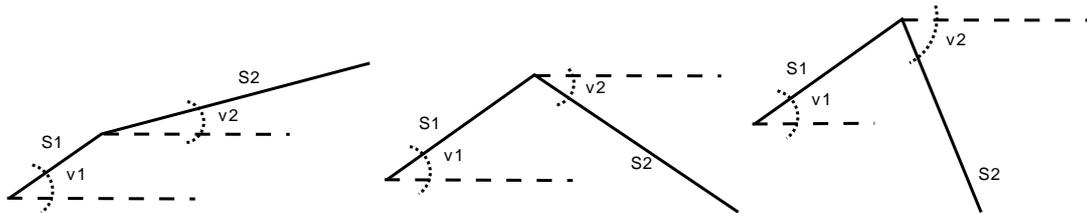


Figura 5.4: Esboço das secantes implicando em um critério $t < 1$.

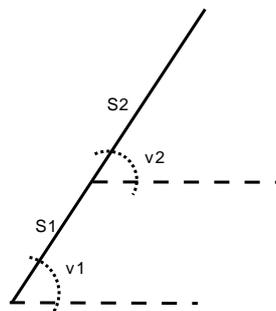


Figura 5.5: Esboço das secantes implicando em um critério $t = 1$.

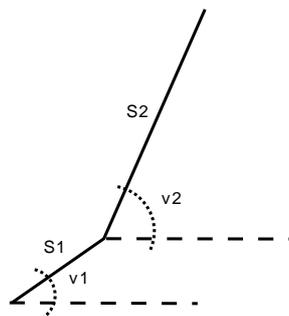


Figura 5.6: Esboço das secantes implicando em um critério $t > 1$.

Os casos ilustrados representam todas as possibilidades para a detecção de presença de um vale.

Para casos onde $\mathbf{v}_1 < \mathbf{0}$ definimos um coeficiente que denotaremos por τ que mede a razão de v_2 por v_1 , isto é,

$$\frac{v_2(x_k, d, \Delta_k, \Delta_k^e)}{v_1(x_k, d, \Delta_k)} = \frac{v_2}{v_1} = \tau$$

temos as seguinte possibilidades para distinção de valores do coeficiente τ :

- $\tau < 1$
- $\tau = 1$
- $\tau > 1$

Ilustramos essas possibilidades nas Figuras 5.7, 5.8 e 5.9, respectivamente.

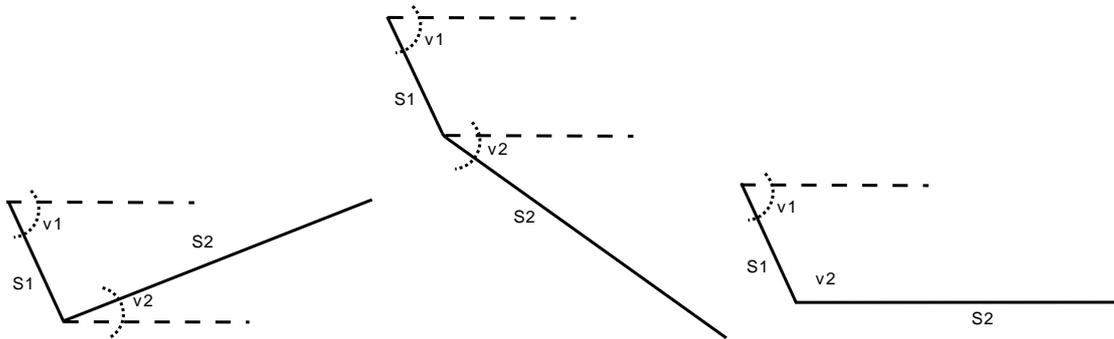


Figura 5.7: Esboço das secantes implicando em um critério $\tau < 1$.

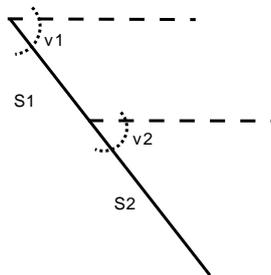


Figura 5.8: Esboço das secantes implicando em um critério $\tau = 1$.

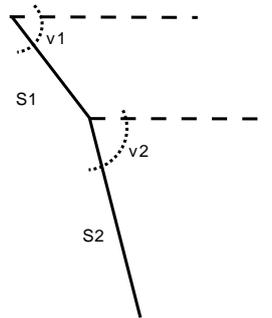


Figura 5.9: Esboço das secantes implicando em um critério $\tau > 1$.

Estes casos, das Figuras 5.8 e 5.9, representam possibilidades de decaimento melhorado, onde podemos obter um decréscimo maior na função objetivo dentro da mesma iteração do método.

Para os casos onde $\mathbf{v}_1 = \mathbf{0}$ temos uma série de comparações a serem feitas, para todas as possibilidades de inclinação do segmento S_2 relativo ao segmento S_1 , isto é, as possibilidades dependendo exclusivamente do coeficiente v_2 .

No caso onde $v_2 < 0$ temos um ganho no valor da função objetivo para o Passo Externo. No caso onde $v_2 > 0$ temos um caso de vale a uma distância longa. Finalmente, o caso $v_2 = 0$ não nos ajuda muito na progressão do algoritmo, porém pode indicar um vale em linha como indicado na Figura 5.1. Os casos para os valores de v_2 são ilustrados nas Figuras 5.10, 5.11 e 5.12, respectivamente.

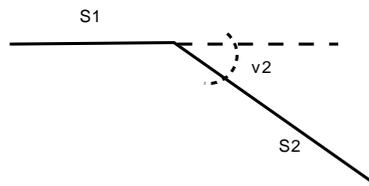


Figura 5.10: Caso $v_1 = 0$ com inclinação $v_2 < 0$.



Figura 5.11: Caso $v_1 = 0$ com inclinação $v_2 = 0$.

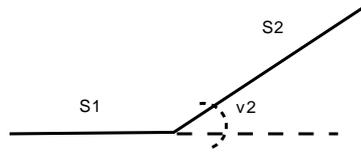


Figura 5.12: Caso $v_1 = 0$ com inclinação $v_2 > 0$.

Os casos onde $v_1 = 0$ apresentam um grupo de possíveis **microvales** locais, e necessitam de uma análise rigorosa dos pontos intermediários dos segmentos de $(x_k, x_k + \Delta_k d)$ ou $(x_k + \Delta_k d, x_k + \Delta_k^e d)$ para analisar corretamente os casos nesses intermédios.

Dado isso, podemos passar ao ponto onde analisaremos nossas opções para decidirmos os critérios mais adequados ao decaimento e a fuga de mínimos estritamente locais.

5.5 Análise de casos com secantes e fundamentos de decisão

Depois de definidas as formas das secantes, observamos que temos duas opções: analisar casos onde $v_1(x_k, d, \Delta_k) > 0$, pois nossa informação nos leva a crer que estamos presos em um vale, ou casos onde $v_1(x_k, d, \Delta_k) < 0$, e daí tentar obter um decaimento significativo na nossa busca local.

Casos onde $v_1 = 0$ serão considerados como degeneração. Sob a ótica do Método de Busca Padrão, se isso ocorre, então nesta direção há uma falha no Algoritmo de Movimentos Exploratórios. Assim, a busca com Passo Externo é uma forma de tentar reavaliar essa direção.

Apesar de simples, esta análise é bem robusta, pois não precisa de muitas hipóteses, e sua convergência é garantida localmente por hipóteses regendo o Método de Busca

Padrão em si.

Explicitamente nos referiremos aos casos onde $v_1 > 0$ como **casos de possíveis vales**, aos casos onde $v_1 < 0$ como **casos de decaimento melhorado** e quando $v_1 = 0$ diremos que temos um **caso degenerado**.

A análise destes casos deve ser seguida de um conjunto de valores para uma análise final de potencialidade de mínimo. Após cada uma destas avaliações, devem ser retornados valores de ponto de deslocamento onde houve ganho no valor da função objetivo ou se descobriu um ponto num outro vale.

5.5.1 Casos de possíveis vales

Tomando apenas os $d \in P_k$ tais que $v_1(x_k, d, \Delta_k) > 0$, nos preocupamos em analisar as possibilidades para $\frac{v_2(x_k, d, \Delta_k, \Delta_k^e)}{v_1(x_k, d, \Delta_k)} = \frac{v_2}{v_1} = t$. As possibilidades para t são:

- $t < 1$ - Significa que há uma possibilidade de mudança de curvatura nesta direção;
- $t = 1$ - Não apresenta tendência para comprovação de permanência de vale ou mudança para um outro;
- $t > 1$ - Significa que há uma evidência maior de que não há outro possível minimizador nesta direção.

Das possibilidades apresentadas anteriormente, apenas a primeira merece investigação subsequente, pois as demais apenas favorecem o uso do Algoritmo de Busca Padrão puro.

Sugerimos uma investigação subsequente à ocorrência de $t < 1$ para determinar a viabilidade de se alterar o ponto base. Se nossa ação fosse apenas selecionar $x_k + \Delta_k^e$ seria possível que este ponto fosse tomado como o próximo iterando. Porém, como não houve qualquer análise sobre ele, seria possível que na nova iteração voltássemos a x_k como possível iterando. Isso não garantiria um novo iterando em uma região de decaimento da função. Chamemos x' o ponto a ser avaliado devido à obtenção de um valor $t > 1$. Como queremos entrar em uma região de decaimento, vamos analisar as imagens da função $f(x)$ nos pontos x' e $x^e = x_k + \Delta_k^e d$. Caso haja decaimento, tomamos este ponto (x') para análise posterior. Caso contrário, descartamos o mesmo.

No caso de descarte, não é interessante enviar esses pontos e suas inclinações para uma segunda análise. Já obtivemos um valor $t < 1$ analisando na região de Passo Interno e Passo Externo. Ao tomarmos um x' relevantemente distante de x_k , perdemos o sentido de analisar a certeza de vale nesta direção. Ao mesmo tempo, um x' distante de x_k e com um decaimento relativo a x^e nos abre possibilidade de analisarmos uma outra região de decaimento. Assim, e para que não fiquemos muito próximos do vale atual, tomamos $x' = x_k + 2\Delta_k^e d$.

Denominaremos esse procedimento de **análise de possibilidade de novo vale** e, caso haja possibilidade de troca de ponto base em consequência direta deste procedimento, denominaremos esta troca por **substituição por possibilidade de novo vale**. Apresentamos o pseudo-código para esta análise no Algoritmo 5.

Como não é garantida a melhoria do resultado, é melhor guardar a posição atual e, talvez para uma avaliação mais detalhada, o padrão atual e os passos também. Pois uma nova avaliação por esta região pode levar ao melhor resultado. Se este pensamento for mantido, a heurística atual se comporta como um mapeamento da função, no sentido de poder estimar seus pontos críticos dentro de certa precisão.

Algoritmo 5 Algoritmo de análise de possibilidade de novo vale

Entrada: O ponto base x_k , direção d , Passo Interno Δ_k e Passo Externo Δ_k^e ;

Saída: Uma variável lógica **vale**;

- 1: $x' \leftarrow x_k + 2\Delta_k^e d$;
 - 2: $x^e \leftarrow x_k + 2\Delta_k^e d$;
 - 3: **se** $f(x^e) > f(x')$ **então**
 - 4: **vale** \leftarrow **verdadeiro**;
 - 5: **senão**
 - 6: **vale** \leftarrow **falso**;
 - 7: **fim se**
-

Ao final desta análise, devemos retornar duas listas. Uma lista contendo itens da seguinte forma:

- Ponto x' onde pode ser feita Substituição por Possibilidade de Novo Vale;

Denotaremos esta lista por L_k .

A outra lista deve ser formatada a fim que se faça uma análise para parada do uso desta análise. Esta lista deve ser formada exclusivamente por dados onde $v_1 > 0$ e não foi feita a **Análise de Possibilidade de Vale**.

Nesta lista precisamos armazenar os seguintes dados:

- A direção d sobre a qual foi feita a análise;
- O valor do Coeficiente t resultante.

Denotaremos esta lista por V^+ , lista para vales.

5.5.2 Decaimento melhorado

Discutiremos nesta seção os casos onde nós já pudemos encontrar decaimento na análise local proveniente do uso do Método de Busca Padrão. Nosso objetivo é estender essa análise de forma que possamos obter um ganho ainda maior na função objetivo.

Tomando apenas os $d \in P_k$ tais que $v_1(x_k, d, \Delta_k) < 0$, nos preocupamos em analisar as possibilidades para $\frac{v_2(x_k, d, \Delta_k, \Delta_k^e)}{v_1(x_k, d, \Delta_k)} = \frac{v_2}{v_1} = \tau$. As possibilidades para τ são:

- $\tau \geq 1$ - Significa que há um decaimento maior, imediato ao se tomar o ponto obtido pelo Passo Externo;
- $\tau < 1$ - Não nos dá certeza de decaimento ou aumento da função ao se tomar o Passo Externo.

A primeira possibilidade é ilustrada nas Figuras 5.8 e 5.9. A segunda é ilustrada nas três possibilidades apresentadas na Figura 5.7. A segunda opção carece de uma segunda análise no valor de $v_2(x_k, d, \Delta_k, \Delta_k^e)$.

Daí, somente se $v_2(x_k, d, \Delta_k, \Delta_k^e) < 0$, é imediatamente interessante trocar o ponto base x_k para o ponto $x_k + \Delta_k^e d$, pois ainda é um caso de decaimento melhorado. O caso onde $v_2(x_k, d, \Delta_k, \Delta_k^e) = 0$ constitui uma **degeneração com estabilização**, o que nos pode ser útil em alguns casos onde podemos nos encontrar em um vale e que o mínimo esteja em um raio de busca menor que Δ_k . É interessante tomarmos aqui uma fração muito pequena do comprimento do segmento de reta S_2 . Ilustramos essa situação na Figura 5.13. O caso onde $v_2(x_k, d, \Delta_k, \Delta_k^e) > 0$ é o caso que limita o decaimento a uma vizinhança muito pequena do ponto $x_k + \Delta_k d$, uma **degeneração**

com vale próximo. Análises subsequentes nesta região devem levar em conta um pequeno passo.

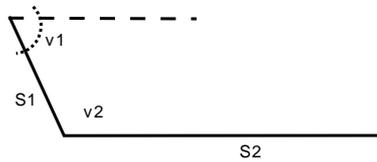


Figura 5.13: Ganho em Passo Interno com alerta para degeneração no Passo Externo.

O decaimento melhorado deve retornar uma lista L_k com os seguintes dados:

- Ponto $x_k + \Delta_k^e d$ onde houve decaimento melhorado, ou $x_k + \Delta_k d$ se houver degeneração de qualquer tipo.

Essa divisão nos deixa com 3 possibilidades

1 $\tau \geq 1$ ou ($\tau < 1$ e $v_2 < 0$);

2 $\tau < 1$ e $v_2 > 0$;

3 $\tau < 1$ e $v_2 = 0$.

A possibilidade **1** gera um item da lista L_k com ponto centrado em $x_k + \Delta_k^e d$. A segunda possibilidade é gerada pela existência de um vale entre x_k e $x_k + \Delta_k^e d$. Finalmente, a possibilidade **3** gera um item da lista L_k com ponto centrado em $x_k + \Delta_k d$.

5.5.3 Degeneração

O caso onde $v_1(x_k, d, \Delta_k) = 0$ é o mais complicado, pois ele sugere cessamento de variação da função nesta direção, o que chamaremos de **estabilização**. A análise de $v_2(x_k, d, \Delta_k, \Delta_k^e)$ é que nos dará uma idéia melhor sobre o comportamento da função neste entorno.

Caso $v_2(x_k, d, \Delta_k, \Delta_k^e) < 0$, temos um decaimento e podemos considerar que o segmento $x_k + \Delta_k d$ tem uma certa concentração de pontos críticos, com um decaimento subsequente. Assim esse ponto $(x_k + \Delta_k^e d)$ deve ser avaliado subsequentemente. Isso foi ilustrado na Figura 5.12.

Para o caso de $v_2(x_k, d, \Delta_k, \Delta_k^e) > 0$, temos indícios que esse segmento S_1 indica que x_k está em uma região de mínimo local e que na direção d analisada encontramos evidência de que S_1 representa um vale como na Figura 5.1. Isso descarta d de maiores análises pois não há indícios de melhora do valor da função objetivo para maiores investigações. A Figura 5.10 ilustra este caso.

O pior caso de análise é $v_2(x_k, d, \Delta_k, \Delta_k^e) = 0$. Como temos por hipótese que $\Delta_k^e \gg \Delta_k$, o segmento S_2 que gera o valor $v_2(x_k, d, \Delta_k, \Delta_k^e)$ é muito extenso e não conserva características suficientes da vizinhança de $x_k + \Delta_k d$ para que possamos inferir algo. Isto foi ilustrado na Figura 5.11.

Neste caso, uma maneira mais eficaz de proceder é analisar pontos intermediários do segmento de reta S_2 para podermos ter alguma noção de comportamento da função $f(x)$ nessa região. Mas, para cada ponto intermediário considerado em S_2 , aumentamos em 1 o número de segmentos de reta ligando $x_k + \Delta_k d$ e $x_k + \Delta_k^e d$ e multiplicamos por 3 o número de possibilidades da árvore de decisão (começando por um mínimo de 6, para a inserção de 1 ponto intermediário). Devido a este alto

custo computacional, consideramos este caso muito problemático para se analisar, e não adicionamos a direção d em uma lista para futura análise.

Os casos onde $v_2 < 0$ nos levarão a criar uma lista contendo o seguinte dado:

- O ponto $x_k + \Delta_k^e d$;

Os casos onde $v_2 > 0$ serão considerados como casos de vale, mas com uma depreciação para o valor do coeficiente t . Assim, eles serão armazenados na lista V_k^+ com direção d e coeficiente $t = 0$.

5.5.4 Critério de parada por análise de certeza de vale

Essa nossa análise, apesar de rica em possibilidades, é muito custosa pois estamos não só dobrando o número de análises diretas, como levantando muitos outros dados no decorrer da nossa análise. Para evitar que isso onere muito o algoritmo vamos colocar uma salvaguarda para que não haja um acesso a este módulo caso estejamos em uma função globalmente convexa, ou fiquemos presos em um vale extremamente mal dimensionado para os parâmetros.

Suponha que para o padrão P_k tenhamos apenas obtido vetores $d \in P_k$ pertencentes aos dados da lista V_k^+ e tais que gerem um coeficiente $t > 1$, isto é, nossa busca por inclinações favorece apenas os vales. Então para decidirmos o momento de abandonar esta análise, vamos analisar as inclinações obtidas após algumas iterações sob esta condição.

Estando em uma situação tal que $t(x_k, d, \Delta_k, \Delta_k^e) > 1, \forall d \in P_k$, temos que levantar a possibilidade que entramos em um vale muito profundo, ou que a função seja

convexa em uma região muito extensa do domínio. A possibilidade de vale profundo não deve impedir que a busca por mudança de inclinação continue, mas se Δ_k^e for grande o suficiente, devemos nos contentar com as informações obtidas da função até o presente momento para que não fiquemos em uma busca incessante e infrutífera.

Analisando a função $t(x_k, d, \Delta_k, \Delta_k^e)$, notamos que ela suprime em muito o fator de crescimento, para o exterior de x_k , da função no segmento S_2 relativo ao crescimento no segmento S_1 , expressando isso matematicamente, temos

$$t = \frac{f(x_k + \Delta_k^e d) - f(x_k + \Delta_k)}{f(x_k + \Delta_k d) - f(x_k)} \frac{\Delta_k}{\Delta_k^e - \Delta_k}, \text{ onde } \frac{\Delta_k}{\Delta_k^e - \Delta_k} \ll 1$$

Vemos que o fator $\frac{\Delta_k}{\Delta_k^e - \Delta_k}$ suprime, em muito, a taxa de crescimento relativo da função $f(x)$. A partir daí estabelecemos uma hierarquia de níveis para a função t . Antes, como foi definido a **análise por possibilidade de novo vale**, iremos definir sua contraparte: a **análise por certeza de vale**. Com ela, queremos afirmar que o vale atual não oferece uma “possibilidade de fuga”.

Definição 5.5.1 *Sejam $p \in \mathbb{N}$ e $t(x_k, d, \Delta_k, \Delta_k^e) > 1, \forall d \in P_k$. Dizemos que t é de nível de certeza de vale p se $p \leq t < p + 1$, isto é, $p = \lfloor t \rfloor$.*

Assim se t apresentar nível de certeza de vale crescente com o número das iterações, já não é mais recomendável continuar a utilização das avaliações de mudança de inclinação. Com isso queremos estabelecer que, se houver blocos de iterações tais que o Nível de Certeza de Vale $p(t)$ é crescente em relação à iteração k , gerando uma taxa $\frac{\delta p}{\delta k} > 0$, então podemos voltar a aplicar a rotina tradicional do Método de Busca Padrão.

Contudo t é calculado para cada $d \in P_k$, daí temos que tirar uma taxa média \bar{t}_k . Armazenaremos o valor de $p(\bar{t}_k) = \bar{p}_k$ como base de comparação.

O critério de parada será $\frac{\bar{p}_{k+1} - \bar{p}_k}{1} = \bar{p}_{k+1} - \bar{p}_k > 0$, visto que os valores de k são inteiros naturais e sequenciais.

O pseudo-código para o **cálculo de certeza de vale** é apresentado no Algoritmo 6 e o pseudo-código para a **análise de certeza de vale** é apresentada no Algoritmo 7.

Algoritmo 6 Algoritmo de cálculo de nível de certeza de vale médio

Entrada: Lista de Direções V_k^+ ;

Saída: $p_k \leftarrow \left\lfloor \frac{\bar{t}}{m} \right\rfloor$;

- 1: $m \leftarrow \#V_k^+$;
 - 2: $\bar{t} \leftarrow 0$;
 - 3: **para** $i = 1, \dots, m$ **faça**
 - 4: $\bar{t} \leftarrow \bar{t} + V_k^+[i].t$;
 - 5: **fim para**
-

Algoritmo 7 Algoritmo de avaliação de parada de análise de vale

Entrada: Lista V_k^+ , um Padrão P_k e uma variável lógica para quando o Nível de Vale já foi **calculado** na iteração anterior;

Saída: Uma variável lógica para **continuar** a execução do método e uma outra para caso o Nível de Vale tenha sido **calculado**;

- 1: **continuar** \leftarrow **verdadeiro**;
 - 2: **se** $\#V_k^+ = \#P_k$ **então**
 - 3: Aplicar o **Algoritmo de Nível de Certeza de Vale Médio** e armazenar em p_k ;
 - 4: **se** **calculado** = **verdadeiro** e $p_{k-1} - p_k > 0$ **então**
 - 5: **continuar** \leftarrow **falso**;
 - 6: **senão se** **calculado** = **falso** **então**
 - 7: **calculado** = **verdadeiro**;
 - 8: **senão se** **calculado** = **falso** e $p_{k-1} - p_k \leq 0$ **então**
 - 9: **calculado** = **falso**;
 - 10: **fim se**
 - 11: **senão**
 - 12: **calculado** = **falso**;
 - 13: **fim se**
-

5.5.5 Algoritmo de movimentos exploratórios analítico

No Método de Busca Padrão, o algoritmo de movimentos exploratórios é usado para encontrar o ponto de maior decaimento interno na vizinhança do ponto x_k sob a ótica do padrão P_k . Caso não encontre nenhum, é dito que ele falhou.

Para agilizar nossas análises externas, vamos modificar levemente este algoritmo de forma que ele retorne todas as direções analisadas e não somente a de melhor ganho. Com isso não é possível dizer que ele falhou em obter seu resultado, visto que ele sempre vai chegar a uma conclusão para um padrão finito.

Queremos que este algoritmo nos gere 3 listas básicas, que vão separar as direções $d \in P_k$ nas suas três variedades:

- $d \in P_k$ tal que $v_1(x_k, d, \Delta_k) < 0 \implies$ lista M_k^- ;
- $d \in P_k$ tal que $v_1(x_k, d, \Delta_k) = 0 \implies$ lista M_k^0 ;
- $d \in P_k$ tal que $v_1(x_k, d, \Delta_k) > 0 \implies$ lista M_k^+ .

Com isso podemos agilizar a nossa análise fazendo uma única varredura em P_k para os dados internos.

Um segundo ponto de agilidade está em armazenarmos o valor de $f(x)$ obtido em cada exploração nas listas iniciais.

O pseudo-código para este algoritmo é apresentado no Algoritmo 8.

Algoritmo 8 Algoritmo de movimentos exploratórios analítico

Entrada: Ponto base x_k , um padrão P_k e um passo Δ ;

Saída: As listas M_k^- , M_k^0 e M_k^+ ;

- 1: **para todo** $d \in P_k$ **faça**
 - 2: Calcular $v_1(x_k, d, \Delta_k)$;
 - 3: **se** $v_1(x_k, d, \Delta_k) < 0$ **então**
 - 4: Adicionar a direção d e o valor $v_1(x_k, d, \Delta_k)$ na lista M_k^- ;
 - 5: **senão se** $v_1(x_k, d, \Delta_k) = 0$ **então**
 - 6: Adicionar a direção d e o valor $v_1(x_k, d, \Delta_k)$ na lista M_k^0 ;
 - 7: **senão se** $v_1(x_k, d, \Delta_k) > 0$ **então**
 - 8: Adicionar a direção d e o valor $v_1(x_k, d, \Delta_k)$ na lista M_k^+ ;
 - 9: **fim se**
 - 10: **fim para**
-

5.6 Critério de decisão para novo iterando

A decisão sobre qual ponto escolher para o próximo iterando, deve ser tomada sobre os pontos fornecidos pela lista L_k . Inicialmente tomaremos uma decisão utilizando um critério que mescla as estruturas do Método de Busca Padrão e do Método Nelder-Mead.

A lista L_k fornece um grupo de pontos que podem formar um politopo, mesmo que degenerado, do R^n . Analisaremos a imagem da função $f(x)$ em cada um destes pontos e separaremos o ponto de maior imagem x_h e o ponto de menor imagem x_l . Calcularemos o centróide compensado o ponto x_c do politopo, tal qual o algoritmo do Método Nelder-Mead.

Caso esse ponto (x_c) tenha imagem menor que $f(x_l)$, tomaremos x_c como o próximo iterando ($x_{k+1} = x_c$). Se isso não se verificar, utilizaremos uma reflexão entre este ponto x_c e o ponto x_h por um fator s , que é calculado como a metade do comprimento médio das arestas do politopo gerado pelos pontos apresentados na lista L_k . Denominaremos o ponto refletido por x_r .

Algoritmo 9 Algoritmo de análise de mudança de curvatura

Entrada: O ponto base x_k , padrão P_k , Passo Interno Δ_k e Passo Externo Δ_k^e ;

Saída: As listas L_k e V_k^+

- 1: Obter as listas M_k^- , M_k^0 e M_k^+ do **Algoritmo de Movimentos Exploratórios Analítico** dados ponto base x_k , padrão P_k e passo Δ_k ;
 - 2: **para todo** $d \in M_k^-$ **faça**
 - 3: Avaliar $f(x_k + \Delta_k^e d)$ e armazenar em f_2 ;
 - 4: Calcular v_1 , v_2 e τ ;
 - 5: **se** $\tau \geq 1$ ou $(\tau < 1$ e $v_2 < 0)$ **então**
 - 6: Adicionar o ponto $x_k + \Delta_k^e d$ na lista L_k ;
 - 7: **senão se** $\tau < 1$ e $v_2 = 0$ **então**
 - 8: Adicionar o ponto $x_k + \Delta_k d$ na lista L_k ;
 - 9: **senão se** $\tau < 1$ e $v_2 = 0$ **então**
 - 10: Adicionar o ponto $x_k + \Delta_k d$ na lista L_k ;
 - 11: **fim se**
 - 12: **fim para**
 - 13: **para todo** $d \in M_k^0$ **faça**
 - 14: Calcular v_2 ;
 - 15: **se** $v_2 < 0$ **então**
 - 16: Adicionar o ponto $x_k + \Delta_k^e d$ na lista L_k ;
 - 17: **senão se** $v_2 > 0$ **então**
 - 18: Adicionar a direção d e o valor $t \leftarrow 0$ na lista V_k^+ ;
 - 19: **fim se**
 - 20: **fim para**
 - 21: **para todo** $d \in M_k^+$ **faça**
 - 22: Avaliar $f(x_k + \Delta_k^e d)$ e armazenar em f_2 ;
 - 23: Calcular v_1 , v_2 e t ;
 - 24: **se** $t < 1$ **então**
 - 25: Armazenar na variável lógica **resposta** o resultado do **Algoritmo de Análise de Possibilidade de Novo Vale** dados ponto base x_k , direção d , Passo Interno Δ_k e Passo Externo Δ_k^e ;
 - 26: **se resposta = verdadeira** **então**
 - 27: Adicionar $x_k + 2\Delta_k^e d$ na lista L_k ;
 - 28: **fim se**
 - 29: **senão**
 - 30: Adicionar a direção d e o valor t na lista V_k^+ ;
 - 31: **fim se**
 - 32: **fim para**
-

Compararemos $f(x_r)$ com $f(x_l)$ e o menor valor determinará o próximo iterando x_{k+1} . Isto é, se $f(x_r) < f(x_l)$ então $x_{k+1} = x_r$, senão $x_{k+1} = x_l$

O pseudo-código do Algoritmo de Decisão de Próximo Iterando é apresentado no Algoritmo 10.

Algoritmo 10 Algoritmo de decisão de próximo iterando

Entrada: A lista L_k ;

Saída: O ponto \tilde{x} ;

1: Determinar $x_l = \arg \min_{\{x \in L_k\}} f(x)$ e $x_h = \arg \max_{\{x \in L_k\}} f(x)$;

2: $s \leftarrow$ Média das distâncias entre os pontos $x \in L_k$;

3: $x_c = \sum_{x \in L_k} \left(\frac{x}{\#L_k} \right) - \frac{x_h}{\#L_k}$;

4: **se** $f(x_c) < f(x_l)$ **então**

5: $\tilde{x} \leftarrow x_c$;

6: **senão**

7: $x_r = (1 + l)x_c - lx_h$;

8: **se** $f(x_r) < f(x_l)$ **então**

9: $\tilde{x} \leftarrow x_r$;

10: **senão**

11: $\tilde{x} \leftarrow x_l$;

12: **fim se**

13: **fim se**

O critério de decisão tende a ser uma “perturbação” do resultado obtido. Em uma região de regularidade x_c pode ser uma melhor aproximação de um mínimo local. Por outro lado, o resultado da Análise de Mudança de Vale pode mostrar uma nova tendência pra o mínimo.

A justificativa para esta decisão é baseada em se tentar encontrar mais rapidamente mínimos locais onde o método possa colocar sua malha sobre uma região convexa.

5.7 Método MONCAS

Apresentamos o Método MONCAS na forma de um fluxograma exibido na Figura 5.14 e um pseudo-código exibido como o Algoritmo 11.

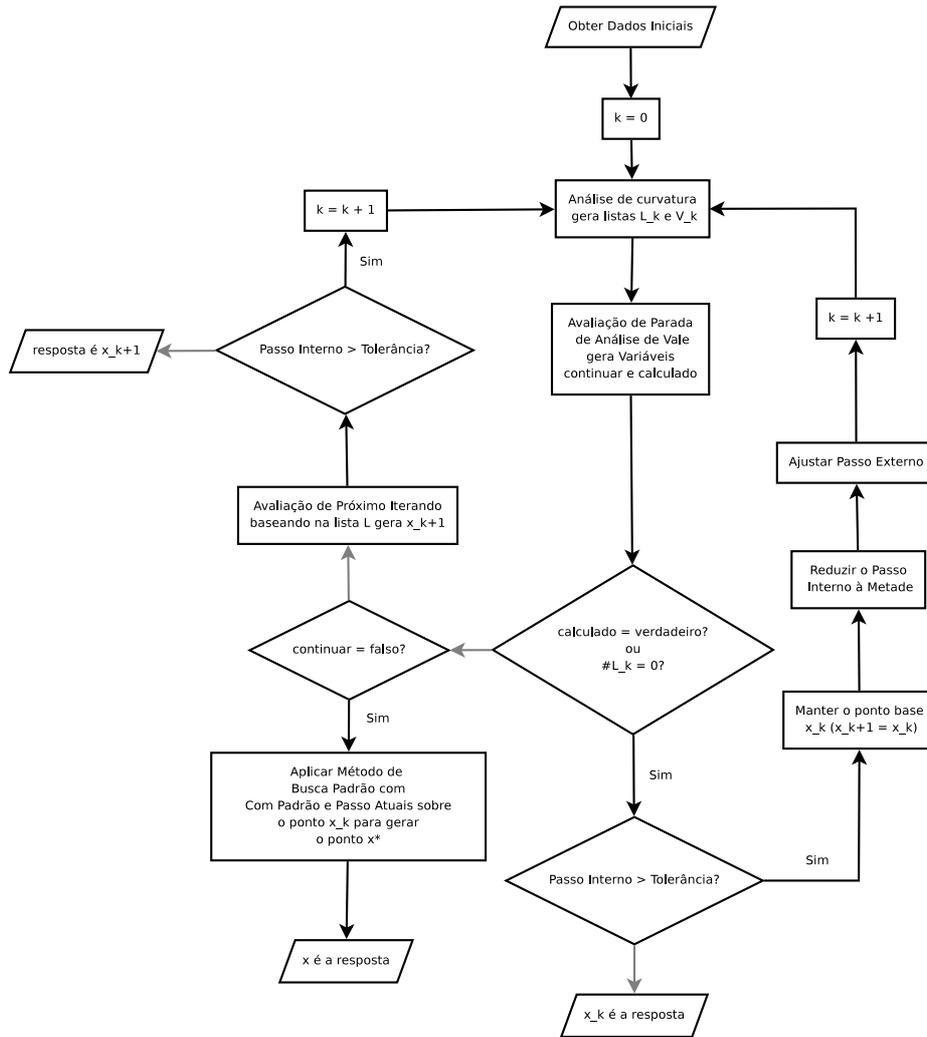


Figura 5.14: Fluxograma do Algoritmo MONCAS. Setas em cinza são negativas nos blocos de decisão

Algoritmo 11 Algoritmo MONCAS

Entrada: O ponto inicial x_0 , um padrão inicial P_0 , um passo inicial Δ_0 , um parâmetro de tolerância ϵ e um limite máximo de iterações k_{max} ;

Saída: O ponto de ótimo, aproximado, \bar{x} ;

- 1: $k \leftarrow 0$;
 - 2: **acesso** \leftarrow verdadeiro, **calculado** \leftarrow falso;
 - 3: **enquanto** $\Delta_k > \epsilon$ e $k < k_{max}$ **faça**
 - 4: Obter as listas L_k e V_k^+ do **Algoritmo de Análise de Mudança de Curvatura** dados ponto base x_k , padrão P_k , passo interno Δ_k e Passo Externo $\Delta_k^e \gg \Delta_k$;
 - 5: Guardar em **acesso** e **calculado**, respectivamente, o resultado do **Algoritmo de Avaliação de Parada de Análise de Vale** dados a lista V_k^+ e o padrão P_k ;
 - 6: **se** **calculado** = verdadeiro ou $\#L_k = 0$ **então**
 - 7: $x_{k+1} \leftarrow x_k$;
 - 8: $\Delta_{k+1} = \frac{\Delta_k}{2}$;
 - 9: Ajustar Δ_k^e ;
 - 10: **senão**
 - 11: Obtenha x_{k+1} do resultado do **Algoritmo de Decisão de Próximo Iterando** dada a lista L_k ;
 - 12: **fim se**
 - 13: **se** **acesso** = falso **então**
 - 14: Sair do **enquanto**;
 - 15: **fim se**
 - 16: $k \leftarrow k + 1$;
 - 17: **fim enquanto**
 - 18: **se** **acesso** = falso **então**
 - 19: Obter \bar{x} do resultado o **Algoritmo de Busca Padrão** dados o ponto x_k , a direção P_k e o passo Δ_k ;
 - 20: **senão**
 - 21: $\bar{x} \leftarrow x_{k+1}$;
 - 22: **fim se**
-

5.8 Passo Externo

Em nossa exposição, por várias vezes fizemos uso de Δ_k^e e mencionamos que ele deveria ser $\Delta_k^e \gg \Delta_k$. Contudo não foi realizado nenhum aprofundamento na abordagem deste parâmetro fundamental para a análise externa.

Há algumas considerações a respeito de Δ_k^e que devemos fazer. A primeira é que ele também deve se alterar à medida que Δ_k é alterado, devido à relação próxima entre eles. A segunda é que Δ_k^e deve absorver características locais da função-objeto.

A terceira consideração sugere que uma certa memória seja disponibilizada ao algoritmo básico. O estudo de como isso deve ser feito, para que haja uma maior significância analítica, é um trabalho a ser desenvolvido futuramente.

Sobre a primeira consideração, devemos lembrar que sempre desejamos que a análise externa se estenda a outras regiões, e assim queremos gerar uma sequência com a característica de $\Delta_k^e \leq \Delta_{k+1}^e$. Isso é uma mudança de tamanho na malha externa, o que se enquadra em redimensionamento da malha. Contudo, isto não é absoluto. Uma vez que uma característica puramente local de uma função arbitrária, pode alterar a relação entre os itens de uma sequência, podemos encontrar situações onde uma relação $\Delta_k^e \geq \Delta_{k+1}^e$ pode nos favorecer.

Sugerimos que Δ_k^e seja avaliada como uma função de Δ_k ,

$$\Delta_k^e = \Delta^e(\Delta_k) \tag{5.1}$$

a ser estipulada de forma conveniente para a análise da função objetivo.

Um Passo externo mais eficiente deve levar em conta outras características da função e com isso tomar uma complexidade maior.

Para este primeiro estudo sobre esta abordagem com Passo Externo, preferimos manter Δ^e como uma função exclusivamente de Δ . Ressaltamos que Δ^e é uma função. Isto ressalta um maior contraste do Método MONCAS com os outros, cujos parâmetros são apenas valores reais ou vetores euclidianos.

Inicialmente estaremos interessados em estudar o comportamento do método para funções Δ^e com presente certo comportamento ao serem comparadas com o parâmetro Δ . Assim, formularemos uma função $g(\Delta) = \frac{\Delta^e}{\Delta}$. Por definição de $\Delta^e \gg \Delta$, e assim $g(\Delta) \gg 1$.

Utilizaremos Passos Externos tais que $g'(\Delta) > 0$, $g'(\Delta) < 0$ e $g'(\Delta) = 0$. Faremos isto com o intuito de analisar o comportamento do método em face de passos externos com taxa de crescimento e decrescimento relativos ao Passo Interno bem variados.

Com isso esperamos mostrar que esta diferença no comportamento do Passo Externo tem influência significativa e se torna digna de estudos posteriores.

6 SIMULAÇÕES, METODOLOGIA DE COMPARAÇÃO E ANÁLISE

Neste capítulo apresentaremos as experiências feitas com programas computacionais. Isso é parte da metodologia de teste aplicada ao MONCAS para mostrar sua utilidade para problemas onde a função objetivo não é convexa em todo seu domínio.

Selecionaremos alguns problemas convexos onde deve ficar claro que o método tem rendimento pior que os mais clássicos em termos do número de avaliações de função devido à forma como foi concebido.

Em seguida serão mostrados alguns problemas não convexos e compararemos a resolução dos mesmos utilizando os métodos apresentados nos capítulos anteriores, onde tentaremos averiguar a robustez e eficiência dos métodos, além da convergência.

Alguns detalhes técnicos serão apresentados para a formulação dos resultados, tais como, linguagem de programação utilizada, compilador e configuração da máquina onde foi executado o programa. Tais detalhes têm por finalidade dar base de comparação do tempo necessário à obtenção da resposta.

6.1 Métodos numéricos e parâmetros iniciais

Aqui deixaremos claro quais métodos foram empregados e a razão para tal. Com isso queremos que o leitor fique atento às peculiaridades de cada um deles antes de fazer alguma comparação mais minuciosa.

Os métodos escolhidos são:

- Método Nelder-Mead (Simplético);
- Método de Busca Padrão (Versão mais simples, com Padrão fixo);
- MONCAS.

Como foi exposto acima lembramos que o Método Nelder-Mead difere dos outros dois em estrutura por ser um método simplético. Com isso nossos dados iniciais se constituem de um politopo inicial somente, não havendo geração de qualquer estrutura de malha.

O Método de Busca Padrão sofreu várias modificações ao passar dos anos, cada uma delas com um objetivo definido. Aqui decidimos utilizar sua versão mais simples, onde tomamos um padrão fixo, o qual permanece constante com a variação do índice k , contador de iterações.

Finalmente temos algumas considerações a fazer quanto ao MONCAS. Como mencionado no capítulo anterior, o parâmetro Δ_k^e é considerado como uma função de Δ_k dentro das iterações do método.

Este conceito foi um tanto vago, pois isso é mérito de discussão a respeito de como

o comportamento deste parâmetro deve ser modelado para um problema específico. Contudo é interessante aumentá-lo toda vez que Δ_k for diminuído. Dependendo do parâmetro de tolerância ϵ , Δ_k pode ser tornar extremamente pequeno, mas isso significa que Δ_k^e deva se tornar extremamente grande?

É claro que o termo “extremamente pequeno” e por consequência “extremamente grande” são dependentes do contexto ou escala do problema. Mas Δ_k^e muito longe de Δ_k torna as análises executadas pelo método muito imprecisas. Desta forma temos como intenção limitar o tamanho do Passo Externo.

Iremos simular o comportamento do MONCAS sob duas perspectivas para o Passo Externo. Tomaremos as funções

$$\Delta_k^e = \Delta_k^e(\Delta_k) = \alpha \sqrt{\Delta_k} \quad (6.1)$$

$$\Delta_k^e = \Delta_k^e(\Delta_k) = \alpha e^{-\Delta_k} \quad (6.2)$$

e

$$\Delta_k^e = \Delta_k^e(\Delta_k) = \alpha \Delta_k. \quad (6.3)$$

Tomaremos $\alpha = 5.5$. A razão é que ele seja significativamente maior que Δ_k . Esperamos que o comportamento da função em (6.1) tenda a uma análise dentro de um vale, enquanto que a função em (6.2) tenderá a um valor fixo.

Os critérios de escolha destas funções $\Delta_k^e(\Delta_k)$ são dados por conta das derivadas da função $g(\Delta)$ definida em 5.1. Notemos:

$$\frac{d}{d\Delta_k} \left[\frac{\alpha \sqrt{\Delta_k}}{\Delta_k} \right] = \frac{-\alpha}{2(\Delta_k)^{2,5}} (1 + 2(\Delta_k)^{1,5}) \quad (6.4)$$

$$\frac{d}{d\Delta_k} \left[\frac{\alpha e^{-\Delta_k}}{\Delta_k} \right] = \frac{-\alpha e^{-\Delta_k}}{(\Delta_k)^2} (\Delta_k - 1) \quad (6.5)$$

$$\frac{d}{d\Delta_k} \left[\frac{\alpha \Delta_k}{\Delta_k} \right] = 0, \quad (6.6)$$

Método	Descrição
S_1	Método de Busca Padrão
S_2	Método Nelder-Mead
S_3	Método MONCAS com Passo Externo $\Delta_k^e = \alpha e^{-\Delta_k}$
S_4	Método MONCAS com Passo Externo $\Delta_k^e = \alpha \sqrt{\Delta_k}$
S_5	Método MONCAS com Passo Externo $\Delta_k^e = \alpha \Delta_k$

Tabela 6.1: Solvers Aplicados na Solução dos Problemas

onde na Equação 6.5 assumimos um valor $\Delta_k < 1$. Isso nos dá três maneiras de encarar esse Passo Externo, decrescente, crescente e constante em relação ao Passo Interno.

É também de nosso interesse observar e analisar o comportamento do Método MONCAS sob essas condições de Passo Externo.

Indexaremos os algoritmos (*Solvers*) como na Tabela 6.1

6.1.1 Padrão de busca

Como mencionado, vamos manter o Método de Busca Padrão com um padrão constante. Apesar de o MONCAS permitir esse tipo de modificação, vamos manter esse padrão fixo para ele também.

Tomaremos o Padrão como sendo um conjunto minimal para geração do espaço E , todos com módulo unitário, mas tal que o primeiro vetor ficará sobreposto ao eixo

Problema	Descrição
P_0	Parabolóide
P_1	Extended Powell singular function
P_2	Griewank function
P_3	Ackley's function
P_4	Generalized Swefel function
P_5	Rotated Hyper Ellipsoid function
P_6	Rastringin function
P_7	Schwefel's function
P_8	Trigonometric function

Tabela 6.2: Lista de Problemas para Avaliação de Métodos

x_1 , na parte positiva. O algoritmo usado foi desenvolvido fazendo uso da Fórmula (2.7).

6.2 Problemas

Foram selecionados 8 problemas tirados de 2 trabalhos. O primeiro trabalho é o de Jorge J. Moré *et al* (MORÉ; GARBOW; HILLSTROM, 1981). Os demais problemas foram retirados de (OMRAN; MAHDAVI, 2008)

As funções ou problemas são indexados por P_i e listados na Tabela 6.2.

A seguir apresentamos as descrições dos problemas.

6.2.1 Parabolóide

Esta função é clássica em matemática e foi mantida simplesmente como uma comparação entre os problemas e métodos.

$$f(x) = \sum_{i=1}^n ((x_i)^2).$$

6.2.2 Extended Powell singular function

Obtida de (MORÉ; GARBOW; HILLSTROM, 1981), a função permite trabalhar em um espaço \mathbb{R}^n com n múltiplo de 4.

$$\begin{aligned} m &= \frac{n}{4}, i = 1, \dots, m \\ f_{4i-3}(x) &= x_{4i-3} - 10x_{4i-2} \\ f_{4i-2}(x) &= \sqrt{5}(x_{4i-1} - x_{4i}) \\ f_{4i-1}(x) &= (x_{4i-2} - 2x_{4i-1})^2 \\ f_{4i}(x) &= \sqrt{10}(x_{4i-3} - x_{4i}), \\ f(x) &= \sum_{i=1}^m ((f_i(x))^2). \end{aligned}$$

6.2.3 Griewank function

Obtida de (OMRAN; MAHDAVI, 2008), é definida para uma dimensão n qualquer.

$$f(x) = \frac{1}{4000} \sum_{i=1}^n ((x_i)^2) - \prod_{i=1}^n \left(\cos \left(\frac{x_i}{\sqrt{i}} \right) \right) + 1.$$

Um esboço desta função é exibido na Figura 6.2.3.

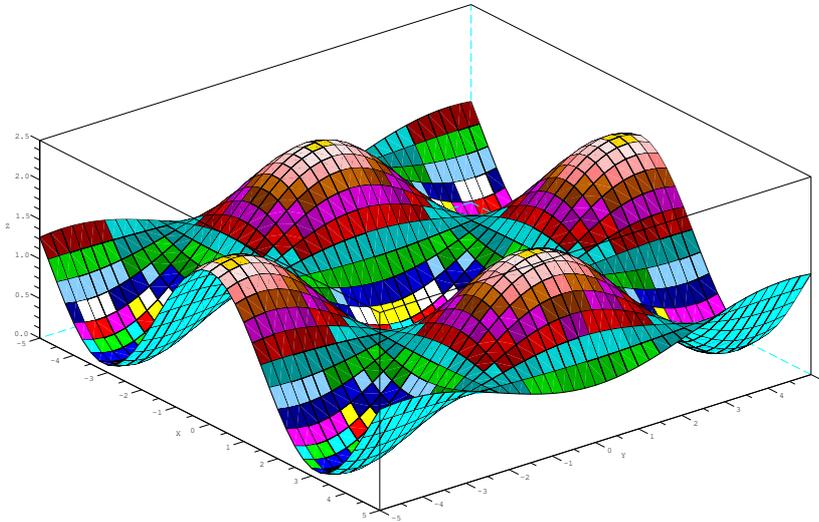


Figura 6.1: Esboço da função Griewank para $n = 2$

6.2.4 Ackley's function

Obtida de (OMRAN; MAHDAVI, 2008), é definida para uma dimensão n qualquer.

$$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^n ((x_i)^2)} \right) - \exp \left(\frac{1}{30} \sum_{i=1}^n (\cos(2\pi x_i)) \right) + 20 + \exp(1)$$

Um esboço desta função é exibido na Figura 6.2.4.

6.2.5 Generalized Swefel function

Obtida de (OMRAN; MAHDAVI, 2008), é definida para uma dimensão n qualquer.

$$f(x) = - \sum_{i=1}^n \left(x_i \operatorname{sen} \left(\sqrt{|x_i|} \right) \right).$$

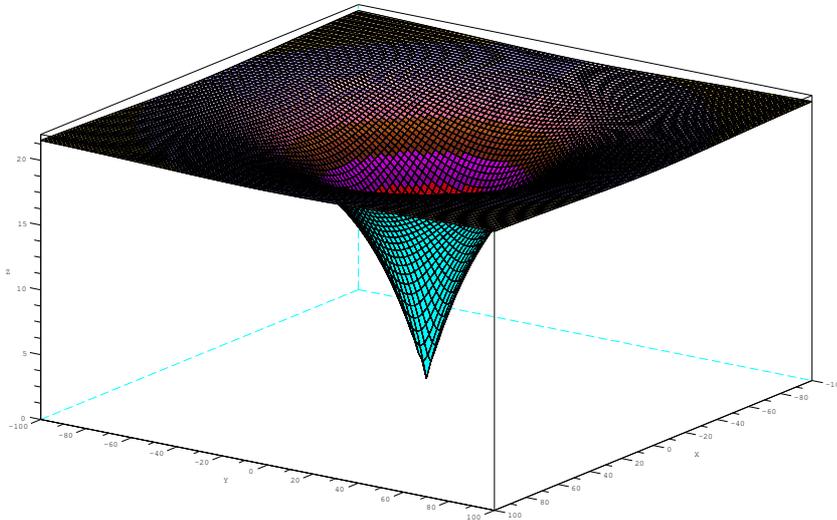


Figura 6.2: Esboço da função de Ackley para $n = 2$

Um esboço desta função é exibido na Figura 6.2.5.

6.2.6 Rotated Hyper-Ellipsoid function

Obtida de (OMRAN; MAHDAVI, 2008), é definida para uma dimensão n qualquer.

$$f(x) = \sum_{i=1}^n \left(\left(\sum_{j=1}^i (x_j) \right)^2 \right).$$

Um esboço desta função é exibido na Figura 6.2.6.

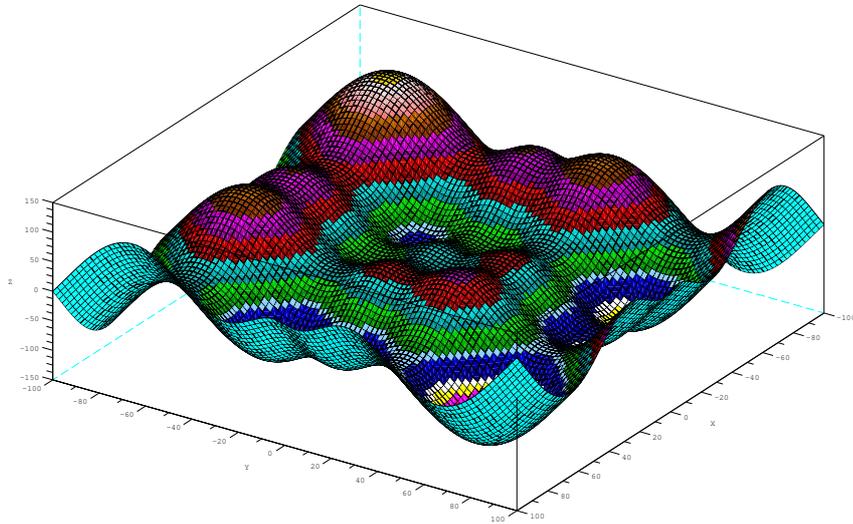


Figura 6.3: Esboço da função Swefel Generalizada para $n = 2$

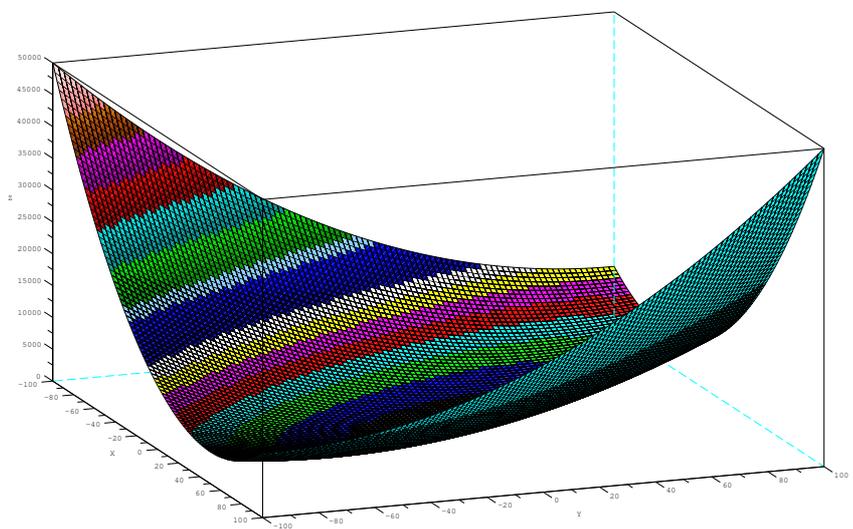


Figura 6.4: Esboço da função Hyper-Elipsóide Rotacionado para $n = 2$

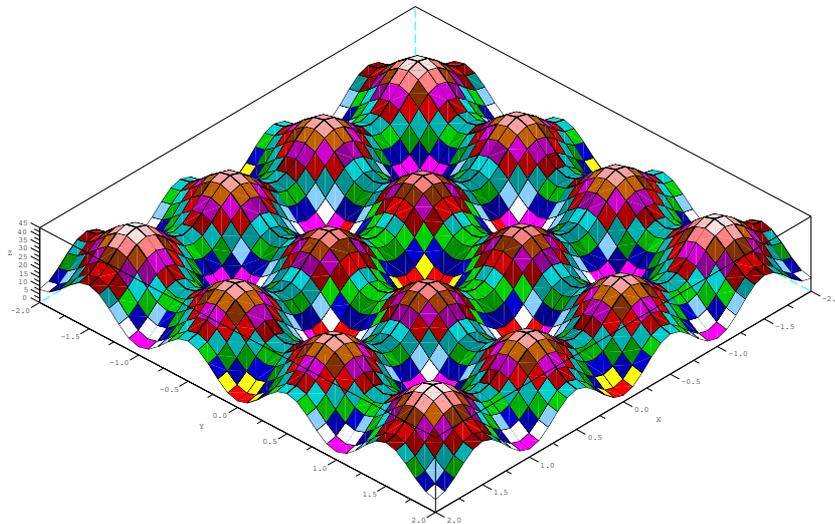


Figura 6.5: Esboço da função Rastrigin para $n = 2$

6.2.7 Rastrigin function

Obtida de (OMRAN; MAHDAVI, 2008), é definida para uma dimensão n qualquer.

$$f(x) = \sum_{i=1}^n ((x_i)^2 - 10\cos(2\pi x_i) + 10).$$

Um esboço desta função é exibido na Figura 6.2.7.

6.2.8 Schwefel's function

Obtida de (OMRAN; MAHDAVI, 2008), é definida para uma dimensão n qualquer.

$$f(x) = \sum_{i=1}^n (|x_i|) + \prod_{i=1}^n (|x_i|).$$

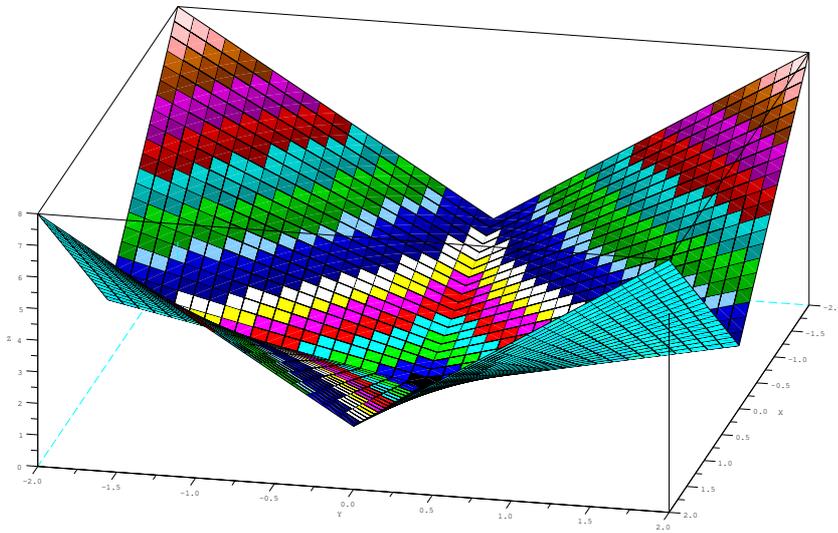


Figura 6.6: Esboço da função de Schwefel para $n = 2$

Um esboço desta função é exibido na Figura 6.2.8.

6.2.9 Trigonometric function

Obtida de (MORÉ; GARBOW; HILLSTROM, 1981), é definida para uma dimensão n qualquer.

$$m = n, i = 1, \dots, m$$

$$f_i(x) = n - \sum_{j=1}^n (\cos(x_j)) + i(1 - \cos(x_i)) - \text{sen}(x_i),$$

$$f(x) = \sum_{i=1}^m ((f_i)^2).$$

Um esboço desta função é exibido na Figura 6.2.9.

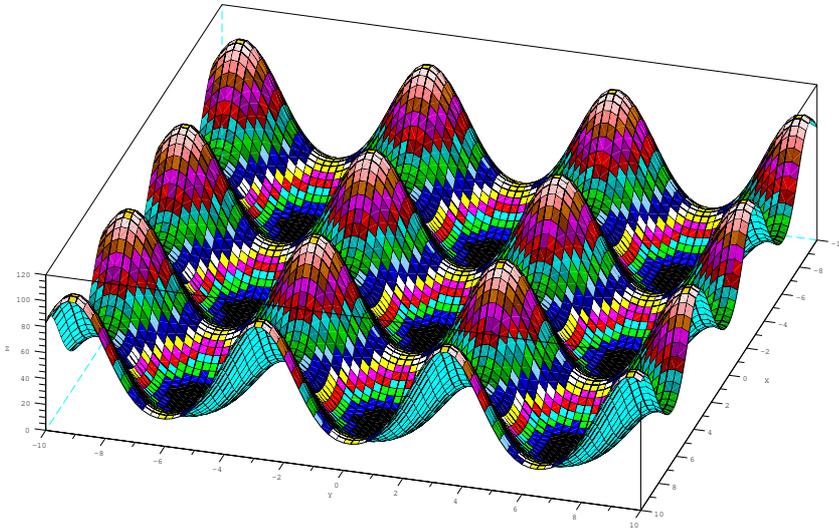


Figura 6.7: Esboço da função Trigonômica para $n = 2$

6.3 Dimensionamento dos problemas

Várias das funções utilizadas permitem uma variação da dimensão do espaço Euclidiano E no qual se trabalha para se encontrar a solução. Daí iremos variar este espaço para alguns valores de n , a fim que possa ser medido o desempenho dos métodos sob várias condições.

Contudo algumas funções apresentam restrições quanto ao valor da dimensão n . Assim apresentaremos a Tabela 6.3 onde estão relacionados os valores de n e os problemas aos quais eles serão aplicados.

Desta forma trabalhamos com duas amostras distintas. Em uma temos um total de 5 métodos, 9 problemas (8 de teste real e 1 como comparação) e 6 dimensões de espaço. Na outra temos os mesmos 5 métodos, 3 problemas e 1 dimensão. Em

n	Problemas
$n_1 = 4$	$P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$
$n_2 = 8$	$P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$
$n_3 = 12$	$P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$
$n_4 = 16$	$P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$
$n_5 = 20$	$P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$
$n_6 = 24$	$P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$

Tabela 6.3: Variação das dimensões dos problemas usados na análise

termos de espaços cartesianos, isto significa que estamos trabalhando com dados do tipo (S_i, P_j, n_k)

Desconsiderando o problema P_0 , temos um total de 240 linhas de dados para analisar. Cada linha contendo 4 valores numéricos a serem discriminados na Seção 6.5.

6.4 Perfil de desempenho

Nesta seção apresentaremos uma breve descrição de como proceder à geração do perfil de desempenho.

Perfil de Desempenho é uma metodologia de análise e comparação de algoritmos que se propõem a resolver um conjunto de problemas. Esta metodologia foi proposta por Elizabeth E. Dolan e Jorge J. Moré em seu trabalho (DOLAN; MORÉ, 2002). E vem sendo utilizada como uma ferramenta muito útil para se avaliar o comportamento dos algoritmos.

A metodologia determina um conjunto de problemas a se trabalhar denominado de

\mathcal{P} e um conjunto de métodos (*Solvers*) que se propõe a resolvê-los denominado de \mathcal{S} . Em geral são tomados conjuntos finitos para a análise, apesar do conjunto de problemas poder ser, teoricamente, infinito. A cardinalidade dos conjuntos, considerando ambos finitos, é dada por $n_p = \#\mathcal{P}$ e $n_s = \#\mathcal{S}$.

Neste contexto é definida uma função de **medida computacional**. Esta função é uma medida de duração do processo que leva, ou tenta chegar, à solução do problema. A função é definida em termos do espaço cartesiano $\mathcal{P} \times \mathcal{S}$, e geralmente denotada por $t(p, s) = t_{p,s}$, $p \in \mathcal{P}$, $s \in \mathcal{S}$. Esta função pode ser medida em termos de tempo, número de iterações, número de avaliações de função, etc.

Definida $t_{p,s}$, os autores formulam uma medida que denominam **razão de desempenho** denotada por $r_{p,s}$, é definida por:

$$r_{p,s} = \frac{t_{p,s}}{\min \{t_{p,b} \mid b \in \mathcal{S}\}}.$$

A partir deste ponto, vamos montar uma função de corte sobre a razão de desempenho r , e como consequência, a medida computacional t . Criaremos uma função $v_s = v_s(\tau)$ dada por

$$v_s(\tau) = \#\{p \in \mathcal{P} \mid r_{p,s} \leq \tau\}.$$

Esta função é auxiliar e não consta no artigo original, contudo ela nos ajuda a formular melhor a probabilidade de um método s resolver o problema p com uma razão de desempenho máxima de τ . Isto é,

$$\rho_s(\tau) = \frac{v_s(\tau)}{n_p}.$$

O artigo (DOLAN; MORÉ, 2002) segue demonstrando propriedades e fatos sobre a métrica da função $\rho_s(\tau)$. Estes tópicos serão omitidos deste texto, pois utilizaremos o Perfil de Desempenho apenas como uma ferramenta de análise.

6.5 Referencial técnico e simulações numéricas

Nesta seção apresentaremos os resultados brutos das simulações e apresentaremos alguns detalhes técnicos como linguagem ou configuração do computador no qual a simulação foi executada.

Os métodos foram programados em linguagem de programação C, utilizando compilador gcc e executados em sistema operacional Linux.

Inicialmente os métodos foram testados utilizando a função $f(x) = \sum_{i=1}^n ((x_i)^2)$, isso foi feito para que os programas pudessem ser depurados, e pudéssemos ter uma noção do comportamento dos métodos em uma função extremamente suave.

Alguns pontos relativos a idiosincrasias do ambiente computacional devem ser mencionados, juntamente com algumas convenções adotadas, para que os dados brutos tenham a correta interpretação. Devido a uma limitação da linguagem, apesar de sua velocidade, não foi possível computar o tempo de execução de alguns métodos. O sistema utilizado para medir o tempo tinha uma marcação baseada em uma medida de *clock* da máquina. Esta medida era registada **1 000 000** de vezes por segundo. Métodos que tiveram um tempo de execução inferior a 1×10^{-6} segundos obtiveram uma medida de clock igual a **0** (zero). Devido a esse pequeno problema, resolvemos marcar o tempo de execução destes métodos como sendo 9×10^{-7} segundos. Ainda assim, nesta seção eles serão apresentados como **0**.

Como convenção da nossa programação, convencionamos que todos os métodos executam pelo menos uma iteração. Contudo apresentaremos valores de **0** (zero) iterações para alguns métodos. Isto significa que este método ativou a salvaguarda que limitava o número de iterações, isto é, este método falhou ao tentar resolver o

Critério	Valor	Descrição
ϵ	1×10^{-8}	Erro (aproximado) admissível à resposta do problema
k_{max}	1×10^5	Limite máximo de iterações aceitável

Tabela 6.4: Dados sobre o limite de iterações e critério de parada

problema em um número limitado de iterações.

O valor da tolerância do Critério de Parada foi mantido o mesmo para todos os métodos, apesar de sua significância ser um pouco diferente. Isso não afeta significativamente o processo pois estas significâncias dos critérios de parada são razoavelmente semelhantes, ou estão em uma região de confiança aceitável. Os dados relativos a esses critérios estão dispostos na Tabela 6.4.

Todos os métodos foram testados a partir do mesmo ponto inicial $x_i = 2.1, i = 1, \dots, n$. O simplex do Método Nelder-Mead foi construído usando as fórmulas da Tabela 2.1, acrescidas deste ponto inicial, utilizando um lado $l = 1.0$. Os métodos que fazem uso de um passo inicial foram executados com um valor $\Delta_0 = 0.8$.

Uma última ressalva é quanto ao valor que foi obtido por certas funções. Apesar de termos feito uso de aritmética de ponto flutuante com precisão dupla (uma variável do tipo **double**), os algarismos significativos para análise extrapolaram a nossa necessidade de visualização. Truncamos estes valores de forma que ainda fossem significativos.

Apresentamos nas Tabelas 6.6, 6.7, 6.8, 6.9, 6.10, 6.11 e 6.12 a lista destes valores. As colunas destas tabelas estão discriminadas na Tabela 6.5.

n_k	Dimensão do Espaço
P_j	Problema analisado
S_i	Método utilizado
n_{it}	Número de iterações utilizadas na simulação
t	Tempo de execução gasto na simulação
n_{av}	Número de avaliações de função feitas no método
$f_{k,j,i}$	Valor da função objetivo atingido pelo método

Tabela 6.5: Discriminação das Colunas dos resultados das simulações numéricas.

6.6 Desenvolvimento e análise sobre perfis de desempenho

Procedemos, agora, ao desenvolvimento e análise de Perfis de Desempenho. Relembrando a Seção 6.4, o perfil de desempenho é formulado sobre uma medida computacional $t_{p,s}$. Nossos ensaios nos permitiram gerar 3 (três) quantidades que podem ser consideradas medidas:

- Número de Iterações;
- Tempo de Cálculo;
- Número de Avaliações de Função.

A partir de cada uma delas, construiremos um gráfico de perfil que nos permitirá avaliar, visualmente, como se comportam os métodos, um em relação ao outro.

n_k	P_j	S_i	n_{it}	t	n_{av}	$f_{k,j,i}$
n_1	P_0	S_1	85	0.000000	510	0.0000000000000000
n_1	P_0	S_2	134	0.000000	2220	0.00000000965924
n_1	P_0	S_3	29	0.000000	196	0.0000000000000000
n_1	P_0	S_4	86	0.000000	544	0.0000000000000000
n_1	P_0	S_5	82	0.000000	538	0.0000000000000000
n_1	P_1	S_1	2505	0.000000	15030	0.000000000000025
n_1	P_1	S_2	130	0.000000	2180	0.00000001027688
n_1	P_1	S_3	2317	0.000000	13924	0.000000000000034
n_1	P_1	S_4	1111	0.000000	6693	0.000000000000019
n_1	P_1	S_5	1859	0.000000	11181	0.000000000000033
n_1	P_2	S_1	114	0.000000	684	0.00739604033412
n_1	P_2	S_2	111	0.000000	1840	0.00000000680171
n_1	P_2	S_3	119	0.000000	754	0.000000000000000
n_1	P_2	S_4	87	0.000000	949	0.00986467206101
n_1	P_2	S_5	0	0.960000	1423078	0.54776783133465
n_1	P_3	S_1	85	0.000000	510	1.57565102140100
n_1	P_3	S_2	233	0.000000	3867	1.57565105454815
n_1	P_3	S_3	28	0.000000	444	1.57565101666332
n_1	P_3	S_4	28	0.000000	328	1.57565101666332
n_1	P_3	S_5	28	0.000000	322	1.57565101666332
n_1	P_4	S_1	105	0.000000	630	-15.78120650113730
n_1	P_4	S_2	0	0.580000	1800000	-10.15297670819060
n_1	P_4	S_3	99	0.010000	1195	-15.78120650113730
n_1	P_4	S_4	86	0.000000	667	-15.78120650113730
n_1	P_4	S_5	83	0.000000	862	-15.78120650113730
n_1	P_5	S_1	637	0.000000	3822	0.000000000000000
n_1	P_5	S_2	118	0.000000	1984	0.00000001272849
n_1	P_5	S_3	28	0.000000	184	0.000000000000000
n_1	P_5	S_4	28	0.000000	184	0.000000000000000
n_1	P_5	S_5	28	0.000000	309	0.000000000000000
n_1	P_6	S_1	74	0.000000	444	15.91932476312179
n_1	P_6	S_2	0	0.600000	1799976	17.42081704883353
n_1	P_6	S_3	0	0.580000	1599998	0.000000000000000
n_1	P_6	S_4	79	0.000000	915	15.91932476312180
n_1	P_6	S_5	29	0.000000	232	0.000000000000000
n_1	P_7	S_1	4017	0.000000	24102	0.00000270271838
n_1	P_7	S_2	255	0.000000	4202	0.00000002033071
n_1	P_7	S_3	29	0.000000	227	0.000000000000000
n_1	P_7	S_4	86	0.000000	554	0.00000001564760
n_1	P_7	S_5	77	0.000000	541	0.00000002922678

Tabela 6.6: Resultados das simulações numéricas.

n_k	P_j	S_i	n_{it}	t	n_{av}	$f_{k,j,i}$
n_1	P_8	S_1	251	0.000000	1506	0.00030282411489
n_1	P_8	S_2	119	0.000000	1991	0.00000000870715
n_1	P_8	S_3	774	0.010000	4754	0.00030282411489
n_1	P_8	S_4	218	0.000000	2695	0.00000000000000
n_1	P_8	S_5	73	0.000000	475	0.00000000000000
n_2	P_0	S_1	177	0.000000	1770	0.00000000000000
n_2	P_0	S_2	235	0.000000	6302	0.00000000946284
n_2	P_0	S_3	167	0.000000	1708	0.00000000000000
n_2	P_0	S_4	138	0.000000	1428	0.00000000000000
n_2	P_0	S_5	144	0.000000	1497	0.00000000000000
n_2	P_1	S_1	6576	0.010000	65760	0.00000000000058
n_2	P_1	S_2	275	0.010000	7358	0.00000000750687
n_2	P_1	S_3	4375	0.010000	43831	0.00000000000083
n_2	P_1	S_4	5740	0.010000	57512	0.00000000000046
n_2	P_1	S_5	5419	0.020000	54257	0.00000000000065
n_2	P_2	S_1	661	0.010000	6610	0.00739604033412
n_2	P_2	S_2	218	0.000000	5832	0.00000000684851
n_2	P_2	S_3	29	0.000000	328	0.00000000000000
n_2	P_2	S_4	208	0.010000	2180	0.00000000000000
n_2	P_2	S_5	0	2.310000	2150011	0.00814959859774
n_2	P_3	S_1	172	0.000000	1720	1.41267666990463
n_2	P_3	S_2	444	0.010000	11922	1.41267668710179
n_2	P_3	S_3	29	0.000000	796	1.41267665639409
n_2	P_3	S_4	131	0.000000	1496	1.41267666450825
n_2	P_3	S_5	79	0.000000	1017	1.41267666518264
n_2	P_4	S_1	211	0.000000	2110	-31.56241300227460
n_2	P_4	S_2	0	1.900000	3000000	-18.98110468163053
n_2	P_4	S_3	160	0.000000	2734	-31.56241300227457
n_2	P_4	S_4	129	0.000000	1927	-31.56241300227460
n_2	P_4	S_5	148	0.000000	2150	-31.56241300227460
n_2	P_5	S_1	4057	0.020000	40570	0.00000000000005
n_2	P_5	S_2	209	0.000000	5781	0.00000001112119
n_2	P_5	S_3	3270	0.020000	32738	0.00000000000006
n_2	P_5	S_4	3038	0.010000	30418	0.00000000000004
n_2	P_5	S_5	3041	0.020000	30466	0.00000000000005
n_2	P_6	S_1	142	0.000000	1420	36.81341957745945
n_2	P_6	S_2	0	1.770000	2999985	36.56266083702262
n_2	P_6	S_3	0	2.000000	2457166	27.63149410663152
n_2	P_6	S_4	212	0.000000	3526	0.00000000000002
n_2	P_6	S_5	0	1.980000	2733329	40.61996739917827

Tabela 6.7: Resultados das simulações numéricas.

n_k	P_j	S_i	n_{it}	t	n_{av}	$f_{k,j,i}$
n_2	P_7	S_1	239	0.000000	2390	0.18050520734063
n_2	P_7	S_2	492	0.000000	13140	0.00000001966018
n_2	P_7	S_3	197	0.000000	2071	0.01654789282096
n_2	P_7	S_4	144	0.000000	1510	0.00000164660182
n_2	P_7	S_5	121	0.000000	1330	0.00104040284745
n_2	P_8	S_1	513	0.020000	5130	0.00000000000000
n_2	P_8	S_2	0	9.410000	2999603	0.00493240421366
n_2	P_8	S_3	1912	0.150000	39663	0.00013564721929
n_2	P_8	S_4	1168	0.090000	24224	0.00013564721929
n_2	P_8	S_5	132	0.000000	1378	0.00000000000000
n_3	P_0	S_1	308	0.000000	4312	0.00000000000000
n_3	P_0	S_2	390	0.000000	14222	0.00000000696077
n_3	P_0	S_3	169	0.000000	2434	0.00000000000000
n_3	P_0	S_4	282	0.000000	4016	0.00000000000000
n_3	P_0	S_5	236	0.000000	3398	0.00000000000000
n_3	P_1	S_1	15315	0.070000	214410	0.00000000000125
n_3	P_1	S_2	423	0.010000	15595	0.00000000742352
n_3	P_1	S_3	11682	0.060000	163687	0.00000000000108
n_3	P_1	S_4	11261	0.070000	157796	0.00000000000122
n_3	P_1	S_5	14313	0.080000	200523	0.00000000000109
n_3	P_2	S_1	1250	0.020000	17500	0.00739604033413
n_3	P_2	S_2	337	0.020000	12388	0.00000000994335
n_3	P_2	S_3	1052	0.020000	14782	0.00000000000000
n_3	P_2	S_4	0	4.660000	2949999	0.25786110710847
n_3	P_2	S_5	0	4.650000	2949998	0.21004538569864
n_3	P_3	S_1	294	0.000000	4116	1.22645713806817
n_3	P_3	S_2	709	0.030000	26028	1.22645714217046
n_3	P_3	S_3	166	0.000000	2511	1.22645713747460
n_3	P_3	S_4	193	0.000000	2929	1.22645714174250
n_3	P_3	S_5	188	0.010000	2818	1.22645714707874
n_3	P_4	S_1	397	0.000000	5558	-47.34361950341179
n_3	P_4	S_2	0	3.780000	4200000	-27.62010831564202
n_3	P_4	S_3	265	0.010000	7556	-47.34361950341189
n_3	P_4	S_4	188	0.010000	5341	-47.34361950341189
n_3	P_4	S_5	196	0.000000	5573	-47.34361950341190
n_3	P_5	S_1	15132	0.170000	211848	0.00000000000026
n_3	P_5	S_2	363	0.010000	13748	0.00000000608385
n_3	P_5	S_3	10519	0.130000	147347	0.00000000000028
n_3	P_5	S_4	11433	0.130000	160130	0.00000000000028
n_3	P_5	S_5	11031	0.140000	154515	0.00000000000025

Tabela 6.8: Resultados das simulações numéricas.

n_k	P_j	S_i	n_{it}	t	n_{av}	$f_{k,j,i}$
n_3	P_6	S_1	194	0.000000	2716	52.73274434058121
n_3	P_6	S_2	0	3.720000	4199963	51.44232884367537
n_3	P_6	S_3	291	0.000000	4667	0.000000000000014
n_3	P_6	S_4	443	0.010000	7799	0.000000000000004
n_3	P_6	S_5	36	0.000000	765	0.000000000000000
n_3	P_7	S_1	264	0.000000	3696	0.23482689823779
n_3	P_7	S_2	780	0.000000	28502	0.00000001918968
n_3	P_7	S_3	49	0.000000	1386	0.00000250149344
n_3	P_7	S_4	202	0.000000	2940	0.01440511710686
n_3	P_7	S_5	196	0.000000	2862	0.06748186463655
n_3	P_8	S_1	6359	0.570000	89026	0.00000406582526
n_3	P_8	S_2	0	26.310000	4199020	0.00186943302750
n_3	P_8	S_3	2708	0.260000	38138	0.00003027047184
n_3	P_8	S_4	534	0.040000	7607	0.00038281189486
n_3	P_8	S_5	674	0.140000	19497	0.000000000000000
n_4	P_0	S_1	308	0.000000	5544	0.000000000000000
n_4	P_0	S_2	555	0.010000	25628	0.00000000794979
n_4	P_0	S_3	296	0.000000	5416	0.000000000000000
n_4	P_0	S_4	223	0.000000	4102	0.000000000000000
n_4	P_0	S_5	281	0.010000	5180	0.000000000000000
n_4	P_1	S_1	23704	0.190000	426672	0.00000000000175
n_4	P_1	S_2	589	0.010000	27592	0.00000000771540
n_4	P_1	S_3	17120	0.150000	308340	0.00000000000172
n_4	P_1	S_4	17362	0.160000	312680	0.00000000000159
n_4	P_1	S_5	17450	0.160000	314206	0.00000000000198
n_4	P_2	S_1	2468	0.090000	44424	0.00739604033413
n_4	P_2	S_2	446	0.030000	21046	0.00000001052703
n_4	P_2	S_3	2240	0.080000	40390	0.000000000000000
n_4	P_2	S_4	299	0.010000	5530	0.000000000000000
n_4	P_2	S_5	2204	0.080000	39743	0.000000000000000
n_4	P_3	S_1	315	0.000000	5670	1.01367698197737
n_4	P_3	S_2	977	0.060000	45308	1.01367698353127
n_4	P_3	S_3	433	0.010000	7987	1.01367697902704
n_4	P_3	S_4	382	0.010000	7136	1.01367697264014
n_4	P_3	S_5	331	0.010000	6179	1.01367697287448
n_4	P_4	S_1	456	0.010000	8208	-63.12482600454901
n_4	P_4	S_2	0	6.350000	5400000	-36.17479161043264
n_4	P_4	S_3	584	0.030000	21486	-63.12482600454882
n_4	P_4	S_4	389	0.020000	14263	-63.12482600454909
n_4	P_4	S_5	491	0.030000	18037	-63.12482600454885

Tabela 6.9: Resultados das simulações numéricas.

n_k	P_j	S_i	n_{it}	t	n_{av}	$f_{k,j,i}$
n_4	P_5	S_1	33152	0.730000	596736	0.000000000000083
n_4	P_5	S_2	503	0.030000	23872	0.00000000685438
n_4	P_5	S_3	24469	0.570000	440530	0.00000000000073
n_4	P_5	S_4	27100	0.630000	487870	0.00000000000075
n_4	P_5	S_5	14558	0.340000	262183	0.00000000000064
n_4	P_6	S_1	373	0.010000	6714	68.65206910370300
n_4	P_6	S_2	0	6.300000	5399924	92.73994601641932
n_4	P_6	S_3	37	0.000000	1062	0.00000000000000
n_4	P_6	S_4	0	5.040000	4499979	0.00000000000000
n_4	P_6	S_5	29	0.000000	679	0.00000000000000
n_4	P_7	S_1	772	0.000000	13896	0.00008417660057
n_4	P_7	S_2	1053	0.020000	48792	0.00000002399222
n_4	P_7	S_3	235	0.000000	4497	0.00294671602861
n_4	P_7	S_4	202	0.000000	4021	0.00014688721405
n_4	P_7	S_5	194	0.000000	3937	0.00004830911788
n_4	P_8	S_1	3469	0.660000	62442	0.00003493473194
n_4	P_8	S_2	0	56.500000	5397923	0.00146317834984
n_4	P_8	S_3	1789	0.360000	32598	0.00003493473197
n_4	P_8	S_4	452	0.090000	8306	0.00000000000000
n_4	P_8	S_5	995	0.420000	36720	0.00000000000000
n_5	P_0	S_1	715	0.010000	15730	0.00000000000000
n_5	P_0	S_2	721	0.010000	40094	0.00000000941281
n_5	P_0	S_3	260	0.000000	5828	0.00000000000000
n_5	P_0	S_4	521	0.010000	11570	0.00000000000000
n_5	P_0	S_5	322	0.010000	7192	0.00000000000000
n_5	P_1	S_1	34381	0.400000	756382	0.00000000000378
n_5	P_1	S_2	777	0.010000	43923	0.00000001149362
n_5	P_1	S_3	30524	0.420000	671750	0.00000000000293
n_5	P_1	S_4	24517	0.330000	539735	0.00000000000368
n_5	P_1	S_5	24770	0.340000	545200	0.00000000000245
n_5	P_2	S_1	4215	0.210000	92730	0.00739604033413
n_5	P_2	S_2	622	0.070000	34663	0.00000000568240
n_5	P_2	S_3	3767	0.210000	82960	0.00000000000000
n_5	P_2	S_4	266	0.020000	6032	0.00000000000000
n_5	P_2	S_5	3011	0.150000	66329	0.00000000000000
n_5	P_3	S_1	639	0.020000	14058	0.77054780344068
n_5	P_3	S_2	1348	0.100000	75196	0.77054779827758
n_5	P_3	S_3	410	0.020000	9262	0.77054779974476
n_5	P_3	S_4	275	0.010000	6461	0.77054780308046
n_5	P_3	S_5	799	0.030000	17791	0.77054780658452

Tabela 6.10: Resultados das simulações numéricas.

n_k	P_j	S_i	n_{it}	t	n_{av}	$f_{k,j,i}$
n_5	P_4	S_1	1165	0.040000	25630	-78.90603250568600
n_5	P_4	S_2	0	9.770000	6599980	-44.68180845873814
n_5	P_4	S_3	729	0.060000	32677	-78.90603250568577
n_5	P_4	S_4	1199	0.110000	53646	-78.90603250568411
n_5	P_4	S_5	989	0.090000	44252	-78.90603250568572
n_5	P_5	S_1	59233	2.240000	1303126	0.00000000000197
n_5	P_5	S_2	668	0.060000	38464	0.00000001090858
n_5	P_5	S_3	48048	1.930000	1057164	0.00000000000186
n_5	P_5	S_4	41618	1.660000	915725	0.000000000000205
n_5	P_5	S_5	48657	1.910000	1070604	0.000000000000200
n_5	P_6	S_1	681	0.020000	14982	84.57139386682532
n_5	P_6	S_2	0	9.720000	6599837	116.15217507143724
n_5	P_6	S_3	237	0.010000	5591	0.000000000000006
n_5	P_6	S_4	0	7.710000	5599975	0.000000000000000
n_5	P_6	S_5	527	0.020000	11934	0.000000000000017
n_5	P_7	S_1	401	0.000000	8822	0.01085561980865
n_5	P_7	S_2	1389	0.030000	77782	0.00000002223773
n_5	P_7	S_3	184	0.000000	4620	0.00038718387194
n_5	P_7	S_4	33	0.000000	1427	0.000000000000000
n_5	P_7	S_5	32	0.000000	1726	0.000000000000000
n_5	P_8	S_1	15175	5.310000	333850	0.000000000000002
n_5	P_8	S_2	0	103.200000	6596723	0.00072079789643
n_5	P_8	S_3	545	0.200000	12570	0.000000000000000
n_5	P_8	S_4	739	0.270000	16606	0.000000000000000
n_5	P_8	S_5	53678	18.880000	1181091	0.00000020674376
n_6	P_0	S_1	1064	0.020000	27664	0.000000000000000
n_6	P_0	S_2	888	0.020000	57977	0.00000001265945
n_6	P_0	S_3	827	0.020000	21630	0.000000000000000
n_6	P_0	S_4	637	0.010000	16741	0.000000000000000
n_6	P_0	S_5	665	0.010000	17544	0.000000000000000
n_6	P_1	S_1	47763	0.780000	1241838	0.000000000000360
n_6	P_1	S_2	992	0.020000	65027	0.00000000635679
n_6	P_1	S_3	40362	0.740000	1049722	0.000000000000473
n_6	P_1	S_4	38604	0.700000	1004048	0.000000000000436
n_6	P_1	S_5	34129	0.620000	887670	0.000000000000288
n_6	P_2	S_1	8563	0.640000	222638	0.000000000000000
n_6	P_2	S_2	791	0.140000	51350	0.00000000704197
n_6	P_2	S_3	4457	0.340000	115984	0.000000000000000
n_6	P_2	S_4	364	0.020000	9676	0.000000000000000
n_6	P_2	S_5	7140	0.530000	185743	0.000000000000000

Tabela 6.11: Resultados das simulações numéricas.

n_k	P_j	S_i	n_{it}	t	n_{av}	$f_{k,j,i}$
n_6	P_3	S_1	891	0.040000	23166	6.30173362513613
n_6	P_3	S_2	1627	0.180000	105876	0.49274091967089
n_6	P_3	S_3	592	0.030000	15728	0.49274093086275
n_6	P_3	S_4	822	0.030000	21757	0.49274092152922
n_6	P_3	S_5	805	0.040000	21186	0.49274093025696
n_6	P_4	S_1	1582	0.070000	41132	-94.68723900682325
n_6	P_4	S_2	0	13.360000	7799976	-53.15839694079122
n_6	P_4	S_3	1009	0.110000	53316	-94.68723900682348
n_6	P_4	S_4	1184	0.140000	62604	-94.68723900682294
n_6	P_4	S_5	914	0.100000	48304	-94.68723900682262
n_6	P_5	S_1	94446	5.700000	2455596	0.000000000000453
n_6	P_5	S_2	878	0.120000	59016	0.00000000678411
n_6	P_5	S_3	58430	3.630000	1519308	0.000000000000313
n_6	P_5	S_4	71053	4.410000	1847506	0.000000000000478
n_6	P_5	S_5	54783	3.400000	1424536	0.000000000000268
n_6	P_6	S_1	649	0.030000	16874	100.49071862994784
n_6	P_6	S_2	0	13.390000	7799728	136.39258835038919
n_6	P_6	S_3	273	0.020000	7585	0.000000000000004
n_6	P_6	S_4	0	10.750000	6699985	0.000000000000000
n_6	P_6	S_5	408	0.020000	11059	0.000000000000009
n_6	P_7	S_1	485	0.000000	12610	1.44816208220457
n_6	P_7	S_2	1698	0.050000	110989	0.00000003310026
n_6	P_7	S_3	72	0.010000	2794	0.00000014769229
n_6	P_7	S_4	32	0.000000	1749	0.000000000000000
n_6	P_7	S_5	31	0.000000	1619	0.000000000000000
n_6	P_8	S_1	0	57.550000	2600000	0.00000000001863
n_6	P_8	S_2	0	169.860000	7794333	0.00100030568326
n_6	P_8	S_3	12835	7.430000	334408	0.00000346424169
n_6	P_8	S_4	976	0.570000	25665	0.000000000000000
n_6	P_8	S_5	998	0.590000	26198	0.000000000000000

Tabela 6.12: Resultados das simulações numéricas.

6.6.1 Comentários técnicos

Criamos um programa, em linguagem C, para nos ajudar a compilar os dados brutos apresentados anteriormente. Este programa executou o cálculo da razão de desempenho para cada um dos métodos em cada um dos problemas selecionados. Foram feitos 6 gráficos para cada medida, relacionados às dimensões n_1, n_2, \dots, n_6 sob as quais os problemas foram calculados.

Os gráficos foram montados com o auxílio do programa Matlab.

6.6.2 Gráficos de perfis

Nesta seção apresentamos os gráficos para as três medidas, **Número de Iterações**, **Tempo** (medido em segundos) e **Número de Avaliações de Função**.

Os perfis com medida baseada em Número de Iterações são exibidos nas Figuras 6.8, 6.9 e 6.10. Os baseados em Tempo estão dispostos nas Figuras 6.11, 6.12 e 6.13. Os perfis baseados no número de Avaliações de Função são mostrados nas Figuras 6.14, 6.15 e 6.16.

A montagem dos perfis de desempenho levou apenas em consideração os Problemas P_1, \dots, P_8 , excluindo o problema do Parabolóide. Isto ocorreu para avaliar o comportamento em funções não-convexas.

O comportamento do MONCAS para o referido problema pode ser avaliado a partir das tabelas de dados brutos.

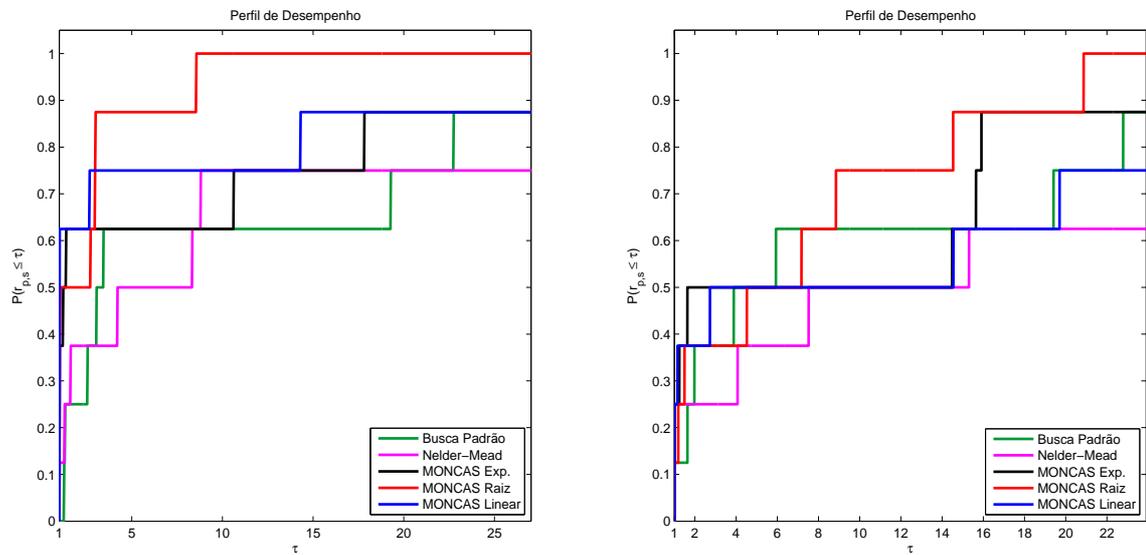


Figura 6.8: Perfil de Desempenho Baseado em Número de Iterações para dimensões n_1 (esquerda) e n_2 (direita).

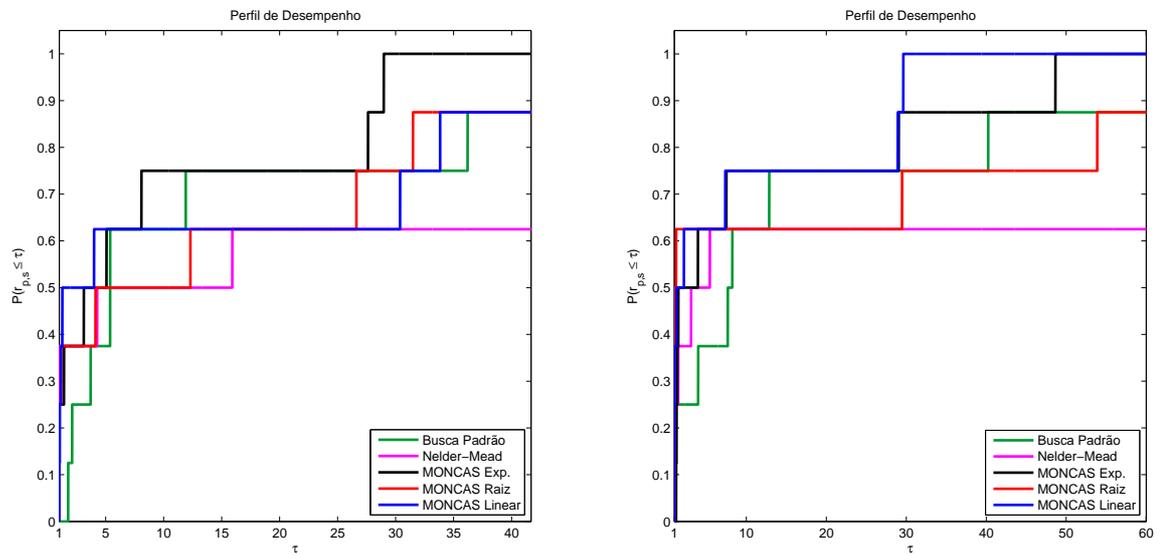


Figura 6.9: Perfil de Desempenho Baseado em Número de Iterações para dimensões n_3 (esquerda) e n_4 (direita).

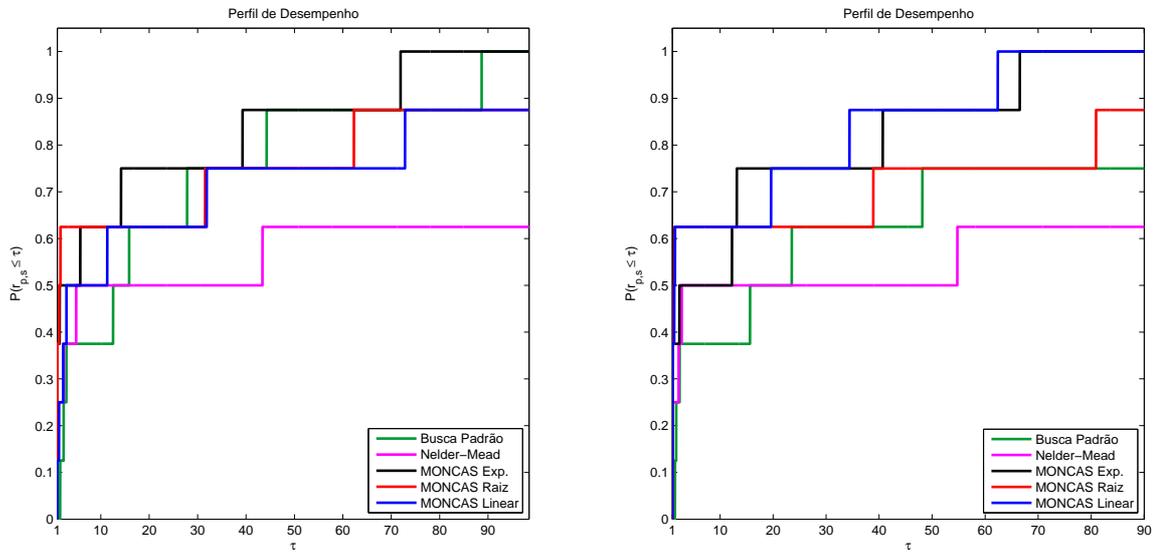


Figura 6.10: Perfil de Desempenho Baseado em Número de Iterações para dimensões n_5 (esquerda) e n_6 (direita).

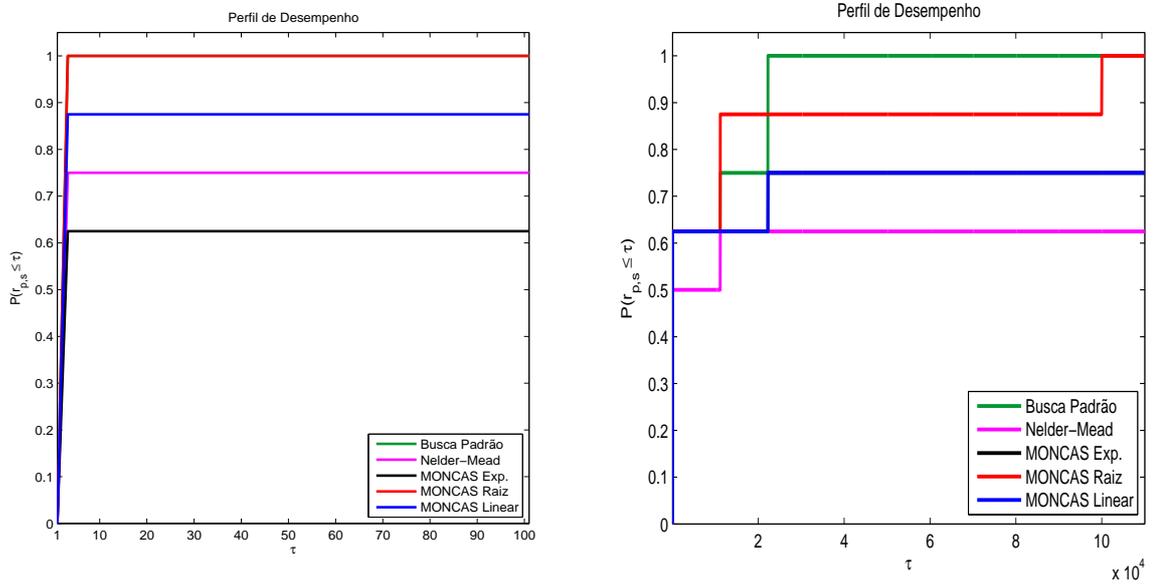


Figura 6.11: Perfil de Desempenho Baseado em Tempo para dimensões n_1 (esquerda) e n_2 (direita).

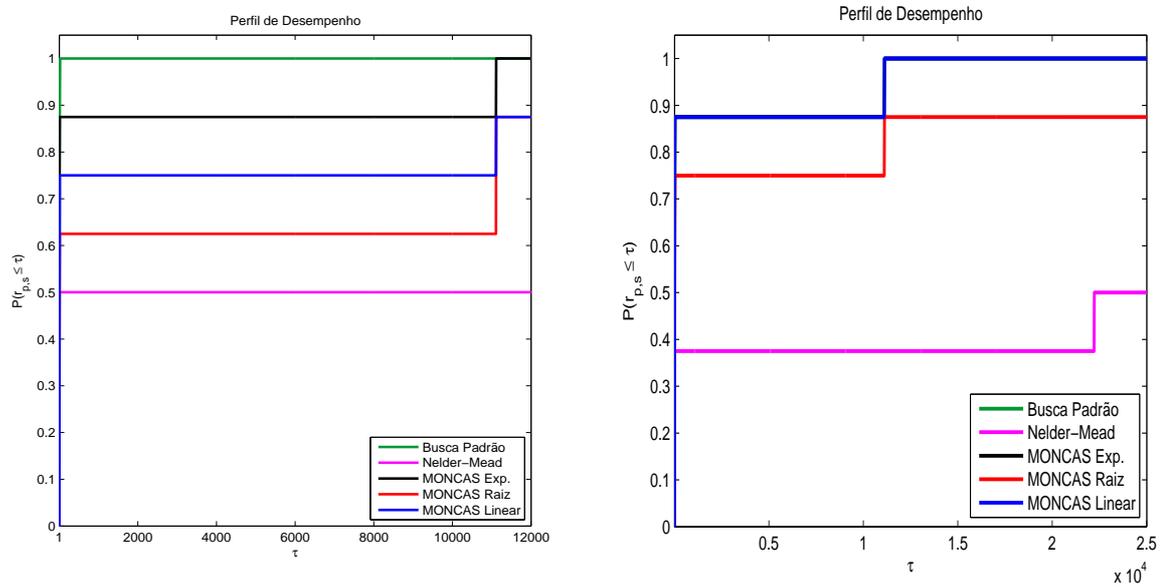


Figura 6.12: Perfil de Desempenho Baseado em Tempo para dimensões n_3 (esquerda) e n_4 (direita).

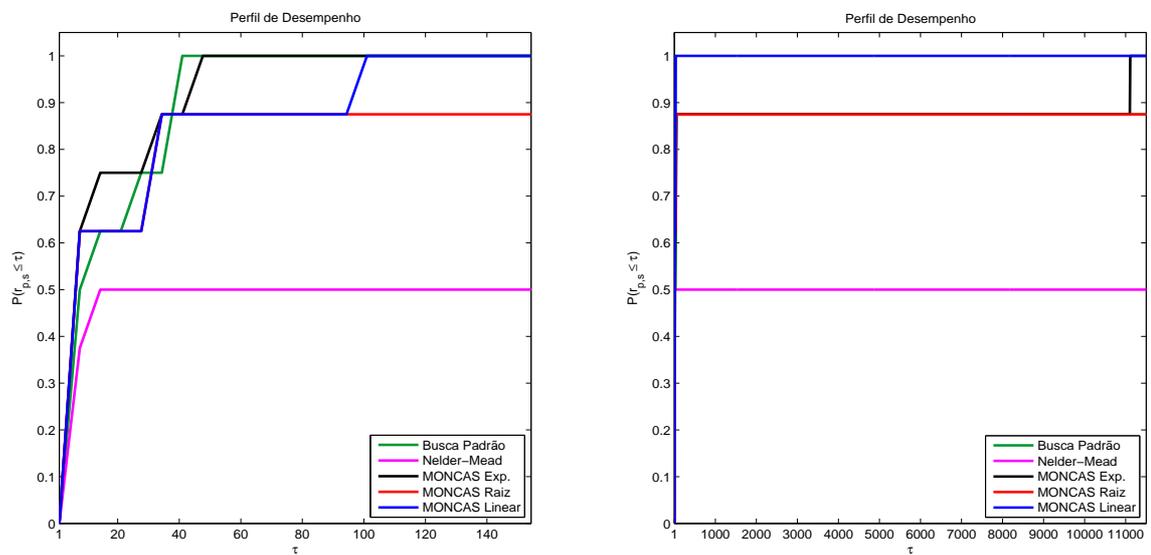


Figura 6.13: Perfil de Desempenho Baseado em Tempo para dimensões n_5 (esquerda) e n_6 (direita).

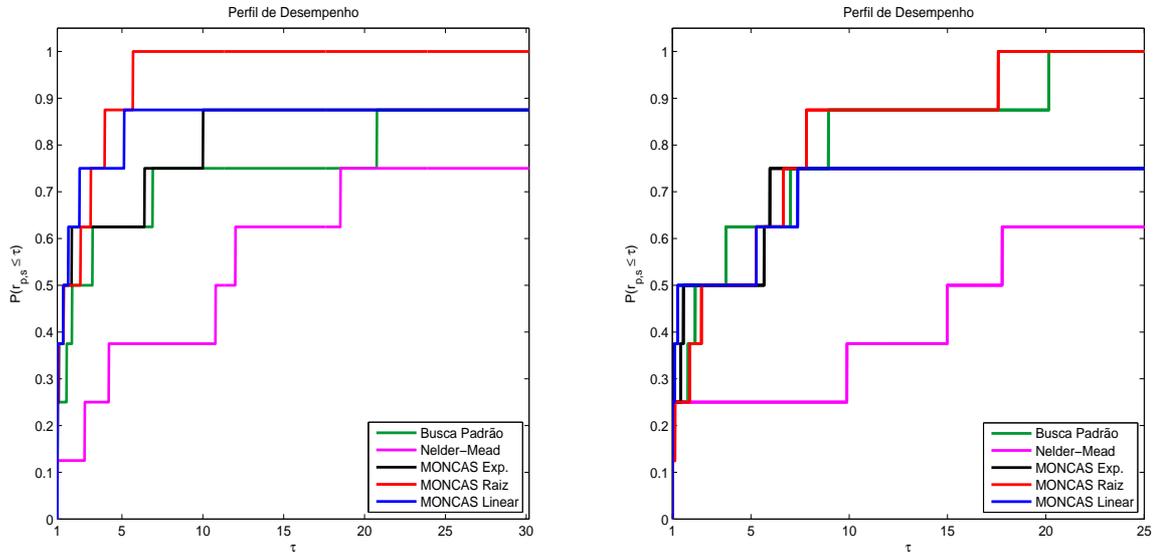


Figura 6.14: Perfil de Desempenho Baseado em Número de Avaliações de Função para dimensões n_1 (esquerda) e n_2 (direita).

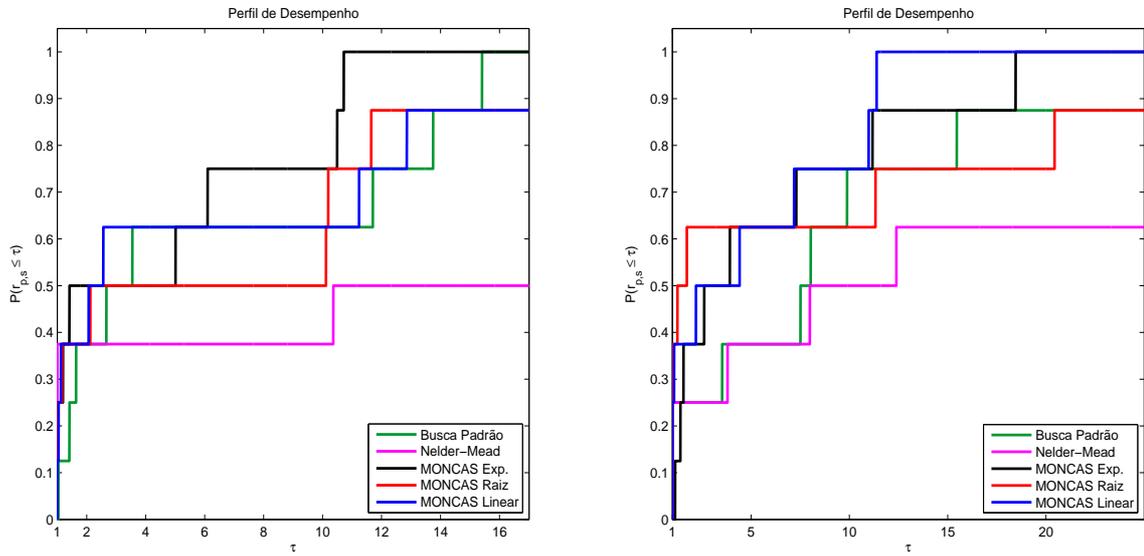


Figura 6.15: Perfil de Desempenho Baseado em Número de Avaliações de Função para dimensões n_3 (esquerda) e n_4 (direita).

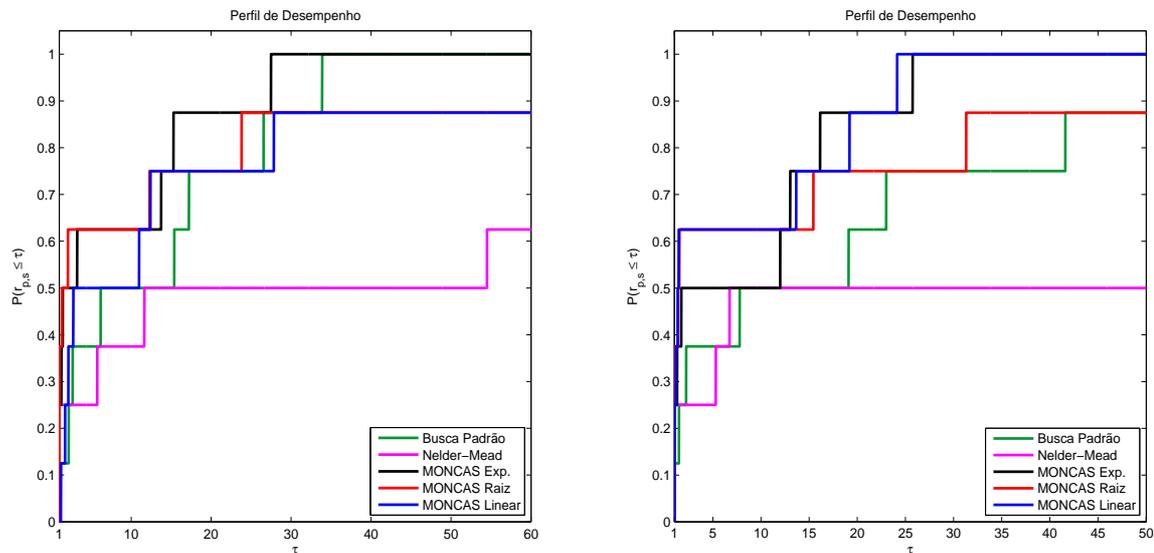


Figura 6.16: Perfil de Desempenho Baseado em Número de Avaliações de Função para dimensões n_5 (esquerda) e n_6 (direita).

6.6.3 Alta variação do perfil gerado por medição de tempo

Observamos um perfil extremamente mal-escalonado ao se usar a medida de tempo físico para uma estimativa do tempo computacional. Isto se deve ao fato da rápida convergência a um ponto estacionário, em alguns problemas, levando a um tempo calculado muito baixo (9×10^{-7}).

Por exemplo, qualquer método que levou 0.9 segundo para resolver o problema tem uma razão

$$r_{p,s} = \frac{0.9}{9 \times 10^{-7}} = 1 \times 10^6.$$

A existência destes valores demonstra que os métodos empregados estão em uma categoria de alta eficiência, o que põe o nosso método em posição de competição com os demais.

6.7 Análise dos resultados

Ao analisarmos os gráficos observamos que, em todos eles, o Método MONCAS atinge o valor de probabilidade 1.0 de se obter uma estimativa ao minimizador dentro do limite máximo de iterações.

Se desconsiderarmos os gráficos relativos às medidas de tempo (tempo físico medido em segundos) podemos observar que o Método MONCAS sempre tem pelo menos uma estratégia que apresenta eficiência superior, além de sua robustez.

Esta eficiência se alterna entre os três tipos de funções que definem o Passo Externo. A alternância ocorre com mudanças de medida tomada e dimensão dos problemas.

Este dado é interessante e deve ser levado em consideração em estudos futuros focados na determinação de um Passo Externo ótimo.

Um comentário pertinente que surge ao se defrontar o pseudo-código do método MONCAS com os perfis de desempenho para a medida de número de avaliações de função, é que era de se esperar que o número de avaliações de função dentro do método fosse, em média, próximo do dobro do número obtido pelo Método de Busca Padrão.

Contudo podemos observar que há um ganho de eficiência do Método MONCAS neste sentido. Isso é, muito provavelmente, devido ao uso do caso de Decaimento Melhorado.

Em resumo, podemos notar que esta estratégia de Análise por Secantes e a escolha de um Passo Externo, as quais o MONCAS emprega, é robusta e eficiente, sendo uma boa escolha para minimizar uma função menos regular.

6.8 Passo Externo e dimensão do espaço E

Durante o processo de análise por perfis de desempenho utilizamos perfis distintos para cada uma das dimensões sob as quais executamos as simulações numéricas.

Ao analisarmos estes perfis notamos que as variações do MONCAS (diferindo apenas com o tipo da função de atualização do Passo Externo) não obtiveram resultados semelhantes de dimensão para dimensão.

Visto isso, apresentamos um breve levantamento do comportamento dos Passos Externos em contraste com a dimensão a que foi aplicado. Neste ponto, preferimos utilizar uma metodologia diferente da que foi utilizada anteriormente.

Ao invés de construirmos um novo perfil de desempenho, utilizamos uma medida de número médio de avaliações de função por dimensão para visualizar este comportamento. Porém a tabela de resultados das simulações numéricas mostra que há casos de falha, no sentido que a salvaguarda para número de iterações foi atingida.

Devido a esta questão, decidimos separar a análise em dois casos:

- Avaliações em falha não são contabilizadas;
- Avaliações em falha são contabilizadas.

Para avaliações que não contabilizamos as avaliações para um procedimento que falhou, exibimos a Tabela 6.13 que relaciona a dimensão do espaço E e o tipo de Passo Externo para o número de sucessos do método. Na Tabela 6.14 exibimos a média do número de avaliações de função em cada dimensão. A Figura 6.17 mostra um gráfico destes dados.

	P.E. Exponencial	P.E. em Raiz	P.E. Linear
n_1	7	8	7
n_2	7	8	6
n_3	8	7	7
n_4	8	7	8
n_5	8	7	8
n_6	8	7	8

Tabela 6.13: Número de términos bem sucedido por Passo Externo e dimensão.

	P.E. Exponencial	P.E. em Raiz	P.E. Linear
n_1	231640.00	2028.5	205285.71
n_2	368475.29	15349.13	828986.33
n_3	47509.25	470648.71	476650.14
n_4	107111.25	762826.43	85973.00
n_5	234573.00	1019943.86	367365.88
n_6	387355.63	1381855.71	325789.38

Tabela 6.14: Média de número de avaliações de função por Passo Externo e dimensão.

	P.E. Exponencial	P.E. em Raiz	P.E. Linear
n_1	3068.86	2028.50	1988.86
n_2	368475.29	15349.13	470651.17
n_3	47509.25	49220.29	55221.86
n_4	107111.25	119972.29	85973.00
n_5	234573.00	219947.43	367365.88
n_6	387355.63	424715.00	325789.38

Tabela 6.15: Média de número de avaliações de função por Passo Externo e dimensão.

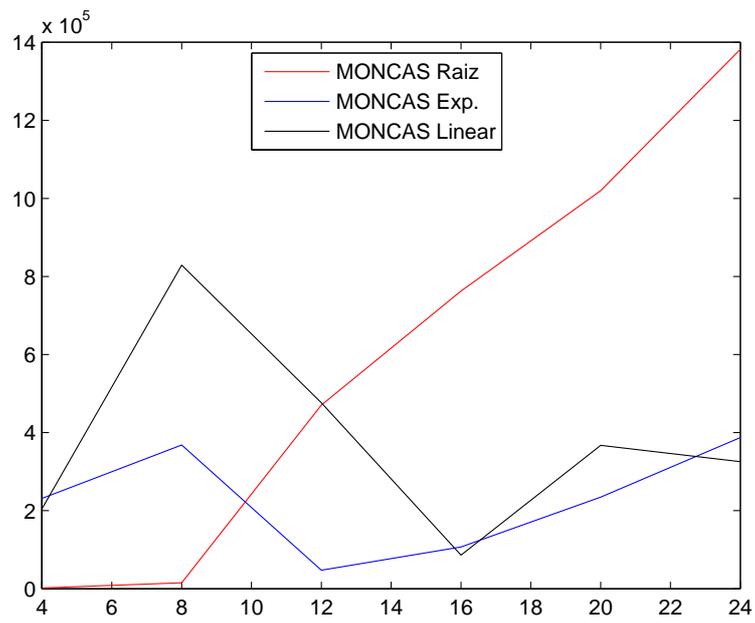


Figura 6.17: Gráfico de média de número de avaliações de função por dimensão do espaço considerando casos de salvaguarda.

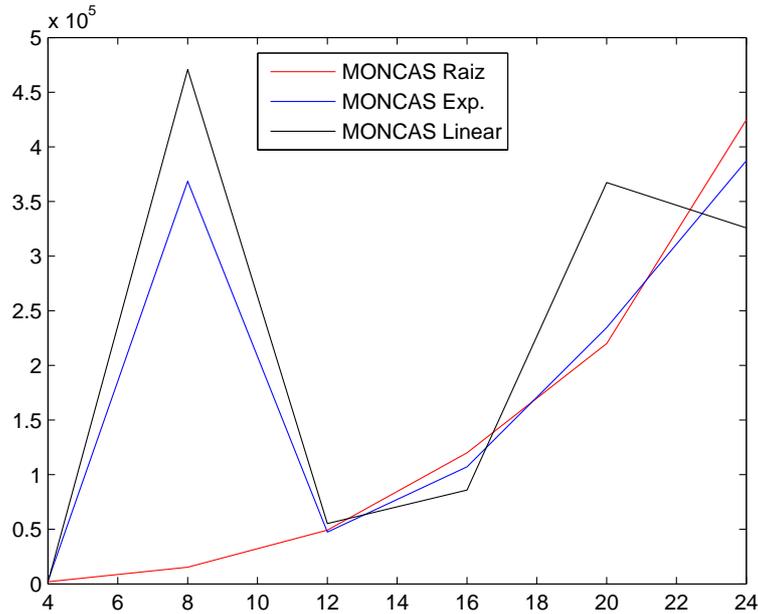


Figura 6.18: Gráfico de média de número de avaliações de função por dimensão do espaço desconsiderando casos de salvaguarda.

Para a análise sobre os casos onde não foram atingidos o número máximo de iterações ainda consideramos a Tabela 6.13. Mas o valor da média é alterado (uma vez que não há contagem de avaliações de função para os ensaios desconsiderados). A tabela desta nova média é apresentada em 6.15. Seu gráfico é esboçado na Figura 6.18.

As duas formas diferem no seguinte sentido: ao contabilizarmos o número de avaliações de função para os casos onde se atingiu a salvaguarda do método estamos acrescentando a característica de robustez ao gráfico. Com isto, o gráfico do caso de análise feita desconsiderando tais casos torna-se puramente um gráfico de eficiência.

Ao contrastarmos os dois gráficos percebemos que o Passo Externo Linear se apresenta como um opção muito instável para o método, uma vez que ele apresenta uma inclinação muito variável de dimensão a dimensão.

Podemos ainda perceber que o Passo Externo em Raiz Quadrada, além de robustez, perde eficiência a medida que aumentamos a dimensão do problema. Vistos estes dois pontos, somos levados a considerar o Passo Externo Exponencial como uma primeira opção mais viável para executarmos o MONCAS.

7 CONCLUSÕES

Neste trabalho apresentamos o método de busca direta¹ MONCAS, que faz uso de uma nova estratégia de abordagem de solução para problemas de otimização não linear com funções de vários ótimos.

Essa nova estratégia é composta de dois pontos principais:

- Utilização de um segundo parâmetro de passo;
- Utilização de secantes para analisar comportamento de variações da função.

que são utilizados em conjunto com argumentos presentes no Método de Busca Padrão e Método Nelder-Mead.

O Método utiliza como um de seus parâmetros uma função que atualiza o valor do segundo parâmetro de passo (passo externo). Executamos simulações numéri-

¹Este pode ser classificado como método de busca direta, uma vez que em todo o desenvolvimento do método não assumimos a existência de derivadas em qualquer ponto da função. Tão pouco assumimos que as funções se restringem a um modelo onde continuidade ou suavidade são assumidas presentes.

cas para observar o comportamento de minimização em três variações deste passo externo:

- Passo externo com uma função exponencial;
- Passo externo com uma função radicial;
- Passo externo com uma função linear.

Comparamos seus resultados com simulações executadas utilizando os métodos de Busca Padrão e Nelder-Mead.

A partir dos dados gerados por essa metodologia podemos ver que o Método MON-CAS, em pelo menos uma de estratégias de Passo Externo adotadas, obteve altíssima robustez e uma eficiência muito boa em comparação com os outros dois métodos analisados. A análise destes dados sugere que a estratégia de passo é também uma função da dimensão do espaço E .

Contudo, ao analisarmos os perfis de desempenho referentes ao número de avaliações de função, percebemos que a estratégia de passo externo exponencial apresenta desempenho superior. Ele não só apresenta eficiência bem competitiva como também atinge robustez máxima em um número maior de gráficos de perfil que as demais estratégias.

O fato de termos obtido uma eficiência alta na medida de número de avaliações de função nos mostrou que a análise de casos que resultam em decaimento melhorado se mostrou extremamente eficiente em encontrar um ponto fixo mais rapidamente.

Por outro lado, os diagramas construídos na medida tempo nos indicam que há um custo computacional alto ao se escolher uma estratégia de passo externo inadequada

para o problema que se deseja trabalhar. Dito isso, podemos reforçar elementos sobre a discussão do passo externo levantados ao final do Capítulo 5, onde questionamos a possibilidade de o passo externo depender das características locais da função.

Mesmo podendo ser imensamente aprofundado em várias vertentes de pesquisa, tais como *Variação de Passo Externo*, *Medidas de Profundidade de Vale* e *Utilização de Memória em Autoanálise* entre outras, podemos notar que o Método MONCAS se apresenta como uma ferramenta poderosa para a otimização não linear.

REFERÊNCIAS

- AUDET, C. **Convergence results for pattern search algorithms are tight**. [S.l.]: Department of Computational and Applied Mathematics, Rice University, 1998.
- CIAPPINA, J. R. **Sobre um método de busca direta sem derivada, com decréscimo fortalecido**. 2008. Dissertação (Mestrado em Ciência da Computação) — Universidade Estadual de Campinas.
- COXETER, H. S. M. **Regular polytopes**. 3.ed. [S.l.]: Courier Dover Publications, 1973.
- DAVIS, C. Theory of positive linear dependence. **American Journal of Mathematics**, [S.l.], v.76, n.4, p.733–746, 1954.
- DOLAN, E. D.; LEWIS, R. M.; TORCZON, V. On the local convergence of Pattern Search. **SIAM Journal on Optimization**, [S.l.], v.14, n.2, p.567–583, 2003.
- DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. **Mathematical Programming**, [S.l.], v.91, n.2, p.201–213, January 2002.
- LEWIS, R. M.; TORCZON, V.; TROSSET, M. W. Direct search methods: then and now. **Journal of Computational and Applied Mathematics**, [S.l.], v.124, p.191–207, 2000.

MCKINNON, K. I. M. Convergence of The Nelder-Mead Simplex Method to a Nonstationary Point. **SIAM Journal on Optimization**, [S.l.], v.9, p.148–158, 1998.

MORé, J. J.; GARROW, B. S.; HILLSTROM, K. E. Testing Unconstrained Optimization Software. **ACM Trans. Math. Softw.**, [S.l.], v.7, n.1, p.17–41, 1981.

NELDER, J. A.; MEAD, R. A simplex method for function minimization. **The Computer Journal**, [S.l.], v.7, p.308–313, 1965.

OMRAN, M. G.; MAHDAVI, M. Global-best harmony search. **Applied Mathematics and Computation**, [S.l.], v.198, p.643–656, 2008.

OSMAN, I.; KELLY, J. **Meta-Heuristics - Theory and Applications**. [S.l.]: Springer, 1996.

TORCZON, V. On the convergence of pattern search algorithm. **SIAM Journal on Optimization**, [S.l.], v.7, p.1–25, 1997.