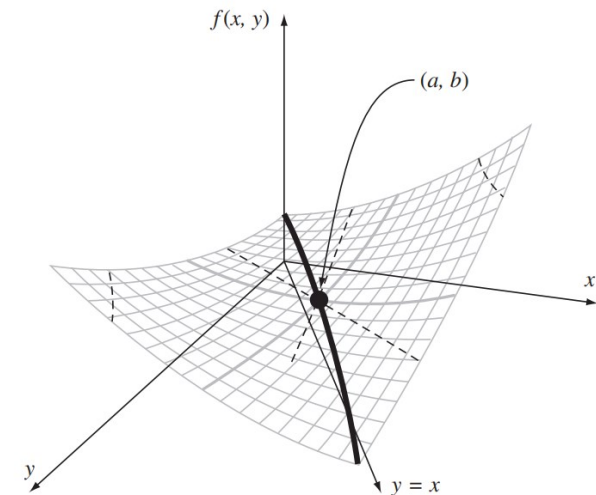
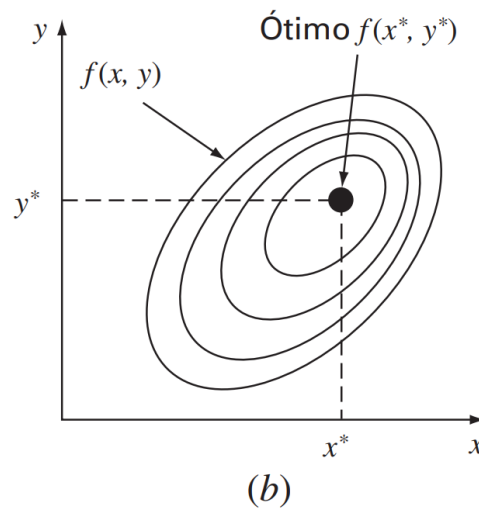
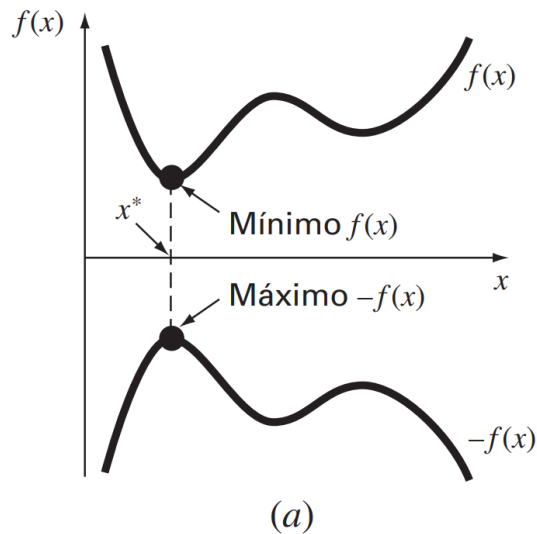




Introdução à Otimização de Projetos de Engenharia

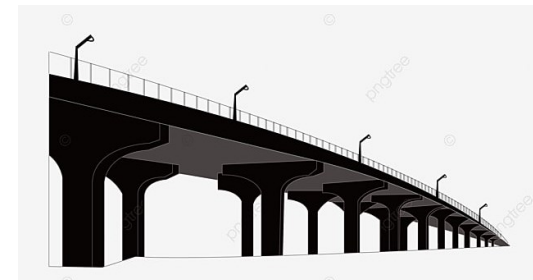
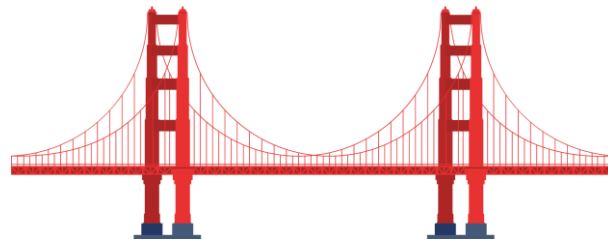
Prof. Dr. Alverlando Silva Ricardo

1. INTRODUÇÃO À OTIMIZAÇÃO

Otimização e a Prática da Engenharia

Exemplo:

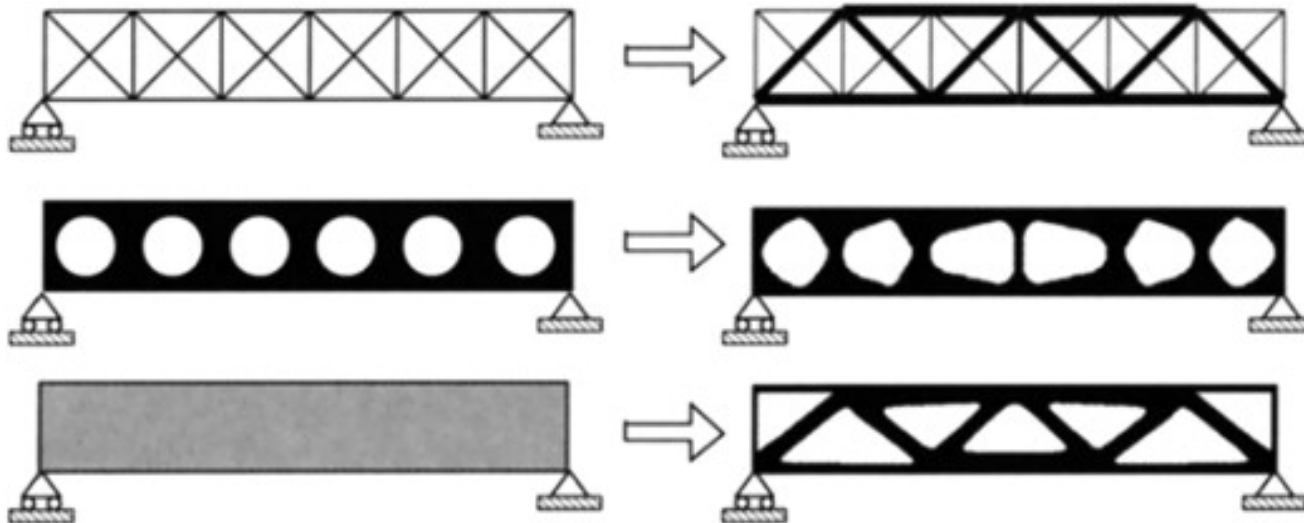
A finalidade de uma ponte é fornecer continuidade no tráfego de um lado para o outro. **Vários tipos de pontes podem servir a esse propósito.** No entanto, analisar e projetar todas as possibilidades pode ser uma tarefa demorada e dispendiosa. Normalmente, **um tipo é selecionado com base em algumas análises preliminares** e assim é projetado em detalhes.



* Imagens da internet

Otimização e a Prática da Engenharia

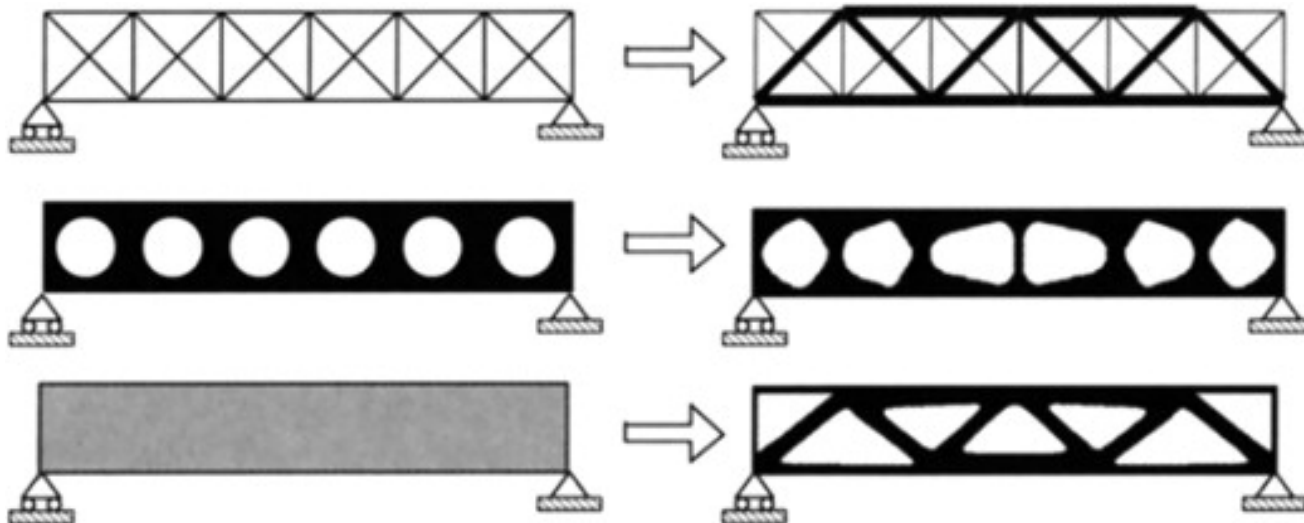
Os engenheiros devem sempre projetar artefatos e produtos que desempenhem **tarefas de forma eficiente**, restritos pelas **limitações físicas** e com **custos baixos**. Portanto, estão sempre confrontando **problemas de otimização** que equilibram **eficiência e limitações**.



* Imagens da internet

Otimização e a Prática da Engenharia

As técnicas de otimização mostram-se extremamente atraentes nesse aspecto onde é possível formular matematicamente um problema de busca de soluções melhores dentre as diversas possíveis.



* Imagens da internet

Otimização e a Prática da Engenharia

1.3 CONVENTIONAL VERSUS OPTIMUM DESIGN PROCESS

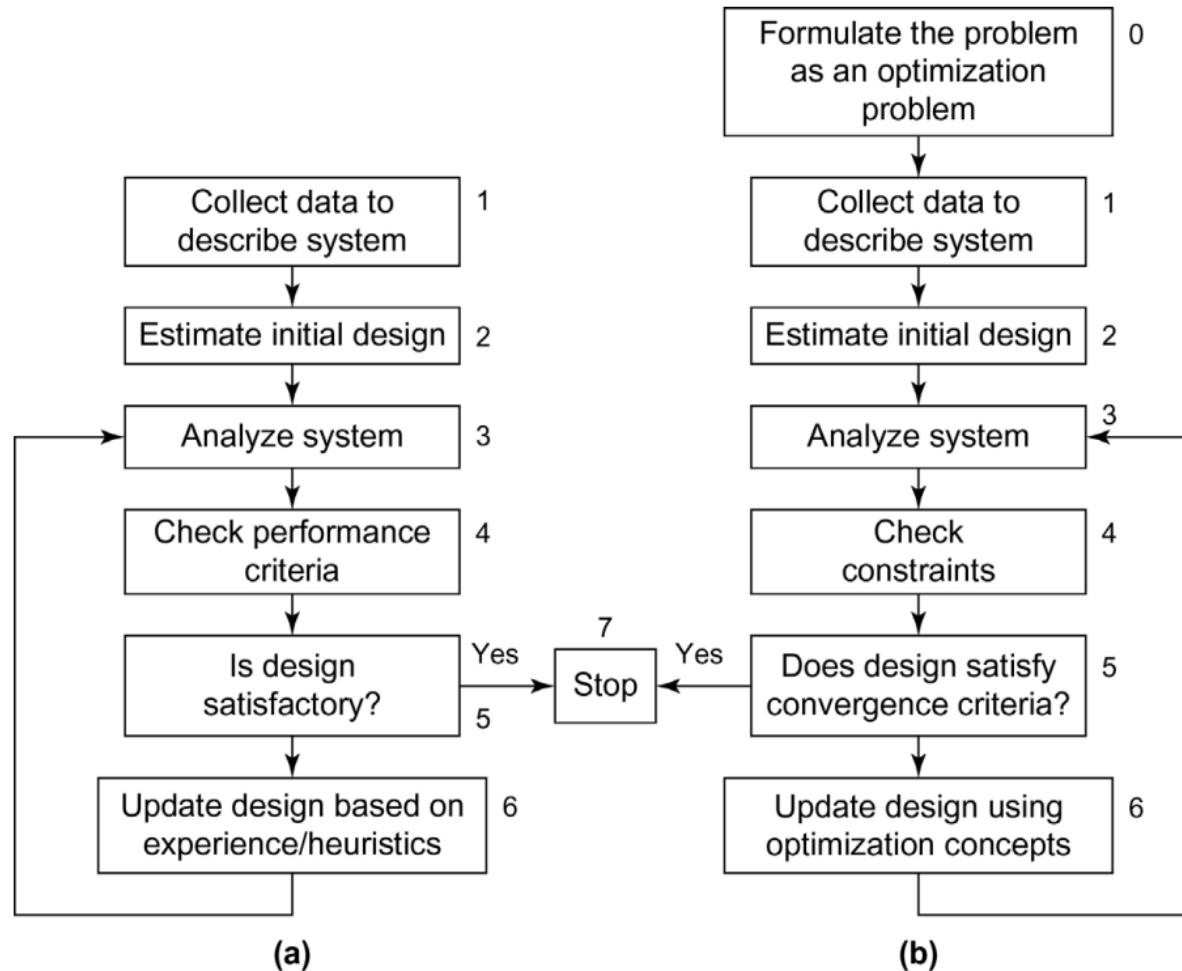


FIGURE 1.2 Comparison of: (a) conventional design method; and (b) optimum design method.

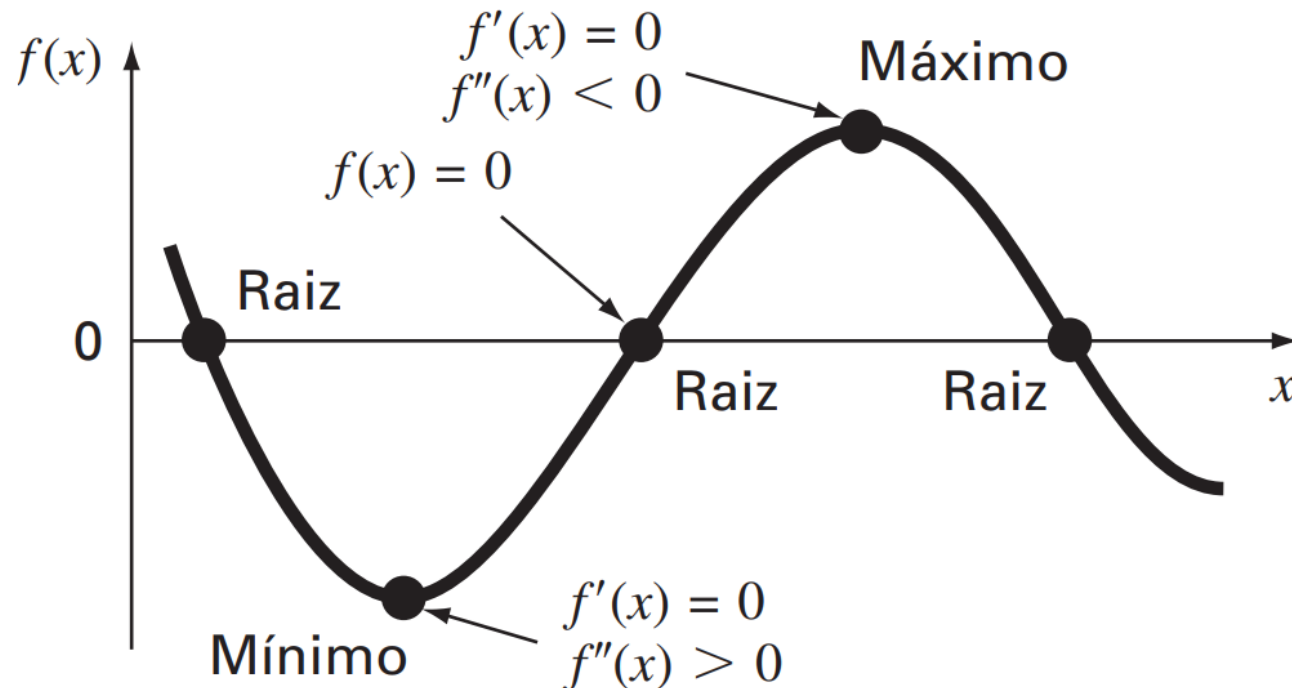
Otimização e a Prática da Engenharia

TABELA PT4.1 Alguns exemplos comuns de problemas de otimização em engenharia.

- Projeto de uma aeronave com peso mínimo e resistência máxima.
 - Trajetória ótima de veículos espaciais.
 - Projeto de estruturas de engenharia civil com custo mínimo.
 - Projeto de reservatórios de água como represas para diminuir os danos de enchentes e ao mesmo tempo fornecendo a máxima potência hídrica.
 - Previsão do comportamento de estruturas pela minimização da energia potencial.
 - Estratégia de corte de material com custo mínimo.
 - Projeto de bombas e equipamento de transferência de calor para eficiência máxima.
 - Maximização da potência de saída de redes elétricas e de máquinas, ao mesmo tempo minimizando a geração de calor.
 - Rota mais curta de um caixeiro-viajante visitando várias cidades em uma viagem de vendas.
 - Planejamento e escalonamento ótimos.
 - Análise estatística e modelos com erro mínimo.
 - Redes de tubulação ótimas.
 - Controle de estoque.
 - Planejamento de manutenção para minimizar custos.
 - Minimização de tempo de espera e de tempo ocioso.
 - Projeto de sistemas de tratamento de dejetos para atingir padrões de qualidade de água com custo mínimo.
-

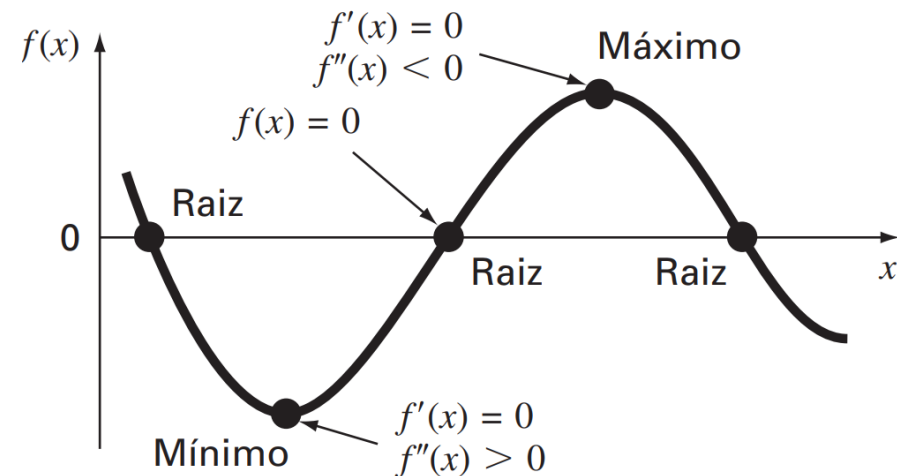
Otimização e a Prática da Engenharia

Matematicamente, a determinação de **raízes** e a **otimização** estão relacionadas. Ambas envolvem a estimativa e a **busca de um ponto na função**.



Otimização e a Prática da Engenharia

A busca por $f'(x) = 0$ é geralmente complicada. Portanto, algumas vezes são usadas aproximações por *diferenças finitas* para obter uma estimativa da derivada.



$$\frac{\partial p_n}{\partial x} = \frac{4f_1 - f_2 - 3f_0}{2h}$$

$$\frac{\partial p_n}{\partial x} = \frac{f_2 - f_0}{2h}$$

*diferenças avançada,
central e atrasada com
três pontos*

$$\frac{\partial p_n}{\partial x} = \frac{f_0 - 4f_1 + 3f_2}{2h}$$

Otimização e a Prática da Engenharia

A tarefa de localização de pontos ótimos é auxiliada por algumas **estruturas matemáticas adicionais** que não são parte da simples determinação de raízes. Isso tende a fazer que **a otimização** seja um problema **mais abordável**, particularmente para o caso **multidimensional**. Desse modo, o uso de técnicas numéricas e de **ferramentas computacionais** se tornam **primordiais**.



OBJETIVOS

OTIMIZAÇÃO ESTRUTURAL é um processo numérico/matemático que visa a obtenção de uma configuração da estrutura que resulta em uma performance ótima segundo alguma medida de desempenho pré-definida (mínima massa, máxima rigidez, máxima carga de flambagem, etc.)

Além disso, devem ser satisfeitas restrições tanto sobre as variáveis de projeto, quanto sobre o comportamento da estrutura (tensão de falha, frequência natural crítica, deslocamentos permitidos, etc.).

OBJETIVOS

Os principais objetivos da otimização estrutural são:

- i. encontrar o menor custo possível no projeto de uma estrutura satisfazendo seus estados limites (de serviço e último, por exemplo);
- ii. eliminar, ou pelo menos diminuir, a dependência da experiência do projetista:

“Um engenheiro com 10 anos de experiência normalmente faz um projeto muito mais rápido e econômico do que alguém que acabou de sair da universidade; a otimização gera um método sistemático de busca pelo melhor projeto de forma a eliminar essa dependência da experiência do projetista.”

As áreas da Otimização Estrutural

1. **Otimização dimensional:** busca encontrar as dimensões da estrutura que minimizem seu custo de fabricação ou peso.

Por exemplo: buscar as dimensões da seção transversal de uma viga (altura e base da viga) que minimizem seu volume.

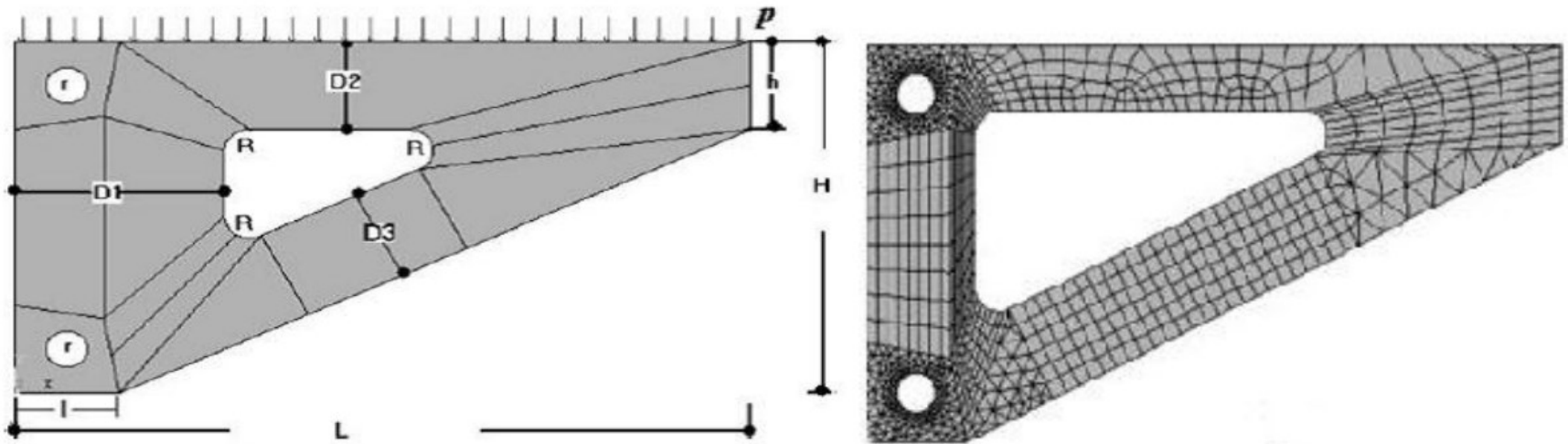


Figura 1. Otimização de estrutura modelada usando o MEF.

As áreas da Otimização Estrutural

2. **Otimização de forma ou geometria:** é aplicada na engenharia civil no projeto de treliças ou torres de transmissão. onde o algoritmo de otimização busca a posição dos nós e a área da seção transversal de cada barra de forma a minimizar o custo final da estrutura e garantir que ela não falhe.

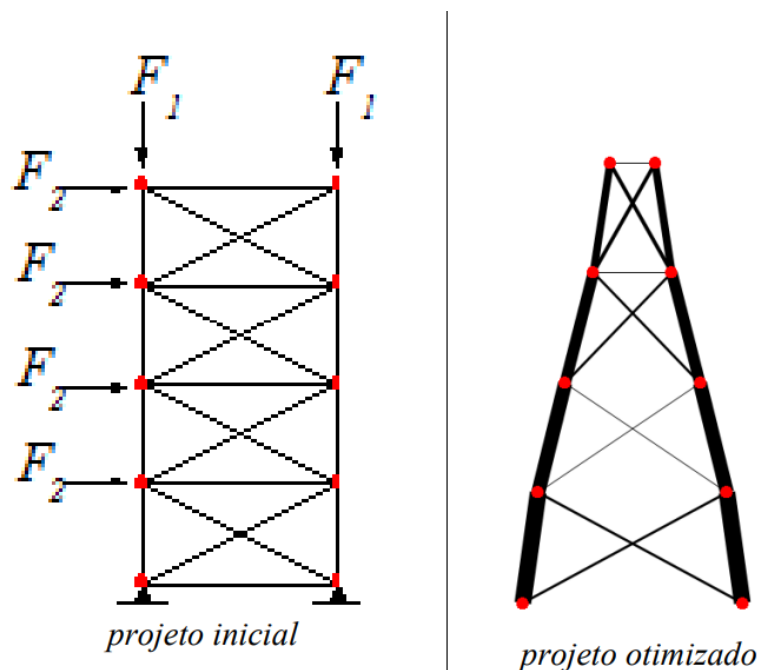


Figura 2. Otimização de geometria de uma torre de transmissão.

As áreas da Otimização Estrutural

3. **Otimização de topologia:** consiste na modificação do padrão de conectividade ou na disposição espacial dos elementos. Em outras palavras, definido um projeto inicial, o algoritmo tem a capacidade de retirar material de qualquer região deste projeto de forma a minimizar seu peso garantindo que a estrutura não falhe.

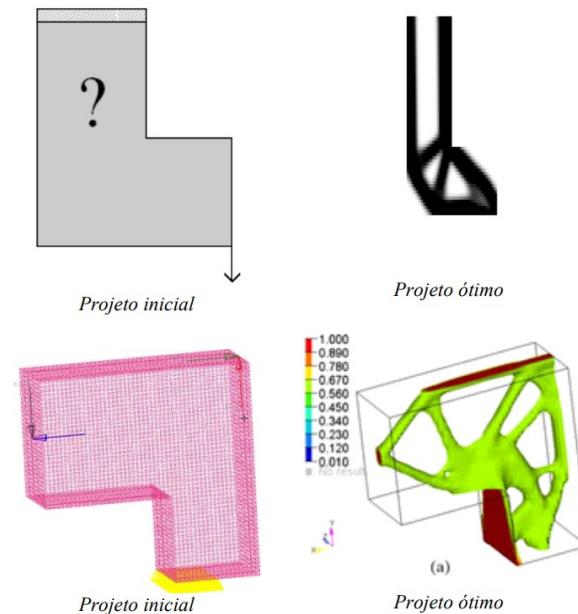


Figura 3. Otimização topológica de estruturas contínuas modeladas pelo MEF.

Curso: Introdução à Otimização de Projetos de Engenharia

Informações gerais:

Ano/Semestre: 2023.1

Disciplina: Introdução à Otimização de Projetos de Engenharia

Natureza: Optativa A partir do 6º Período

Carga Horário: 2 (duas) horas semanais

Docente: Alverlando Ricardo E-mail: alverlando.ricardo@delmiro.ufal.br

Material disponível em:

➤ <https://alverlandoricardo.wixsite.com/professor>

CONTEÚDO

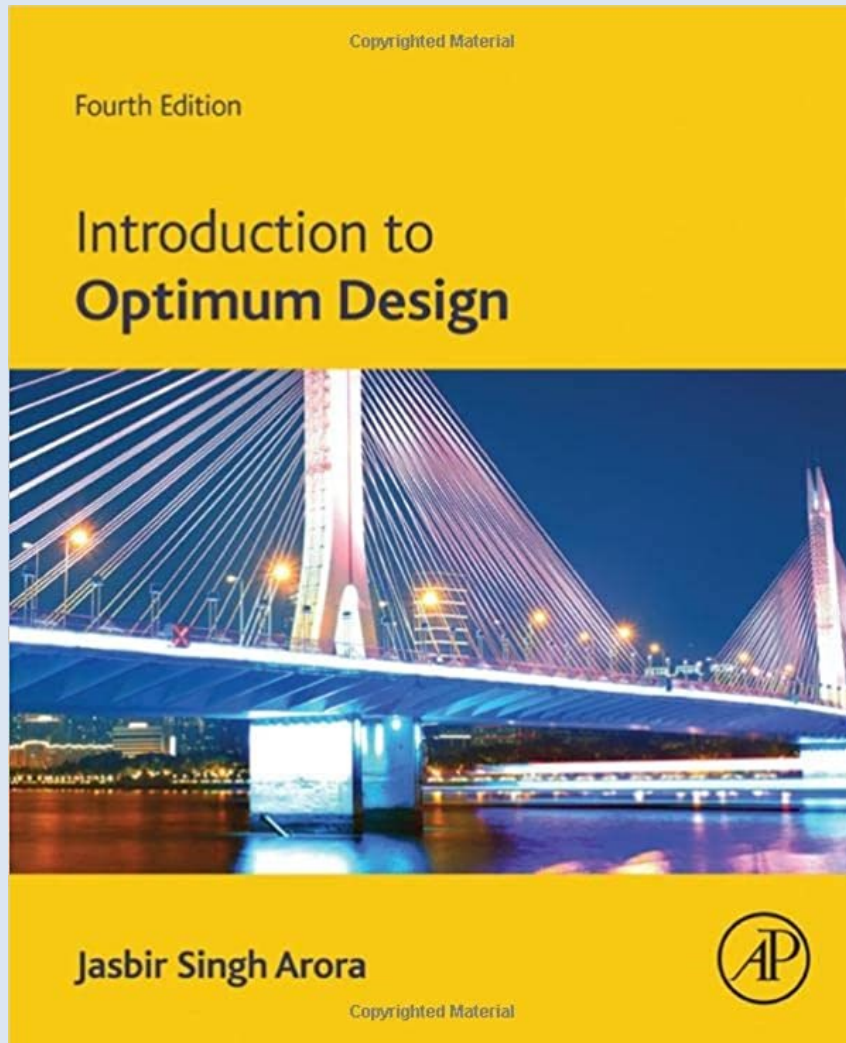
Ementa:

1. Introdução à Otimização
2. Formulação do Problema de Otimização
3. Método de solução gráfica e conceitos básicos de otimização
4. Condições de Otimização
5. Otimização não linear: SOLUÇÃO NUMÉRICA
6. Conceitos e métodos de projeto ótimo de variável discreta
7. Conceitos e métodos de otimização global
8. Métodos de pesquisa inspirados na natureza
9. Conceitos e métodos de projeto otimizado multiobjetivo
10. Tópicos Adicionais sobre Projeto Ideal

A DISCIPLINA: Resistência dos Materiais



REFERENCIAS BÁSICAS



“Introduction to Optimum Design - 4ª edition”

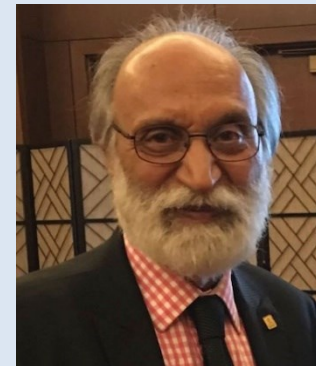
Editora: Academic Press

Páginas: 1381 páginas

ISBN: 0128008067

Ano: 2016

Jasbir S. Arora



Pré-Requisitos

Para um bom entendimento do curso é interessante que o estudante tenha em mente os conhecimentos sobre **Cálculo Básico**, **Álgebra Linear**, **Geometria Analítica**, **Método Numéricos (IC e CN)** e sobre **a Resistências dos Materiais**.

1.1. Terminologia básica e notação

1.1. Terminologia básica e notação

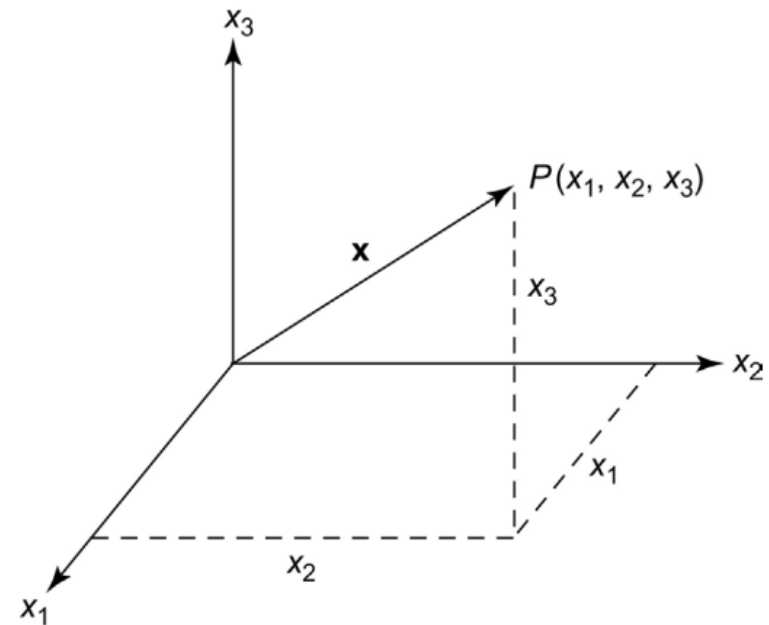
- **Vetores e Pontos:** Como os sistemas realistas geralmente envolvem várias variáveis, é necessário definir e usar algumas **notações convenientes e compactas** para representá-las. As **notações de conjuntos e vetoriais** atendem muito bem a esse propósito.

1.1. Terminologia básica e notação

➤ Os termos vetor e ponto são usados alternadamente, e letras minúsculas em negrito romano são usadas para denotá-los. Letras maiúsculas em negrito romano representam matrizes.

➤ $(x_1, x_2, \dots, x_n)$ é um ponto que consiste em n números. Os n componentes $x_1, x_2, \dots, x_n$ são coletados em um vetor coluna:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = [x_1 \ x_2 \ \dots \ x_n]^T$$

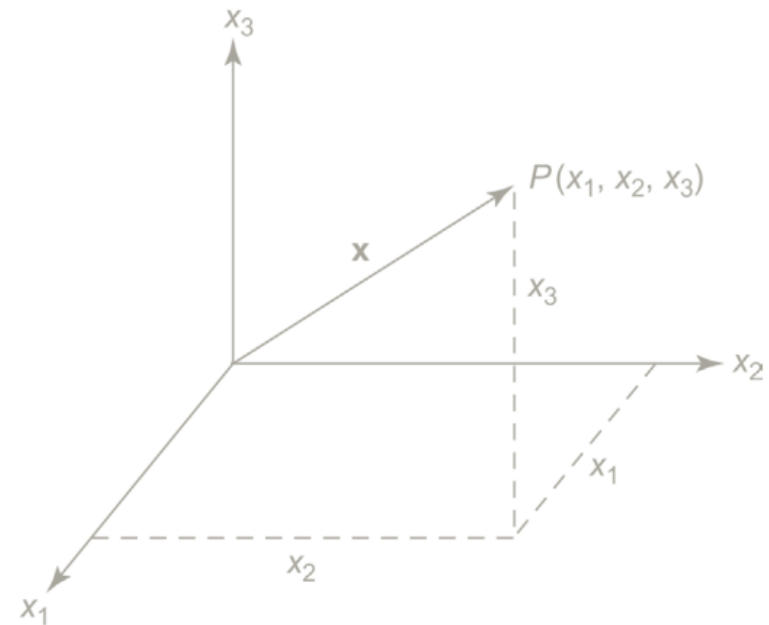


1.1. Terminologia básica e notação

➤ **Letras minúsculas em negrito romano** são usadas para denotar os vetores (**x**). Letras maiúsculas em negrito romano representam matrizes (**X**).

➤ $(x_1, x_2, \dots, x_n)$ é um ponto que consiste em n números. Os n componentes $x_1, x_2, \dots, x_n$ são coletados em um vetor coluna:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = [x_1 \ x_2 \ \dots \ x_n]^T$$



1.1. Terminologia básica e notação

- **Conjuntos:** Muitas vezes lidamos com conjuntos de pontos que satisfazem certas condições. Por exemplo, podemos considerar um conjunto S de todos os pontos com três componentes, com o último tendo um valor fixo de 3, que é escrito como:

$$S = \left\{ \mathbf{x} = (x_1, x_2, x_3) \mid x_3 = 3 \right\}$$

1.1. Terminologia básica e notação

- Conjuntos: Muitas vezes lidamos com conjuntos de pontos que satisfazem certas condições. Por exemplo, podemos considerar um conjunto S de todos os pontos com três componentes, com o último tendo um valor fixo de 3, que é escrito como:

$$S = \left\{ \mathbf{x} = (x_1, x_2, x_3) \mid x_3 = 3 \right\}$$

" S é igual ao conjunto de todos os pontos (x_1, x_2, x_3) com $x_3 = 3$."

A barra vertical ($|$) divide as informações sobre o conjunto S em duas partes: À **esquerda** da barra está a dimensão dos pontos do conjunto; à **direita** estão as propriedades que distinguem esses pontos de outros que não estão no conjunto.

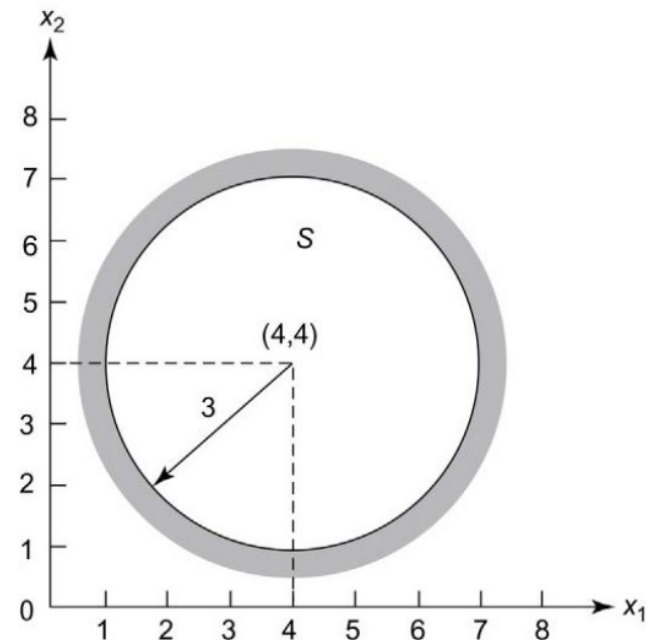
1.1. Terminologia básica e notação

$$S = \left\{ \mathbf{x} \in \mathbb{R}^2 \mid (x_1 - 4)^2 + (x_2 - 4)^2 \leq 9 \right\}$$

O centro do círculo $(4, 4)$ está no conjunto S ? $(4, 4) \in S$?

A origem das coordenadas $(0, 0)$ não pertence S ? $(0, 0) \notin S$?

O conjunto S tem um número infinito de pontos. Muitos outros pontos estão ou não no conjunto!



1.1. Terminologia básica e notação

➤ Notações de Restrições:

Por exemplo, o material do sistema não deve falhar, a demanda deve ser atendida, os recursos não devem ser excedidos e assim por diante.

Restrição "menor ou igual" e é abreviada como " $\leq$ ". Da mesma forma, existem restrições de tipo maior ou igual, abreviadas como " $\geq$ ". Ambos são chamados de restrições de desigualdade.

$$(x_1 - 4)^2 + (x_2 - 4)^2 \leq 9$$

1.1. Terminologia básica e notação

➤ Sobrescritos/Subscritos:

- ❑ Os **sobrescritos** são usados para representar diferentes vetores e matrizes. Por exemplo, $\mathbf{x}^{(i)}$ representa o i -ésimo vetor de um conjunto e $\mathbf{A}^{(k)}$ representa a k -ésima matriz.
- ❑ **Subscritos** são usados para representar componentes de vetores e matrizes. Por exemplo, x_j é o j -ésimo componente de $\mathbf{x}$ e a_{ij} é o i - j -ésimo elemento da matriz $\mathbf{A}$.

Para indicar o intervalo de um subscrito ou sobrescrito, usamos a notação.

$$x_i; i = 1 \text{ to } n \qquad \mathbf{x}^{(j)}; j = 1 \text{ to } k$$

1.1. Terminologia básica e notação

➤ Notação de Soma/Multiplicação:

$$c = x_1 y_1 + x_2 y_2 + \dots + x_n y_n \longrightarrow c = \sum_{i=1}^n x_i y_i$$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} :$$

$$\mathbf{y} = \mathbf{Ax}$$

$$y_i = \sum_{j=1}^n a_{ij} x_j = a_{i1} x_1 + a_{i2} x_2 + \dots + a_{in} x_n; \quad i = 1 \text{ to } m$$

$$\mathbf{y} = \sum_{j=1}^n \mathbf{a}^{(j)} x_j = \mathbf{a}^{(1)} x_1 + \mathbf{a}^{(2)} x_2 + \dots + \mathbf{a}^{(n)} x_n$$

1.1. Terminologia básica e notação

➤ Norma/tamanho de um vetor:

- Se deixarmos $\mathbf{x}$ e $\mathbf{y}$ serem dois vetores n -dimensionais, então seu produto escalar é definido como:

$$(\mathbf{x} \bullet \mathbf{y}) = \mathbf{x}^T \mathbf{y} = \sum_{i=1}^n x_i y_i$$

- $\mathbf{x}$ e $\mathbf{y}$ são ortogonais se $(\mathbf{x} \bullet \mathbf{y}) = 0$. Se os vetores não são ortogonais, o ângulo entre eles pode ser calculado a partir da definição do produto escalar:

$$(\mathbf{x} \bullet \mathbf{y}) = \|\mathbf{x}\| \|\mathbf{y}\| \cos \theta,$$

- $\|\mathbf{x}\|$ representa o comprimento do vetor $\mathbf{x}$ (também chamado de norma do vetor). O comprimento do vetor $\mathbf{x}$ é definido como a raiz quadrada da soma dos quadrados dos componentes:

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n x_i^2} = \sqrt{(\mathbf{x} \bullet \mathbf{x})}$$

1.1. Terminologia básica e notação

➤ Funções de várias variáveis:

- Assim como uma função de uma única variável é representada como $f(x)$, uma função de n variáveis independentes $x_1, x_2, \dots, x_n$ é escrita como:

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$$

- Lidamos com muitas funções de variáveis vetoriais. Para distinguir entre funções, são usados subscripts. Assim, a função i é escrita como:

$$g_i(\mathbf{x}) = g_i(x_1, x_2, \dots, x_n) \quad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_m(\mathbf{x}) \end{bmatrix} = [g_1(\mathbf{x}) \ g_2(\mathbf{x}) \ \dots \ g_m(\mathbf{x})]^T$$

1.1. Terminologia básica e notação

➤ Funções de várias variáveis:

❑ Os métodos apresentados nesse curso são aplicados, apenas, para funções contínuas e pelo menos duas vezes diferenciáveis:

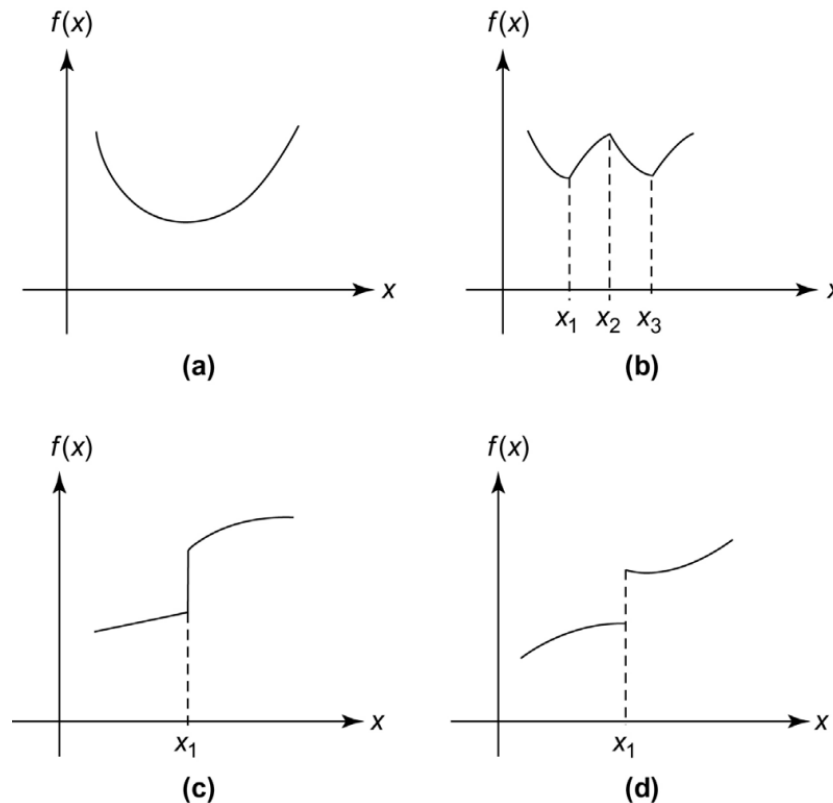


FIGURE 1.5 Continuous and discontinuous functions. (a) and (b) Continuous functions; (c) not a function; and (d) discontinuous function.

1.1. Terminologia básica e notação

➤ Derivadas parciais de funções: 1ª derivada:

□ Para uma função $f(\mathbf{x})$ de n variáveis, as derivadas parciais são escritas como:

$$\frac{\partial f(\mathbf{x})}{\partial x_i}; i = 1 \text{ to } n$$

□ As n derivadas parciais na são usualmente dispostos em um vetor coluna conhecido como **gradiente da função $f(\mathbf{x})$** . O gradiente é escrito como $\partial f / \partial \mathbf{x}$ ou $\nabla f(\mathbf{x})$.

$$\nabla f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} [\partial f(\mathbf{x})] / \partial x_1 \\ [\partial f(\mathbf{x})] / \partial x_2 \\ \vdots \\ [\partial f(\mathbf{x})] / \partial x_n \end{bmatrix}$$

1.1. Terminologia básica e notação

➤ Derivadas parciais de funções: 2ª derivada:

- Cada componente do vetor gradiente na pode ser diferenciado novamente em relação a uma variável para obter as segundas derivadas parciais para a função $f(\mathbf{x})$:

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}; \quad i, j = 1 \text{ to } n$$

- Vemos que existem n^2 derivadas parciais. Estes podem ser organizados em uma matriz conhecida como **matriz Hessiana**, escrita como **$\mathbf{H}(\mathbf{x})$** , ou simplesmente a matriz das segundas derivadas parciais de $f(\mathbf{x})$, escrita como $\nabla^2 f(\mathbf{x})$. **A Hessiana é simétrica.**

$$\mathbf{H}(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \left[\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right]_{n \times n}$$

2. Formulação do Problema de Otimização

2. Formulação do Problema de Otimização

Ao concluir este capítulo, você será capaz de:

1. Traduzir uma **declaração descritiva** do problema em uma **declaração matemática para otimização**.
2. Identificar e definir as **variáveis de projeto** do problema.
3. Identificar e definir um **critério de otimização** para o problema.
4. Identificar e definir os **problemas de restrições** do projeto.
5. Transcreva a formulação do problema em um **modelo padrão** para otimização de projeto.

2. Formulação do Problema de Otimização

Para a maioria dos problemas de otimização, usaremos o seguinte procedimento para formular o problema:

Etapa 1: descrição do projeto/problema;

Etapa 2: coleta de dados e informações;

Etapa 3: definição de variáveis de projeto;

Etapa 4: critério de otimização;

Etapa 5: formulação das restrições.

Exemplo 1: PROJETO DE UMA VIGA ENGASTADA

2. Formulação do Problema de Otimização

Etapa 1: descrição do projeto/problema:

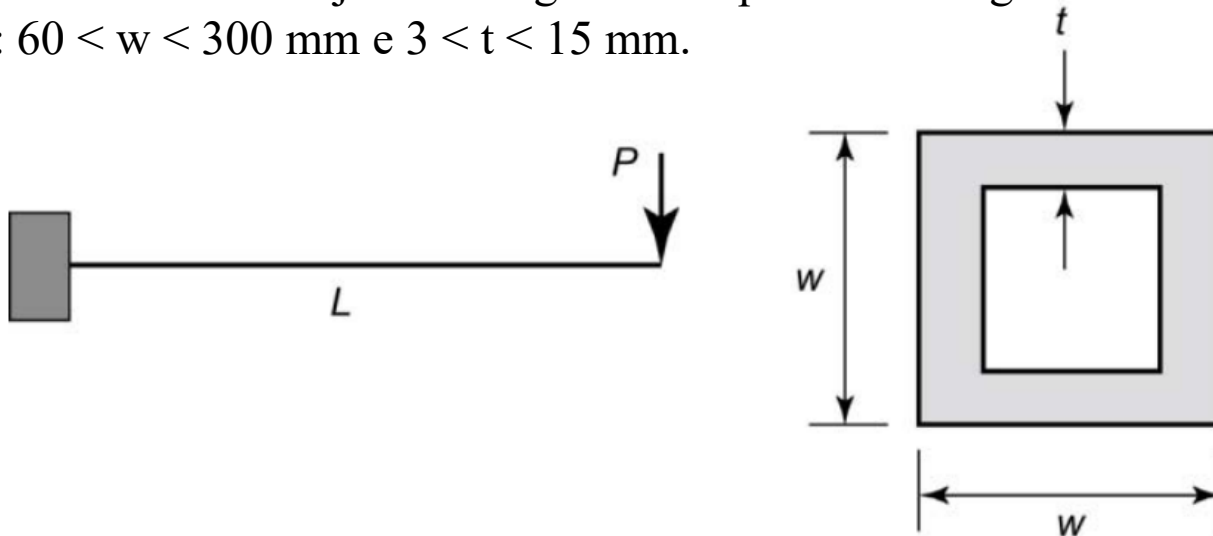
A declaração descreve os objetivos gerais do projeto e os requisitos a serem atendidos. Geralmente é fornecido pelo proprietário/patrocinador do projeto.

2. Formulação do Problema de Otimização

Etapa 1: descrição do projeto/problema:

EXEMPLO 1: PROJETO DE UMA VIGA ENGASTADA:

As vigas engastadas em balanço são usadas em muitas aplicações práticas na engenharia civil, mecânica e aeroespacial. Para ilustrar a etapa de descrição do problema, consideramos o projeto de uma viga em balanço de seção quadrada oca para suportar uma **carga de 20 kN** em sua extremidade. A viga, feita de **aço**, tem **2 m** de comprimento, conforme mostrado na Fig. As condições de ruptura da viga são as seguintes: *(1) o material não deve falhar sob a ação da carga e (2) a deflexão da extremidade livre não deve ser maior que 1 cm*. A relação largura/espessura da viga não deve ser superior a 8 para evitar a flambagem local das paredes. Uma viga de massa mínima é desejada. A largura e a espessura da viga devem estar dentro dos seguintes limites: $60 < w < 300$ mm e $3 < t < 15$ mm.



2. Formulação do Problema de Otimização

Etapa 2: Coleta de Dados e Informações:

Para desenvolver uma formulação matemática para o problema, precisamos reunir informações sobre: propriedades do material, requisitos de desempenho, limites de recursos, custo de matérias-primas e assim por diante. Além disso, a maioria dos problemas requer a capacidade de analisar projetos usando o MEF.

2. Formulação do Problema de Otimização

Etapa 2: Coleta de Dados e Informações:

EXEMPLO 1: COLETA DE DADOS E INFORMAÇÕES DA VIGA :

As informações necessárias para o problema de projeto de vigas em balanço do Exemplo 1 incluem expressões para tensões à flexão e cisalhamento e a expressão para a deflexão da extremidade livre. A notação e os dados para este fim estão definidos na Tabela 2.1:

A seguir estão expressões úteis para a viga:

$$\sigma = \frac{Mw}{2I}, \text{ N/mm}^2$$

$$\tau = \frac{VQ}{2It}, \text{ N/mm}^2$$

$$q = \frac{PL^3}{3EI}, \text{ mm}$$

$$A = w^2 - (w - 2t)^2 = 4t(w - t), \text{ mm}^2$$

$$I = \frac{1}{12}w \times w^3 - \frac{1}{12}(w - 2t) \times (w - 2t)^3 = \frac{1}{12}w^4 - \frac{1}{12}(w - 2t)^4, \text{ mm}^4$$

$$Q = \frac{1}{2}w^2 \times \frac{w}{4} - \frac{1}{2}(w - 2t)^2 \times \frac{(w - 2t)}{4} = \frac{1}{8}w^3 - \frac{1}{8}(w - 2t)^3, \text{ mm}^3$$

$$M = PL, \text{ N/mm}$$

$$V = P, \text{ N}$$

2. Formulação do Problema de Otimização

Etapa 2: Coleta de Dados e Informações:

EXEMPLO 1: COLETA DE DADOS E INFORMAÇÕES DA VIGA :

TABLE 2.1 Notation and Data for Cantilever Beam

Notation	Data
A	Cross-sectional area, mm^2
E	Modulus of elasticity of steel, $21 \times 10^4 \text{ N/mm}^2$
G	Shear modulus of steel, $8 \times 10^4 \text{ N/mm}^2$
I	Moment of inertia of the cross-section, mm^4
L	Length of the member, 2000 mm
M	Bending moment, N/mm
P	Load at the free end, 20,000 N
Q	Moment about the neutral axis of the area above the neutral axis, mm^3
q	Vertical deflection of the free end, mm
q_a	Allowable vertical deflection of the free end, 10 mm
V	Shear force, N
w	Width (depth) of the section, mm
t	Wall thickness, mm
σ	Bending stress, N/mm^2
σ_a	Allowable bending stress, 165 N/mm^2
τ	Shear stress, N/mm^2
τ_a	Allowable shear stress, 90 N/mm^2

2. Formulação do Problema de Otimização

Etapa 3: Definição de Variáveis de Projeto:

Nesse passo identifica-se um conjunto de variáveis que descrevem o sistema, chamadas de variáveis de projeto. Se as variáveis de projeto adequadas não forem selecionadas para um problema, a formulação será incorreta ou não será possível.

2. Formulação do Problema de Otimização

Etapa 3: Definição de Variáveis de Projeto:

- Geralmente, as variáveis de projeto devem ser independentes umas das outras. Se não forem, deve haver algumas restrições de igualdade entre elas (explicadas posteriormente em vários exemplos).
- Um número mínimo de variáveis de projeto é necessário para formular adequadamente um problema de otimização de projeto.
- O maior número possível de parâmetros independentes deve ser designado como variáveis de projeto na fase de formulação do problema. Mais tarde, algumas dessas variáveis podem receber valores numéricos fixos.
- Um valor numérico deve ser dado a cada variável de projeto identificada para determinar se um projeto de teste do sistema foi especificado.

2. Formulação do Problema de Otimização

Etapa 3: Definição de Variáveis de Projeto:

EXEMPLO 1: VARIÁVEIS DE PROJETO DO PROBLEMA :

Para demonstrar o cálculo de várias grandezas de análise, vamos selecionar um projeto de teste como $w = 60 \text{ mm}$ e $t = 10 \text{ mm}$ e calcular as grandezas definidas nas Equações:

$$A = 4t(w - t) = 4(10)(60 - 10) = 2,000 \text{ mm}^2$$

$$I = \frac{1}{12}w^4 - \frac{1}{12}(w - 2t)^4 = \frac{1}{12}(60)^4 - \frac{1}{12}(60 - 2 \times 10)^4 = 866,667 \text{ mm}^4$$

$$Q = \frac{1}{8}w^3 - \frac{1}{8}(w - 2t)^3 = \frac{1}{8}(60)^3 - \frac{1}{8}(60 - 2 \times 10)^3 = 19,000 \text{ mm}^3$$

$$M = PL = 20,000 \times 2,000 = 4 \times 10^7 \text{ N/mm}$$

$$V = P = 20,000 \text{ N}$$

$$\sigma = \frac{Mw}{2I} = \frac{4 \times 10^7 (60)}{2 \times 866,667} = 1,385 \text{ N/mm}^2$$

$$\tau = \frac{VQ}{2It} = \frac{20,000 \times 19,000}{2 \times 866,667 \times 10} = 21.93 \text{ N/mm}^2$$

$$q = \frac{PL^3}{3EI} = \frac{20,000 \times (2,000)^3}{3 \times 21 \times 10^4 \times 866,667} = 262.73 \text{ mm}$$

2. Formulação do Problema de Otimização

Etapa 3: Definição de Variáveis de Projeto:

EXEMPLO 1: VARIÁVEIS DE PROJETO DO PROBLEMA :

Apenas as dimensões da seção transversal são identificadas como variáveis de projeto para o problema de projeto de vigas em balanço do Exemplo 1; todos os outros parâmetros são especificados. Observe que as variáveis de projeto são definidas com precisão, incluindo as unidades a serem usadas para elas.

w = largura externa (profundidade) da seção, mm

t = espessura da parede, mm

2. Formulação do Problema de Otimização

Etapa 4: Critério de Otimização:

O critério deve ser uma função escalar cujo valor numérico possa ser obtido uma vez especificado um projeto; ou seja, deve ser uma função do vetor variável de projeto x . Tal critério é geralmente **chamado de função objetivo/custo (a ser minimizado)/lucro (a ser maximizado)** do problema.

Em algumas situações, duas ou mais funções objetivo podem ser identificadas. Por exemplo, podemos querer minimizar o peso de uma estrutura e ao mesmo tempo minimizar a deflexão ou tensão em um determinado ponto. Estes são chamados de problemas de otimização de projeto multiobjetivo e serão discutidos mais a frente.

2. Formulação do Problema de Otimização

Etapa 4: Critério de Otimização:

EXEMPLO 1: CRITÉRIO DE OTIMIZAÇÃO PARA VIGA:

Para o problema, o objetivo é projetar uma viga em balanço de massa mínima. Como a massa é proporcional à área da seção transversal da viga, a função objetivo para o problema é tomada como a área da seção transversal que deve ser minimizada:

$$f(w, t) = A = 4t(w - t), \text{ mm}^2$$

No projeto experimental $w = 60 \text{ mm}$ e $t = 10 \text{ mm}$, a função de custo é avaliada como

$$f(w, t) = 4t(w - t) = 4 \times 10(60 - 10) = 2,000 \text{ mm}^2$$

2. Formulação do Problema de Otimização

Etapa 5: Formulação das Restrições:

Por exemplo, os membros estruturais não devem falhar sob cargas operacionais normais. Os membros devem caber no espaço disponível, e assim por diante. Essas restrições devem depender das variáveis de projeto, pois somente assim seus valores mudam com diferentes projetos de ensaio.

2. Formulação do Problema de Otimização

Etapa 5: Formulação das Restrições:

Restrições lineares e não lineares: Muitas funções de restrição têm apenas termos de primeira ordem (lineares) em variáveis de projeto. Estas são chamadas de restrições lineares. Problemas mais gerais têm função objetivo e/ou funções de restrição não linear. Estes são chamados de problemas de programação não linear.

Restrições de Igualdade e Desigualdade: Os problemas podem ter restrições de igualdade ($=$), bem como de desigualdade ($<$, $>$, $\leq$ ou $\geq$).

2. Formulação do Problema de Otimização

Etapa 5: Formulação das Restrições:

EXEMPLO 1: RESTRIÇÕES PARA VIGA:

Usando várias expressões dadas nas Equações, formulamos as restrições para o problema de projeto de vigas em balanço do Exemplo 1 como segue:

$$\sigma \leq \sigma_a$$

$$\frac{PLw}{2I} \leq \sigma_a$$

$$\tau \leq \tau_a$$

$$\frac{PQ}{2It} \leq \tau_a$$

$$q \leq q_a$$

$$\frac{PL^3}{3EI} \leq q_a$$

$$\frac{w}{t} \leq 8$$

$$w \leq 8t$$

$$60 \leq w, \text{ mm}; \quad w \leq 300, \text{ mm}$$

$$3 \leq t, \text{ mm}; \quad t \leq 15, \text{ mm}$$

2. Formulação do Problema de Otimização

Etapa 5: Formulação das Restrições:

EXEMPLO 1: RESTRIÇÕES PARA VIGA:

o problema de otimização é encontrar w e t para minimizar a função de custo sujeita às oito restrições de desigualdade. Vamos verificar o estado das restrições para o problema de projeto de vigas em balanço no ponto de projeto de teste $w = 60 \text{ mm}$ e $t = 10 \text{ mm}$:

Bending stress constraint: $\sigma \leq \sigma_a$; $\sigma = 1385 \text{ N/mm}^2$, $\sigma_a = 165 \text{ N/mm}^2$; $\therefore$ violated

Shear stress constraint: $\tau \leq \tau_a$; $\tau = 21.93 \text{ N/mm}^2$, $\tau_a = 90 \text{ N/mm}^2$; $\therefore$ satisfied

Deflection constraint: $q \leq q_a$; $q = 262.73 \text{ mm}$, $q_a = 10 \text{ mm}$; $\therefore$ violated

Width–thickness restriction: $\frac{w}{t} \leq 8$; $\frac{w}{t} = \frac{60}{10} = 6$; $\therefore$ satisfied

Este projeto teste viola as restrições de tensão e deflexão, portanto, não é viável!!! Logo deve-se mudar os valores.

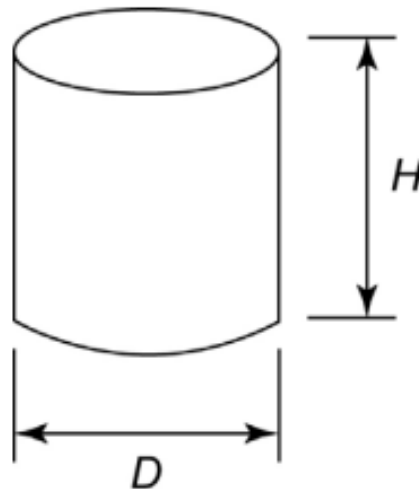
Exemplo 2: PROJETO DE UMA LATA

2. Formulação do Problema de Otimização

Etapa 1: descrição do projeto/problema:

EXEMPLO 2: PROJETO DE UMA LATA:

O objetivo deste problema é projetar uma lata, para conter pelo menos 400 mL de líquido ($1 \text{ mL} = 1 \text{ cm}^3$), bem como atender a outros requisitos. As latas serão produzidas na casa dos bilhões, por isso é desejável minimizar seus custos de fabricação. Como o custo pode estar diretamente relacionado à área da superfície da chapa utilizada, é razoável minimizar a quantidade de chapa metálica necessária. As considerações de fabricação, manuseio, estética e transporte impõem as seguintes restrições ao tamanho da lata: O diâmetro não deve ser superior a 8 cm e não inferior a 3,5 cm, enquanto a altura não deve ser superior a 18 cm e não inferior a 8 cm.



2. Formulação do Problema de Otimização

Etapa 2: Coleta de dados e informações:

Os dados para o problema são fornecidos na declaração do projeto.

Passo 3: Definição da variável de projeto:

As duas variáveis de projeto são definidas como:

D = diâmetro da lata, cm

H = altura da lata, cm

2. Formulação do Problema de Otimização

Etapa 4: Critério de otimização:

O objetivo do projeto é minimizar a área de superfície total S da chapa metálica para as três partes da lata cilíndrica: a área de superfície do cilindro (circunferência $\times$ altura) e a área de superfície das duas extremidades. Portanto, o critério de otimização, ou função objetivo/de custo (a área total da chapa metálica), é dado como.

$$S = \pi DH + 2\left(\frac{\pi}{4}D^2\right), \text{ cm}^2$$

2. Formulação do Problema de Otimização

Passo 5: Formulação das restrições:

A primeira restrição é que a lata deve conter pelo menos 400 cm³ de fluido. E as outras restrições sobre o tamanho da lata são:

$$\frac{\pi}{4} D^2 H \geq 400, \text{ cm}^3$$

$$3.5 \leq D \leq 8, \text{ cm}$$

$$8 \leq H \leq 18, \text{ cm}$$

O problema de otimização da lata consiste em determinar as variáveis de projeto D e H para minimizar a função custo sujeita às cinco desigualdades nas Equações acima.

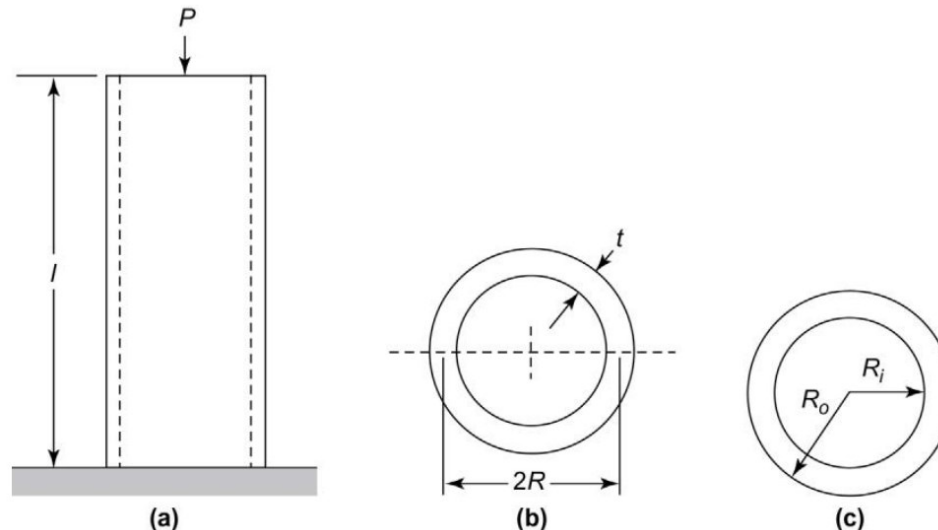
Exemplo 3: PROJETO DE COLUNA TUBULAR DE PESO MÍNIMO

2. Formulação do Problema de Otimização

Etapa 1: descrição do projeto/problema:

EXEMPLO 3: PROJETO DE COLUNA TUBULAR DE PESO MÍNIMO:

Colunas retas são usadas como elementos estruturais em estruturas civis, mecânicas, aeroespaciais, agrícolas e automotivas. Muitas dessas aplicações podem ser observadas na vida cotidiana, por exemplo, um poste de luz de rua, um poste de semáforo, um mastro de bandeira, um suporte de torre de água, um poste de sinalização de rodovia, um poste de transmissão de energia. É importante otimizar o projeto de uma coluna reta, pois ela pode ser produzida em massa. O objetivo deste projeto é projetar um pilar tubular de massa mínima de comprimento l suportando uma carga P sem flambagem ou sobretensão. A coluna é fixada na base e livre no topo, conforme mostrado na Fig.



2. Formulação do Problema de Otimização

Etapa 2: Coleta de dados e informações:

A carga de flambagem (também chamada de carga crítica) para uma coluna em balanço é dada como:

$$P_{cr} = \frac{\pi^2 EI}{4l^2}$$

Passo 3: Definição da variável de projeto:

R = raio médio da coluna

t = espessura da parede

Assumindo que a parede do pilar é fina ($R \gg t$), a área da seção transversal do material e o momento de inércia são

$$A = 2\pi R t; I = \pi R^3 t$$

2. Formulação do Problema de Otimização

Etapa 4: Critério de otimização:

A massa total da coluna a ser minimizada é dada como.

$$Mass = \rho(lA) = 2\rho l\pi Rt$$

Passo 5: Formulação das restrições:

A primeira restrição é que a tensão (P/A) não deve exceder a tensão admissível do material σ_a para evitar a falha do material. Isto é expresso como a igualdade $\sigma \leq \sigma_a$. Substituindo σ por P/A e depois substituindo por A , obtemos:

$$\frac{P}{2\pi Rt} \leq \sigma_a$$

2. Formulação do Problema de Otimização

Passo 5: Formulação das restrições:

A coluna não deve flambar sob a carga aplicada P , o que implica que a carga aplicada não deve exceder a carga de flambagem ($P \leq P_{cr}$).

$$P \leq \frac{\pi^3 E R^3 t}{4l^2}$$

Por fim, as variáveis de projeto R e t devem estar dentro dos valores mínimos e máximos especificados:

$$R_{\min} \leq R \leq R_{\max} ; t_{\min} \leq t \leq t_{\max}$$

Formulação matemática: O problema de otimização é determinar as variáveis de projeto R e t para minimizar a função de custo sujeita a seis restrições de desigualdade.

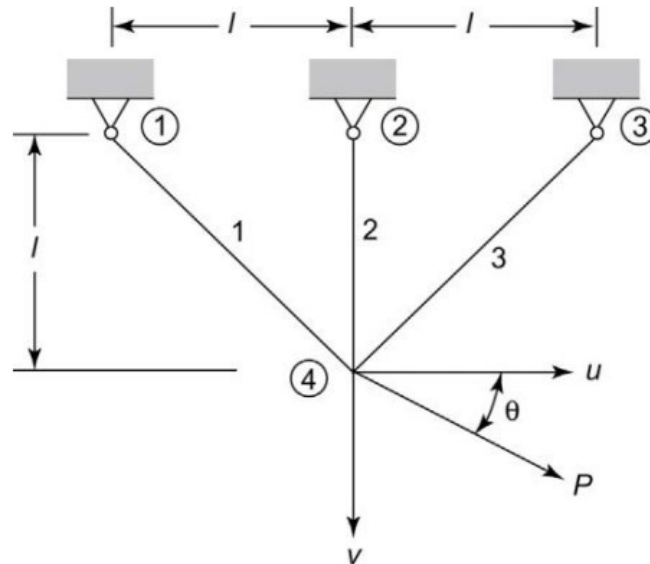
Exemplo 4: PROJETO DE PESO MÍNIMO DE UMA TRELIÇA DE TRÊS BARRAS

2. Formulação do Problema de Otimização

Etapa 1: descrição do projeto/problema:

EXEMPLO 4: PROJETO DE PESO MÍNIMO DE UMA TRELIÇA DE TRÊS BARRAS:

Considere a estrutura de três barras mostrada na Figura. Esta é uma estrutura estaticamente indeterminada para a qual as forças do membro não podem ser calculadas exclusivamente de equações de equilíbrio. A estrutura deve ser projetada para volume mínimo (ou, equivalentemente, massa mínima) para suportar uma força P . Ela deve satisfazer várias restrições de desempenho e tecnológicas, como esmagamento de membro, flambagem de membro e falha por deflexão excessiva do nó 4.



2. Formulação do Problema de Otimização

Etapa 2: Coleta de dados e informações:

Como a estrutura deve ser simétrica, os membros 1 e 3 terão a mesma área de seção transversal, digamos A_1 . Seja A_2 a área da seção transversal do membro 2. Usando procedimentos de análise para estruturas estaticamente indeterminadas, os deslocamentos horizontais e verticais u e v do nó 4 são calculados como:

$$u = \frac{\sqrt{2}lP_u}{A_1E}; \quad v = \frac{\sqrt{2}lP_v}{(A_1 + \sqrt{2}A_2)E}$$

Então as tensões σ_1 , σ_2 e σ_3 nos membros 1, 2 e 3 sob a carga aplicada P podem ser calculadas a partir das forças do membro como (tensão = força/área; $\sigma_i = F_i/A_i$):

$$\sigma_1 = \frac{1}{\sqrt{2}} \left[\frac{P_u}{A_1} + \frac{P_v}{A_1 + \sqrt{2}A_2} \right] \quad \sigma_2 = \frac{\sqrt{2}P_v}{(A_1 + \sqrt{2}A_2)} \quad \sigma_3 = \frac{1}{\sqrt{2}} \left[-\frac{P_u}{A_1} + \frac{P_v}{A_1 + \sqrt{2}A_2} \right]$$

2. Formulação do Problema de Otimização

Etapa 3: Definição da variável de projeto:

A_1 = área da seção transversal do material para os membros 1 e 3;

A_2 = área da seção transversal do material para o membro 2.

Etapa 4: Critério de otimização :

O peso total de três membros da treliça serve como uma função de custo (peso de um membro = área da seção transversal $\times$ comprimento $\times$ densidade de peso):

$$Volume = l\gamma(2\sqrt{2}A_1 + A_2)$$

2. Formulação do Problema de Otimização

Passo 5: Formulação das restrições:

$$\sigma_1 \leq \sigma_a ; \sigma_2 \leq \sigma_a$$

a tensão no membro 3 pode ser positiva (tração) ou negativa (compressão) dependendo do ângulo de carga. Portanto, ambas as possibilidades precisam ser consideradas na formulação da restrição de tensão para o membro 3:

$$\text{IF } (\sigma_3 < 0) \text{ THEN } -\sigma_3 \leq \sigma_a \text{ ELSE } \sigma_3 \leq \sigma_a$$

As deflexões horizontais e verticais do nó 4 devem estar dentro dos limites especificados Δu e Δv , respectivamente. Usando a Eq. (a), as restrições de deflexão são:

$$u \leq \Delta_u ; v \leq \Delta_v$$

A maioria dos problemas de projeto práticos exige que cada membro tenha uma certa área mínima, $A_{\min}$.

$$A_1, A_2 \geq A_{\min}$$

MODELO MATEMÁTICO GERAL PARA O PROJETO ÓTIMO

MODELO MATEMÁTICO GERAL

Podemos definir um **modelo matemático geral/padrão** para um projeto ótimo para abranger todas essas possibilidades.

MODELO MATEMÁTICO GERAL

Podemos definir um modelo matemático geral/padrão para um projeto ótimo para abranger todas essas possibilidades.

Find an n -vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ of design variables
to *minimize a cost function*

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$$

subject to the p equality constraints

$$h_j(\mathbf{x}) = h_j(x_1, x_2, \dots, x_n) = 0; \quad j = 1 \text{ to } p$$

and the m inequality constraints

$$g_i(\mathbf{x}) = g_i(x_1, x_2, \dots, x_n) \leq 0; \quad i = 1 \text{ to } m$$

MODELO MATEMÁTICO GERAL

O modelo de **PROJETO GERAL** trata apenas de problemas de **MINIMIZAÇÃO**. Isso não é uma restrição, pois a maximização de uma função $F(\mathbf{x})$ é o mesmo que a minimização de uma função transformada $f(\mathbf{x}) = -F(\mathbf{x})$.

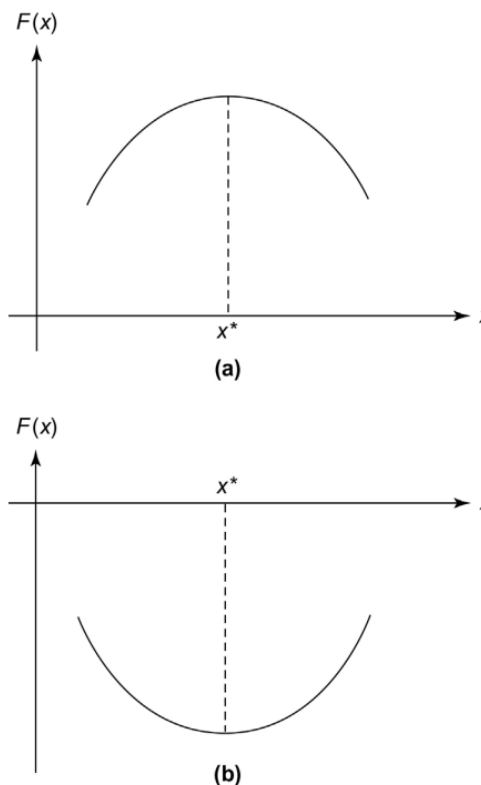


FIGURE 2.10 Point maximizing $F(x)$ equals point minimizing $-F(x)$. (a) Plot of $F(x)$; (b) plot of $f(x) = -F(x)$.

MODELO MATEMÁTICO GERAL

Problemas de otimização de projeto de diferentes áreas da engenharia podem ser transcritos no modelo padrão.

Uma vez que os problemas de diferentes campos foram transcritos em declarações matemáticas usando uma notação padrão, eles têm a mesma forma matemática.

Os conceitos e métodos de otimização descritos no texto são bastante gerais e podem ser usados para resolver problemas de diversas áreas.

MODELO MATEMÁTICO GERAL

O modelo de otimização de projeto padrão trata **apenas restrições de desigualdade “ $\leq$ ”**. Muitos problemas de projeto também podem ter desigualdades “ $\geq$ ”.

Essas restrições podem ser convertidas na forma padrão sem muita dificuldade. A restrição “ $\geq$ ” $G_j(\mathbf{x}) \geq 0$ é equivalente à desigualdade “ $\leq$ ” $g_j(\mathbf{x}) = -G_j(\mathbf{x}) \leq 0$.

Portanto, podemos multiplicar qualquer restrição “ $\geq$ ” por -1 para convertê-lo em um “ $\leq$ ”.

Observações importantes sobre o modelo padrão

MODELO MATEMÁTICO GERAL

1. As funções $f(\mathbf{x})$, $h_j(\mathbf{x})$ e $g_i(\mathbf{x})$ devem depender, explícita ou implicitamente, de algumas das variáveis de projeto. Funções que não dependem de nenhuma variável podem ser ignoradas com segurança.
2. O número de restrições de igualdade independentes deve ser menor ou no máximo igual ao número de variáveis de projeto (ou seja, $p \leq n$).
3. Não há limitação no número de restrições de desigualdade.

MODELO MATEMÁTICO GERAL

4. Alguns problemas de projeto podem não ter restrições. Estes são chamados de irrestritos;
5. Problemas de programação linear: Se todas as funções $f(\mathbf{x})$, $h_j(\mathbf{x})$ e $g_i(\mathbf{x})$ são lineares nas variáveis de projeto $\mathbf{x}$, então o problema é chamado de problema de programação linear. Se alguma dessas funções for não linear, o problema é chamado de problema de programação não linear.
6. É importante notar que se a função de custo for escalada multiplicando-a por uma constante positiva, o projeto ótimo não muda. No entanto, o valor ótimo da função de custo muda.

MODELO MATEMÁTICO GERAL

7. Muitas vezes, algumas variáveis são necessárias ter valores discretos ou inteiros. Tais variáveis aparecem com bastante frequência em problemas de projeto de engenharia. Por exemplo, uma espessura de chapa deve ser aquela que está disponível comercialmente.

3. Método de solução gráfica e conceitos básicos de otimização

3. Solução Gráfica e Conceitos Básicos

Ao concluir este capítulo, você será capaz de:

1. **Traçar restrições** de desigualdade e identificar seu lado viável/inviável
2. **Traçar contornos da função objetivo** através da região viável
3. **Localizar graficamente a solução ideal** para um problema e identifique restrições ativas/inativas
4. **Identificar problemas que podem ter soluções múltiplas**, ilimitadas ou inviáveis
5. **Entender conceitos e termos básicos** associados à abordagem de projeto ideal.

3.1 PROCESSO DE SOLUÇÃO GRÁFICA

Problemas de otimização com apenas duas variáveis de projeto podem ser resolvidos observando como elas são representadas graficamente:

Vamos usar um Problema de Maximização de Lucro como exemplo:

3.1 PROCESSO DE SOLUÇÃO GRÁFICA

EXEMPLO:

Etapa 1: Descrição - Uma empresa fabrica duas máquinas, **A** e **B**. Usando os recursos disponíveis, pode-se fabricar até **28 A** ou até **14 B**, diariamente. O departamento de vendas pode vender até **14 máquinas A** ou **24 máquinas B**. A equipe de transporte não pode lidar com mais de **16 máquinas** por dia. A empresa obtém um lucro de **\$ 400 em cada máquina A** e **\$ 600 em cada máquina B**. Quantas máquinas **A** e **B** a empresa deve fabricar todos os dias para maximizar seu lucro?

Etapa 2: Coleta de dados e informações. Dados e informações são definidos na declaração do projeto. Nenhuma informação adicional é necessária.

Etapa 3: Definição das variáveis de projeto. As duas variáveis de projeto a seguir são identificadas na declaração do problema:

x_1 = nº de máquinas **A** fabricadas a cada dia | x_2 = nº de máquinas **B** fabricadas a cada dia

Etapa 4: Critério de otimização. O objetivo é maximizar o lucro diário, que pode ser expresso em termos de variáveis de projeto usando os dados fornecidos na etapa 1 como.

$$P = 400x_1 + 600x_2, \$$$

3.1 PROCESSO DE SOLUÇÃO GRÁFICA

EXEMPLO:

Passo 5: Formulação de restrições. As restrições de projeto são colocadas na capacidade de fabricação, no pessoal de vendas e nas capacidade de transporte e manuseio. A restrição na facilidade de transporte e manuseio é bastante direta:

$$x_1 + x_2 \leq 16$$

Primeiro, considere a limitação de fabricação. Assume-se que se a empresa fabrica x_1 **A** máquinas por dia, então os recursos e equipamentos restantes podem ser usados proporcionalmente para fabricar x_2 **B** máquinas, e vice-versa. Portanto, observando que $x_1/28$ é a fração de recursos usada para produzir **A** e $x_2/14$ é a fração usada para produzir **B**, a restrição é expressa como:

$$\frac{x_1}{28} + \frac{x_2}{14} \leq 1$$

Da mesma forma, a restrição de recursos do departamento de vendas é dada como:

$$\frac{x_1}{14} + \frac{x_2}{24} \leq 1$$

Finalmente, as variáveis de projeto devem ser não negativas como:

$$x_1, x_2 \geq 0$$

3.1 PROCESSO DE SOLUÇÃO GRÁFICA

EXEMPLO:

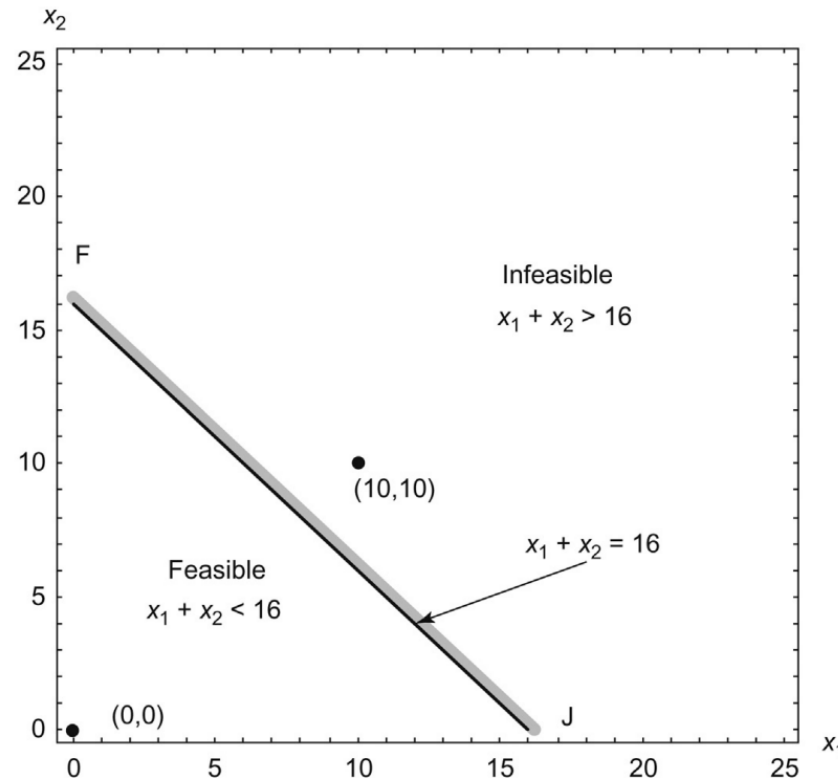
Formulação matemática: O problema de otimização é encontrar duas variáveis de projeto x_1 e x_2 para maximizar a função lucro sujeito a cinco restrições de desigualdade. Todas as funções do problema são lineares nas variáveis x_1 e x_2 . Portanto, é um problema de programação linear. Observe também que para uma solução significativa, ambas as variáveis de projeto devem ter valores inteiros no ponto ótimo.

3.1 PROCESSO DE SOLUÇÃO GRÁFICA

PASSO a PASSO:

Passo 1: O primeiro passo no processo de solução é estabelecer uma origem para o sistema de coordenadas x - y com escalas adequadas.

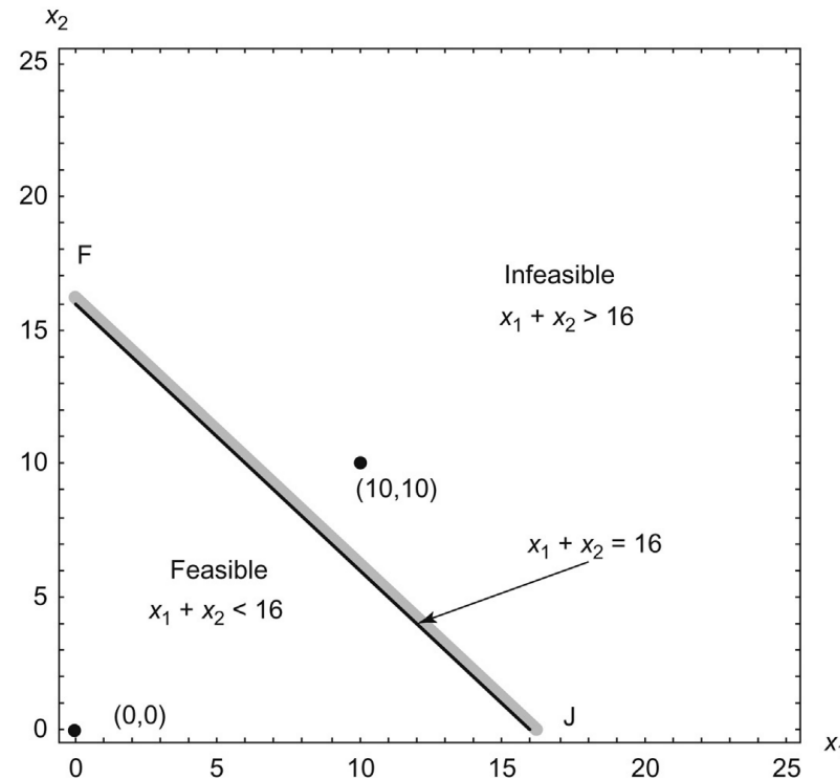
Passo 2: Gráfico de limite de restrição de desigualdade. Para ilustrar o gráfico de uma restrição, vamos considerar a desigualdade $x_1 + x_2 \leq 16$. Para traçar tal função, precisamos de dois pontos que satisfaçam a equação $x_1 + x_2 = 16$. Sejam esses pontos calculados como $(16,0)$ e $(0,16)$.



3.1 PROCESSO DE SOLUÇÃO GRÁFICA

PASSO a PASSO:

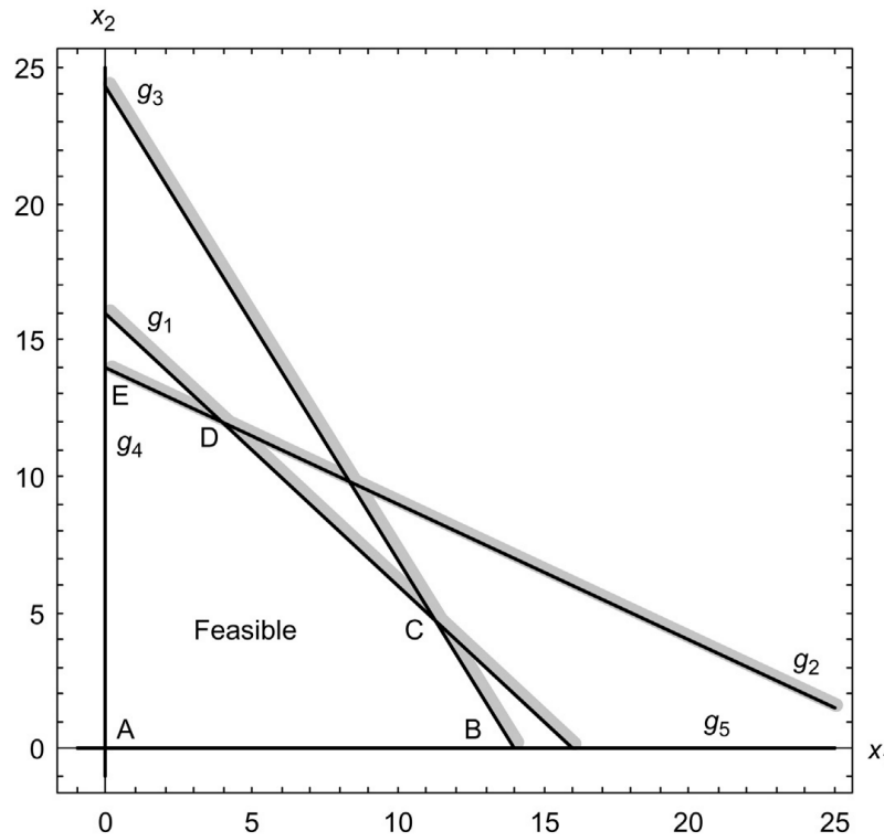
Passo 3: A próxima tarefa é determinar qual lado do limite de restrição F–J é viável para a restrição $x_1 + x_2 \leq 16$. Para isso, selecionamos um ponto em cada lado de F–J e avaliamos a função de restrição ali.



3.1 PROCESSO DE SOLUÇÃO GRÁFICA

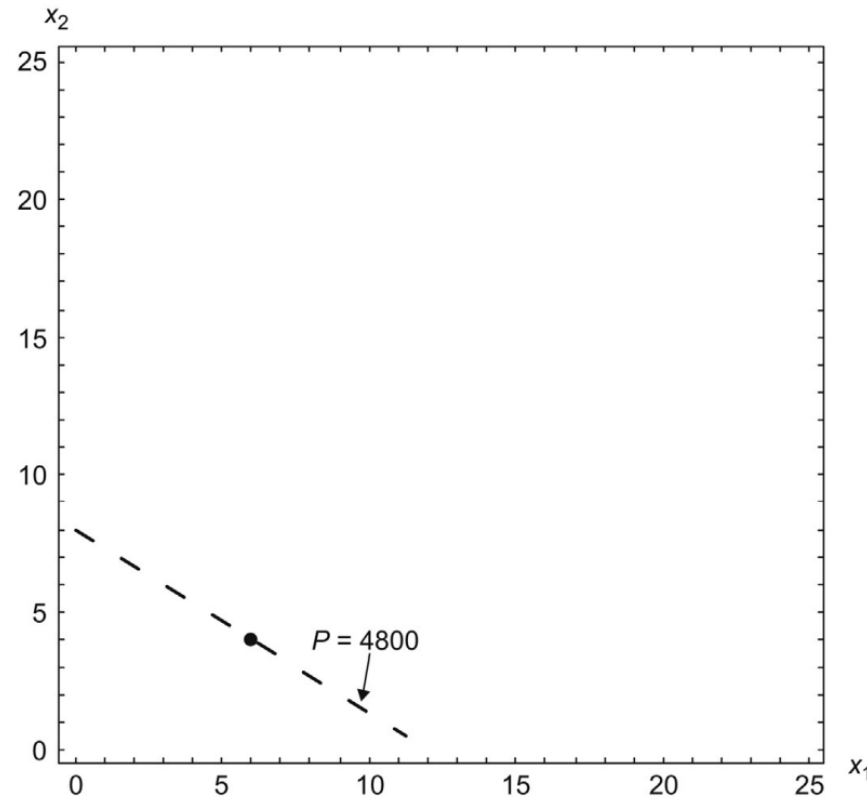
PASSO a PASSO:

Etapa 4: Identificação da região viável. Todas as desigualdades são plotadas no gráfico e o lado viável de cada uma é identificado. A intersecção das regiões factíveis para todas as restrições fornece a região factível para o problema de maximização do lucro, indicada como **ABCDE**. Qualquer ponto nesta região ou em sua fronteira fornece uma solução viável para o problema.



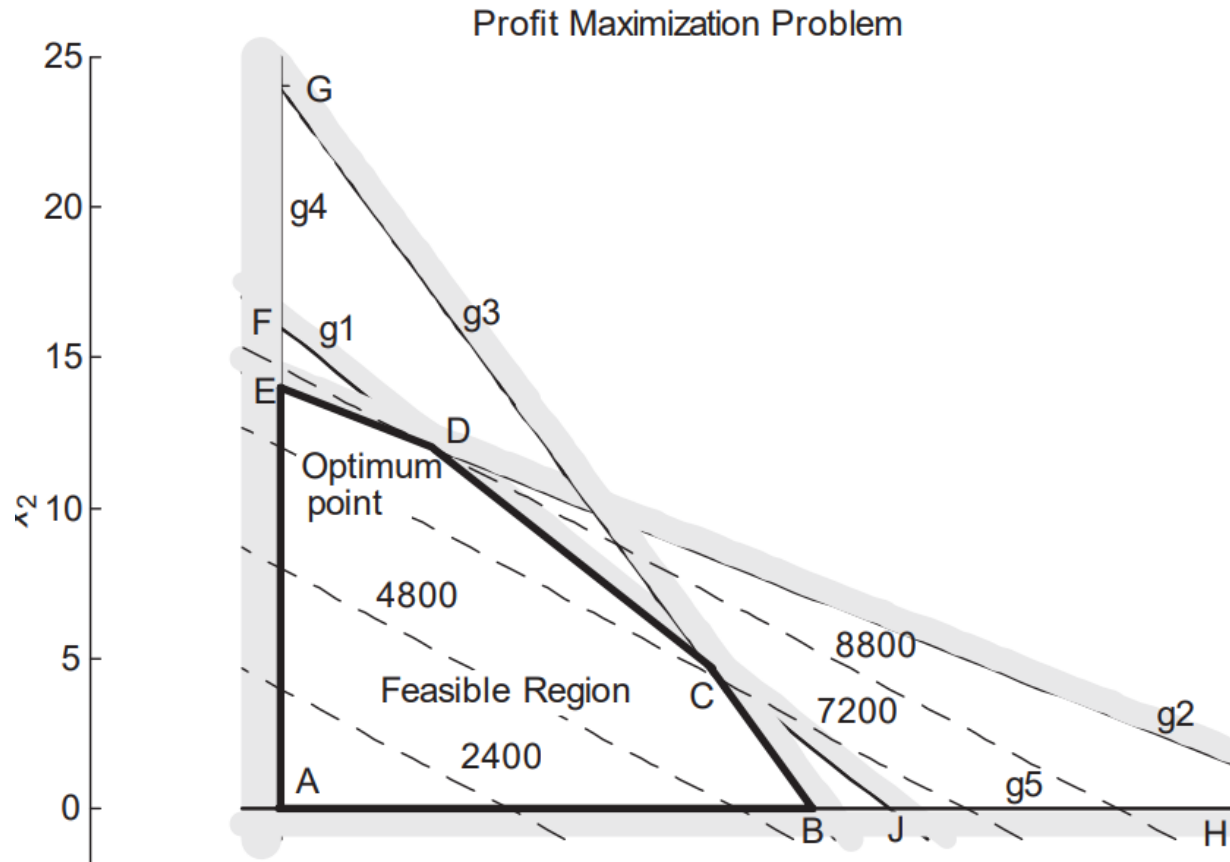
3.1 PROCESSO DE SOLUÇÃO GRÁFICA

Passo 5: A próxima tarefa é traçar a função objetivo no gráfico e localizar seus pontos ótimos. Para o presente problema, o objetivo é maximizar o lucro $P = 400x_1 + 600x_2$, que envolve três variáveis: P , x_1 e x_2 . No entanto, como existem infinitos pontos viáveis, não é possível avaliar a função objetivo em todos os pontos. Uma maneira de superar esse impasse é traçar os contornos da função objetivo. Para traçar um contorno da função objetivo através da região viável, selecione um ponto conveniente no conjunto viável. Avalie a função objetivo no ponto selecionado. Atribua isso como o valor do contorno e plote a equação da função objetivo.



3.1 PROCESSO DE SOLUÇÃO GRÁFICA

Passo 6: Identificação da solução ótima. Contornos para $P = 2400, 4800$ e 7200 são plotados na Figura. Observamos agora a seguinte tendência: À medida que os contornos se movem em direção ao ponto **D**, projetos viáveis podem ser encontrados com valores maiores para P . Vê-se a partir desta observação que o ponto **D** tem o maior valor para P na região viável. Agora simplesmente lemos as coordenadas do ponto **D** (**4;12**) para obter o projeto ótimo, tendo um valor máximo para a função lucro como $P = 8800$.



3.1 PROCESSO DE SOLUÇÃO GRÁFICA

```
clear all
clc
%Create a grid from -1 to 25 with an increment of 0.01 for the variables x1 and x2
[x1,x2]=meshgrid(-1:0.01:25, -1:0.01:25);
%Enter functions for the profit maximization problem
f=400*x1+600*x2;
g1=x1+x2-16;
g2=x1./28+x2./14-1;
g3=x1./14+x2./24-1;
g4=-x1;
g5=-x2;
%Initialization statements; these need not end with a semicolon
cla reset
axis auto %Minimum and maximum values for axes are determined automatically
%Limits for x- and y-axes may also be specified with the command
%axis ([xmin xmax ymin ymax])
xlabel('x1 - Número de Máquinas A'),ylabel('x2 - x1 - Número de Máquinas B') %Specifies labels for x- and y-axes
title('Problema de maximização de lucro') %Displays a title for the problem
hold on %retains the current plot and axes properties for all subsequent plots
%Use the "contour" command to plot constraint and cost functions
cv1=[0 0]; %Specifies two contour values
const1=contour(x1,x2,g1,cv1,'k', 'LineWidth',3); %Plots two specified contours of g1; k = black color
%clabel(const1) %Automatically puts the contour value on the graph
text(2,16,'g1') %Writes g1 at the location (1, 16)
cv11=[0.15:0.02:0.9];
const1=contour(x1,x2,g1,cv11,'r');
const2=contour(x1,x2,g2,cv1,'k','LineWidth',3);
text(15.5,5.4,'g2') %Writes g1 at the location (1, 16)
cv21=[0.01:0.002:0.05];
const2=contour(x1,x2,g2,cv21,'r');
const3=contour(x1,x2,g3,cv1,'k','LineWidth',3);
const3=contour(x1,x2,g3,cv21,'r');
text(2,23,'g3')
cv41=[0.1:0.01:.6];
const4=contour(x1,x2,g4,cv1,'k','LineWidth',3);
const4=contour(x1,x2,g4,cv41,'r');
text(.25,20,'g4')
const5=contour(x1,x2,g5,cv1,'k','LineWidth',3);
const5=contour(x1,x2,g5,cv41,'r');
text(19,1,'g5')
text(2,5,'Região viável')
fv=[2800:2000:8800]; %Defines 4 contours for the profit function
fs=contour(x1,x2,f,fv,'b--','LineWidth',1);
xlim([-5 25])
ylim([-5 25])
hold off %Indicates end of this plotting sequence %Subsequent plots will appear in separate windows
```

3.1 PROCESSO DE SOLUÇÃO GRÁFICA

- PROBLEMA DE PROJETO COM MÚLTIPLAS SOLUÇÕES:**

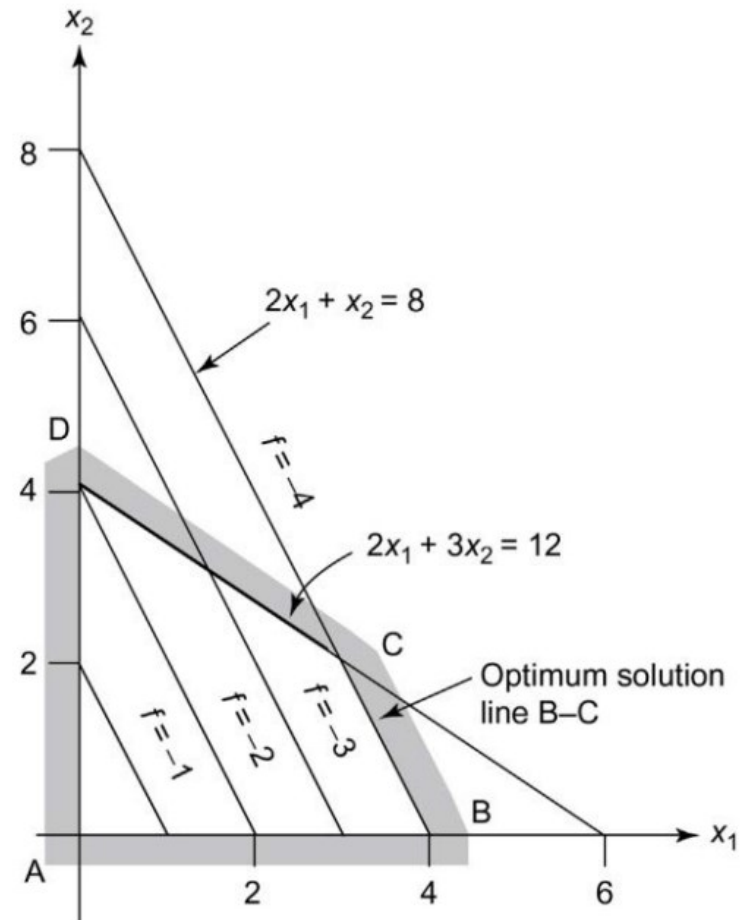
Pode surgir uma situação em que uma restrição seja paralela à função de custo. Nesse caso, existem várias soluções para o problema. Vê-se que qualquer ponto na linha B-C fornece um projeto ótimo, dando ao problema infinitas soluções ótimas.

Minimize

$$f(x) = -x_1 - 0.5x_2$$

subject to

$$2x_1 + 3x_2 \leq 12, 2x_1 + x_2 \leq 8, -x_1 \leq 0, -x_2 \leq 0$$



3.1 PROCESSO DE SOLUÇÃO GRÁFICA

- PROBLEMA COM SOLUÇÕES ILIMITADAS:**

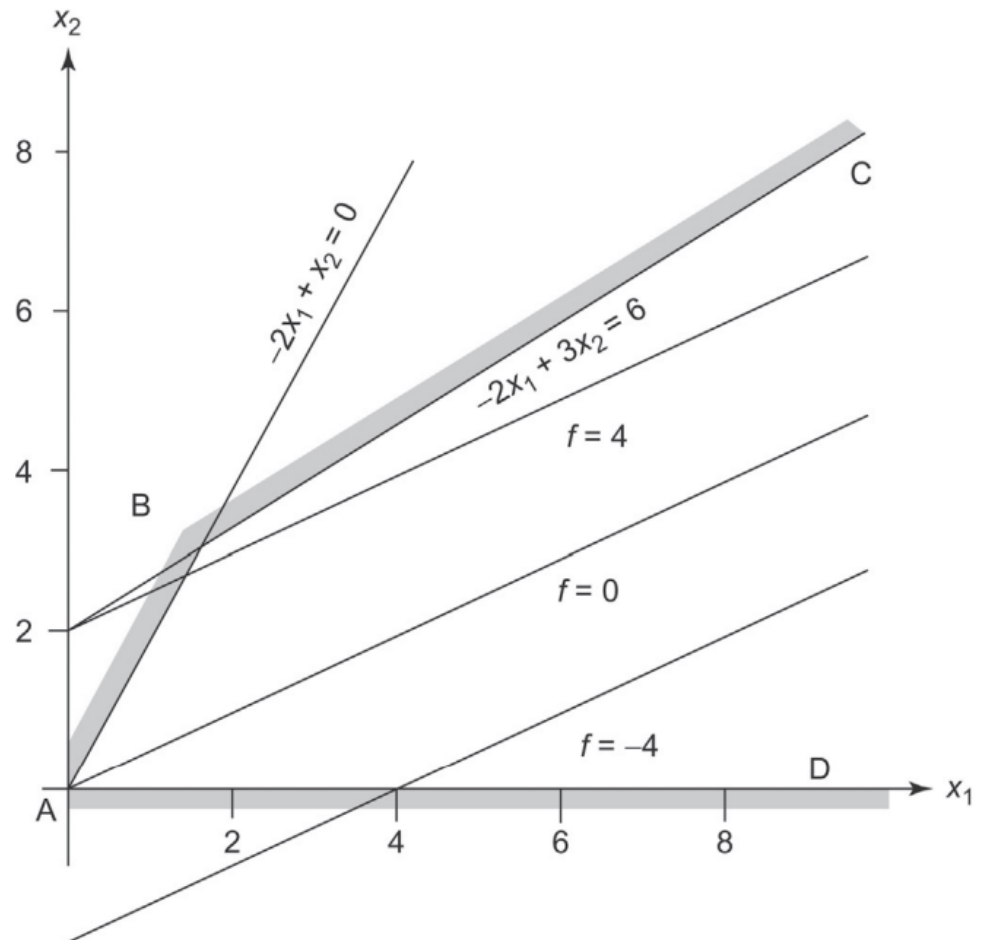
Alguns problemas podem não ter uma solução limitada. Essa situação pode surgir se esquecermos uma restrição ou formularmos incorretamente o problema.

Minimize

$$f(x) = -x_1 + 2x_2$$

subject to

$$-2x_1 + x_2 \leq 0, -2x_1 + 3x_2 \leq 6, -x_1 \leq 0, -x_2 \leq 0$$



3.1 PROCESSO DE SOLUÇÃO GRÁFICA

- PROBLEMA DE PROJETO INVIÁVEL:**

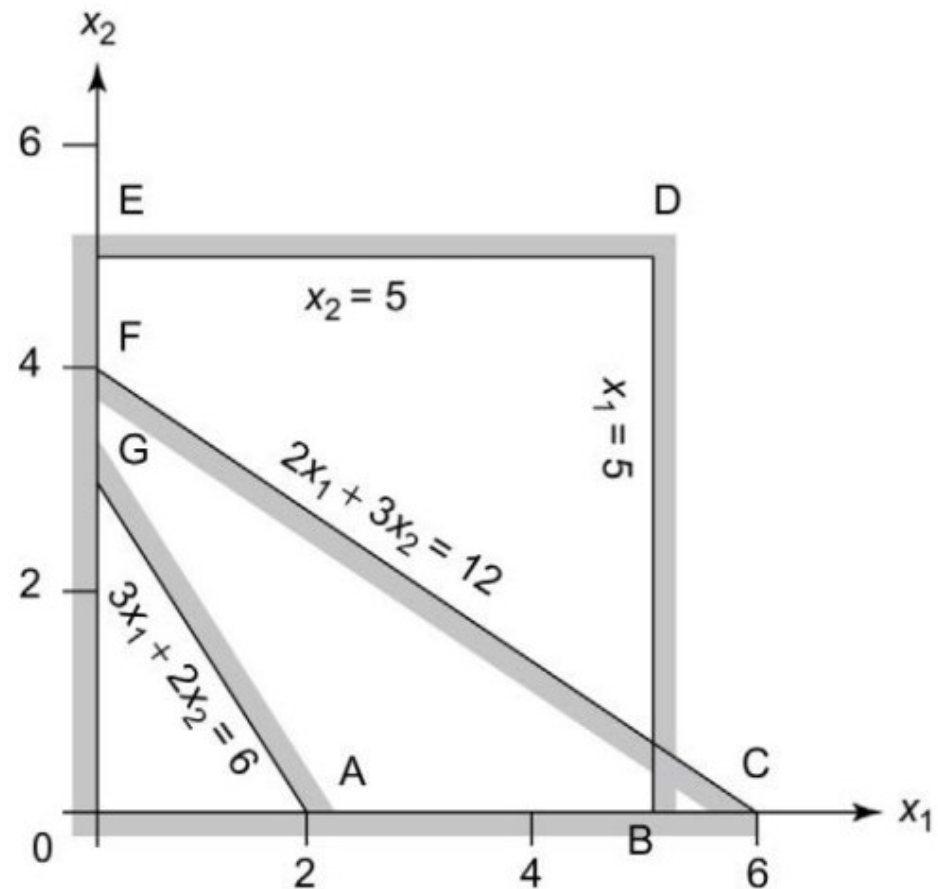
um problema de projeto pode não ter nenhuma solução viável, o que acontece quando há requisitos conflitantes ou equações de restrição inconsistentes.

Minimize

$$f(x) = x_1 + 2x_2$$

subject to

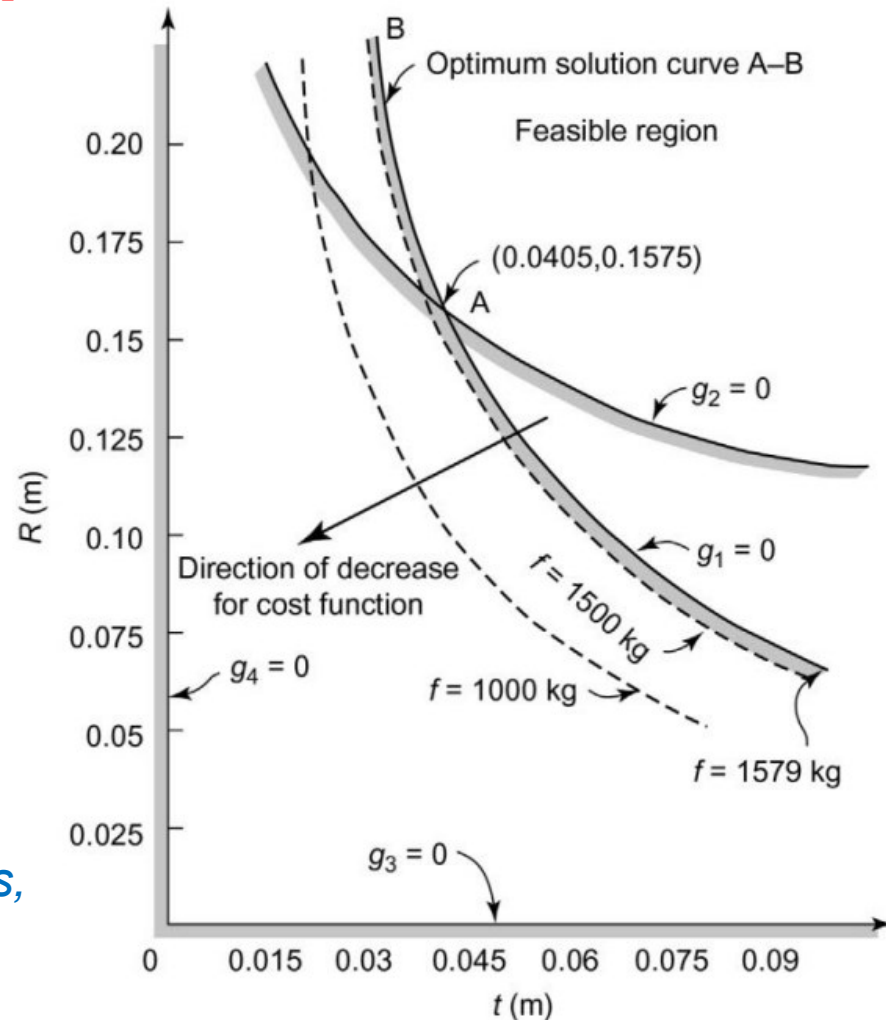
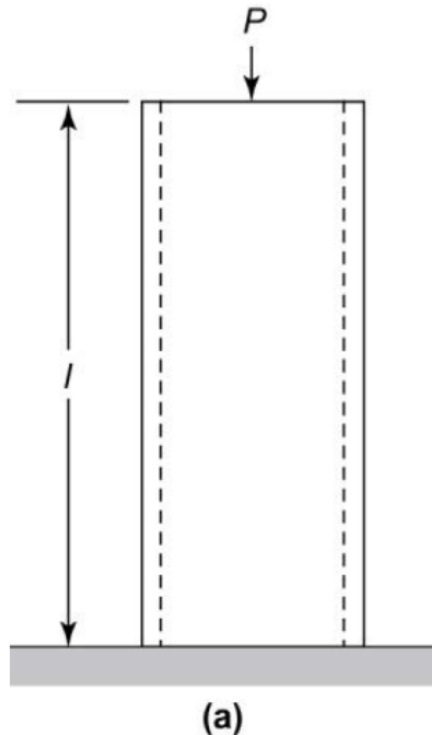
$$3x_1 + 2x_2 \leq 6, 2x_1 + 3x_2 \geq 12, x_1, x_2 \leq 5, x_1, x_2 \geq 0$$



3.1 PROCESSO DE SOLUÇÃO GRÁFICA

- SOLUÇÃO GRÁFICA PARA A COLUNA TUBULAR DE PESO MÍNIMO:**

Dados: $P = 10\text{ MN}$, $E = 207\text{ GPa}$, $\rho = 7833\text{ kg/m}^3$, $l = 5,0\text{ m}$, e $\sigma_a = 248\text{ MPa}$. “Encontre a espessura t (m) e o raio médio R (m) para minimizar a função de massa”

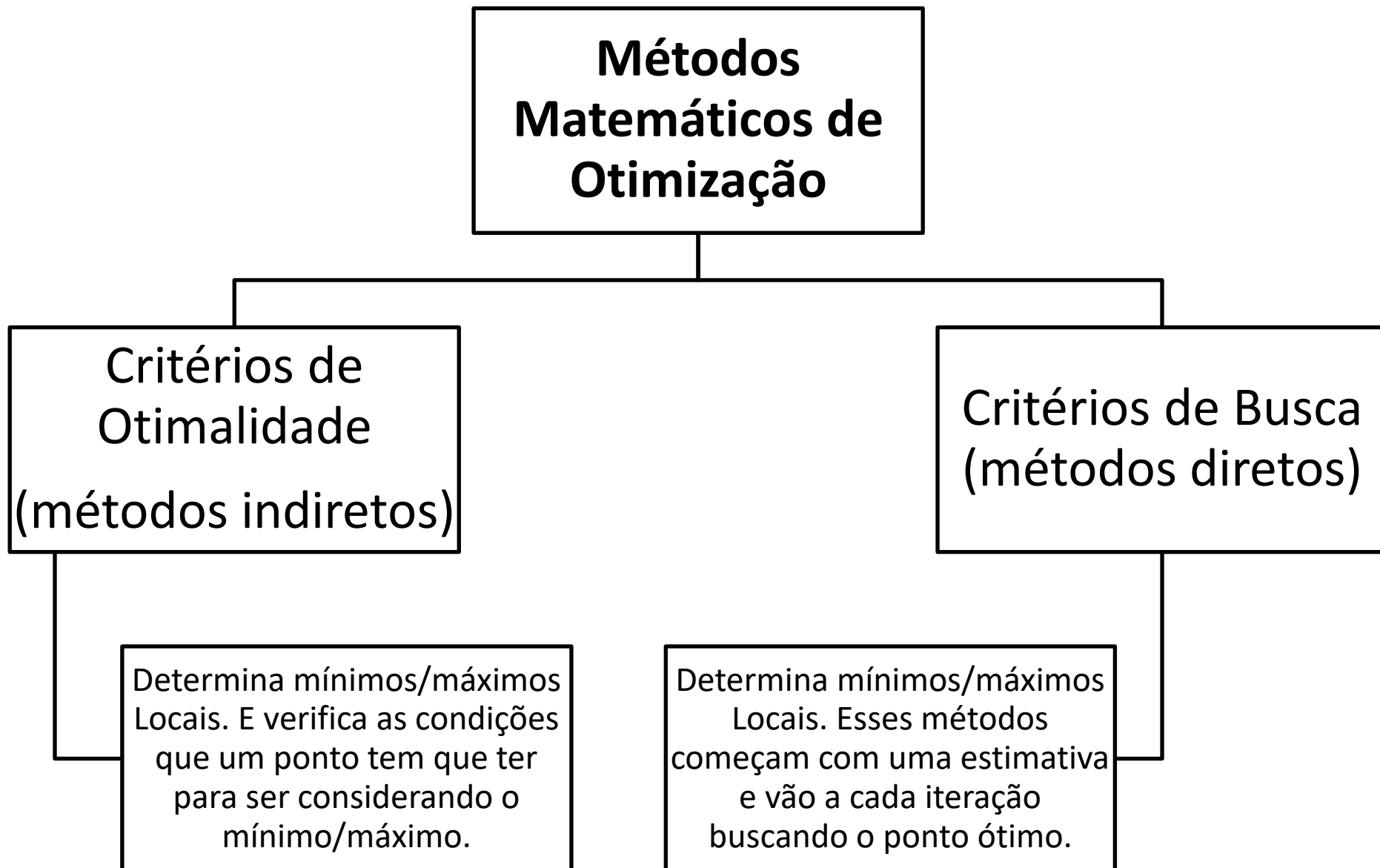


O problema tem infinitos ótimos, ou seja, toda a curva A-B!

4. Condições de Otimização

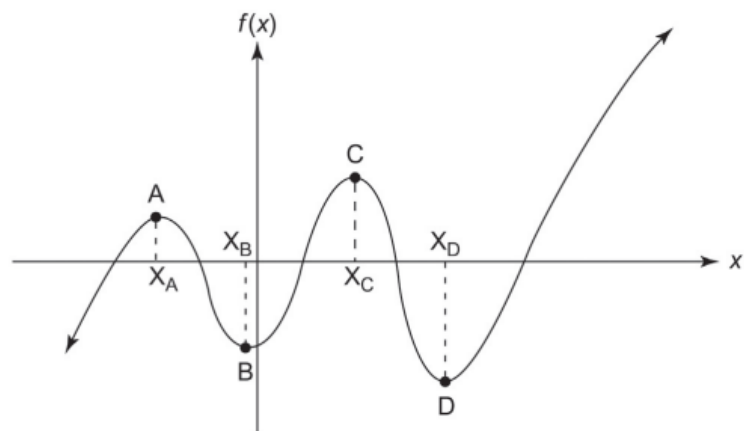
4.1 MÍNIMOS GLOBAIS E LOCAIS

4.1 MÍNIMOS GLOBAIS E LOCAIS

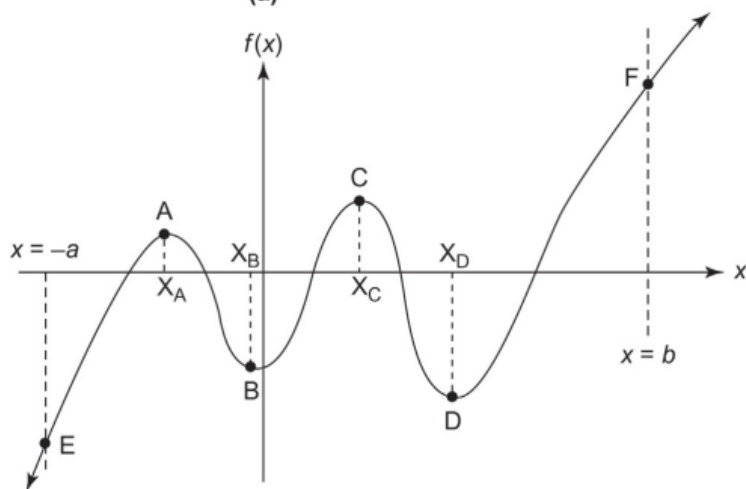


4.1 MÍNIMOS GLOBAIS E LOCAIS

Para entender a diferença entre os mínimos **GLOBAIS** e **LOCAIS** de uma função, considere os gráficos de uma função $f(x)$ de uma variável.



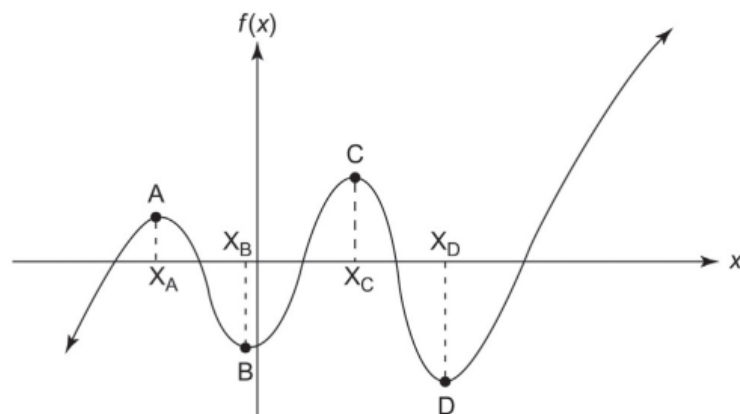
(a)



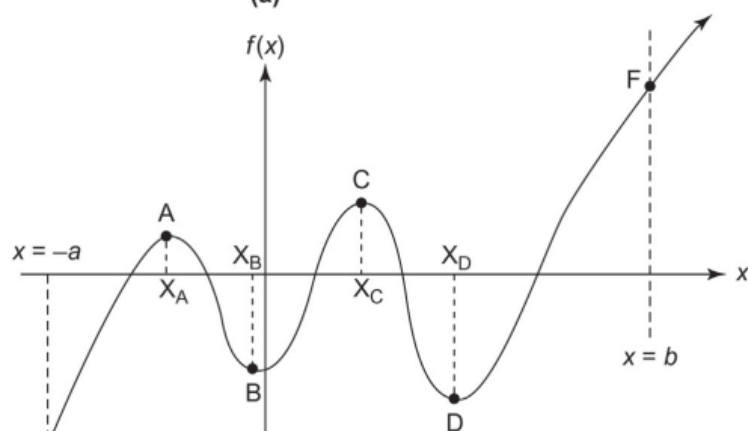
(b)

4.1 MÍNIMOS GLOBAIS E LOCAIS

Para entender a diferença entre os mínimos **GLOBAIS** e **LOCAIS** de uma função, considere os gráficos de uma função $f(x)$ de uma variável.



(a)

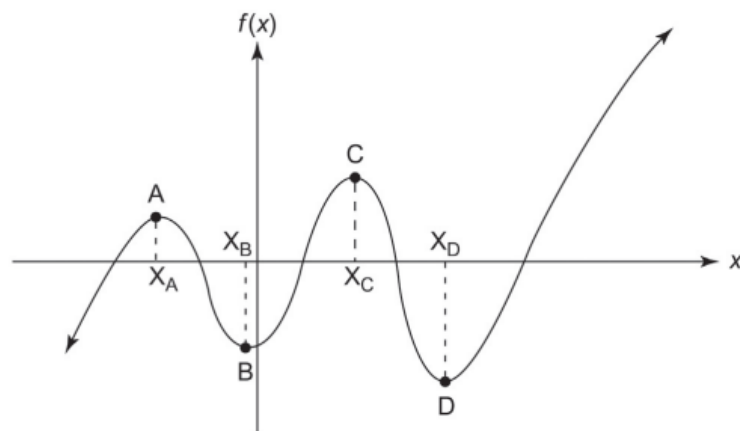


(b)

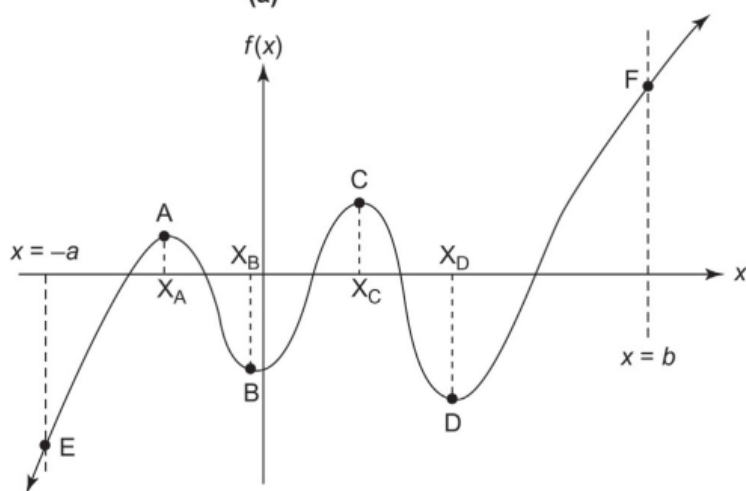
Os pontos (x_B e x_D) são mínimos locais, e (x_A e x_C) máximos locais. Estes são chamados de pontos mínimos/máximos locais porque examinamos apenas uma pequena vizinhança dos pontos. Não há mínimo/máximo global para a função da Figura (a), pois o domínio e a função $f(x)$ são ilimitados.

4.1 MÍNIMOS GLOBAIS E LOCAIS

Para entender a diferença entre os mínimos **GLOBAIS** e **LOCAIS** de uma função, considere os gráficos de uma função $f(x)$ de uma variável.



(a)



(b)

Para um ponto mínimo/máximo global, precisamos examinar todo o domínio da função e determinar se ela tem o valor mínimo/máximo em todo o seu domínio.

4.1 MÍNIMOS GLOBAIS E LOCAIS

Matematicamente:

Mínimo Global: $\mathbf{x}^*$ é um mínimo global se $f(\mathbf{x}^*) \leq f(\mathbf{x})$ para todo $\mathbf{x} \in S$.

Mínimo Local: $\mathbf{x}^*$ é um mínimo local se $f(\mathbf{x}^*) \leq f(\mathbf{x})$ para todo $\mathbf{x} \in N$. Sendo N um conjunto na vizinhança de $\mathbf{x}^*$.

$$N = \{\mathbf{x} \mid \mathbf{x} \in S \text{ with } \|\mathbf{x} - \mathbf{x}^*\| < \delta\} \quad \delta > 0$$


4.1 MÍNIMOS GLOBAIS E LOCAIS

Observe também que os máximos globais e locais são definidos de maneira semelhante, simplesmente invertendo a desigualdade:

Máximo Global: $\mathbf{x}^*$ é um máximo global se $f(\mathbf{x}^*) \geq f(\mathbf{x})$ para todo $\mathbf{x} \in S$.

Máximo Local: $\mathbf{x}^*$ é um máximo local se $f(\mathbf{x}^*) \geq f(\mathbf{x})$ para todo $\mathbf{x} \in N$. Sendo N um conjunto na vizinhança de $\mathbf{x}^*$.

$$N = \{\mathbf{x} \mid \mathbf{x} \in S \text{ with } \|\mathbf{x} - \mathbf{x}^*\| < \delta\} \quad \delta > 0$$


4.1.2 Existência de um Mínimo

Em certos casos, podemos garantir a existência de um mínimo, mesmo que não saibamos como encontrá-lo.

O teorema de **WEIERSTRASS** garante isso quando certas condições são satisfeitas:

4.1.2 Existência de um Mínimo

Teorema de **WEIERSTRASS**:

Se $f(\mathbf{x})$ é **contínua** em um conjunto admissível de S , **não vazio**, **fechado**¹ e **limitado**², então $f(\mathbf{x})$ tem um mínimo global em S .

4.1.2 Existência de um Mínimo

Teorema de **WEIERSTRASS**:

Se $f(\mathbf{x})$ é **contínua** em um conjunto admissível de $\mathbf{S}$, não vazio, fechado¹ e limitado², então $f(\mathbf{x})$ tem um mínimo global em $\mathbf{S}$.

Fechado¹: Isso implica que as restrições de desigualdade deve ser sempre: “ $\leq$ ” ou “ $\geq$ ”.

$$[3, 8] = \{x \in \mathbb{R} \mid 3 \leq x \leq 8\}$$



4.1.2 Existência de um Mínimo

Teorema de **WEIERSTRASS**:

Se $f(\mathbf{x})$ é **contínua** em um conjunto admissível de $\mathbf{S}$, não vazio, fechado¹ e limitado², então $f(\mathbf{x})$ tem um mínimo global em $\mathbf{S}$.

Limitado²: Um conjunto é limitado se, para qualquer ponto, $\mathbf{x} \in \mathbf{S}$, $(\mathbf{x}^T \cdot \mathbf{x}) < c$, onde c é um número finito.

Geralmente, é difícil verificar a condição de limitação, pois esta condição deve ser verificada para pontos infinitos em $\mathbf{S}$.

TEOREMA DE WEIERSTRASS

EXEMPLO 1: Considere uma função $f(x) = -1/x$ definida no conjunto $S = \{x \mid 0 < x \leq 1\}$. Verifique a existência de um mínimo global para a função.

TEOREMA DE WEIERSTRASS

EXEMPLO 1: Considere uma função $f(x) = -1/x$ definida no conjunto $S = \{x \mid 0 < x \leq 1\}$. Verifique a existência de um mínimo global para a função.

Solução:

1. O conjunto viável **S não é fechado**, pois não inclui o ponto de fronteira $x = 0$.
2. As condições do teorema de **Weierstrass não são satisfeitas**, embora f seja contínua em S .
3. A **existência de um mínimo global não é garantida** e não há ponto x^* satisfazendo $f(x^*) \leq f(x)$ para todo $x \in S$.

TEOREMA DE WEIERSTRASS

EXEMPLO 2: Considere agora a mesma função $f(x) = -1/x$, porém, agora, definida no conjunto $S = \{x \mid 0 \leq x \leq 1\}$. Verifique a existência de um mínimo global para a função.

TEOREMA DE WEIERSTRASS

EXEMPLO 2: Considere agora a mesma função $f(x) = -1/x$, porém, agora, definida no conjunto $S = \{x \mid 0 \leq x \leq 1\}$. Verifique a existência de um mínimo global para a função.

Solução:

1. O conjunto é fechado e limitado.
2. f não é definida em $x = 0$ (portanto, **não contínua**), portanto as **condições do teorema** ainda **não são satisfeitas** e não há garantia de um mínimo global para f no conjunto S .

TEOREMA DE WEIERSTRASS

É importante, frisar que quando as condições do teorema não são satisfeitos, ainda pode existir uma solução global! ou seja, não é um teorema “se e somente se”.

O teorema não descarta a possibilidade de um mínimo global se suas condições não forem atendidas!

TEOREMA DE WEIERSTRASS

EXEMPLO 3: Considere o problema de minimizar $f(x) = x^2$ sujeito às restrições $-1 < x < 1$.

Solução:

1. O conjunto não é fechado, as condições do teorema de Weierstrass não são atendidas. Portanto, a existência de um mínimo global não pode ser garantida.
2. Porém, a função tem um mínimo global no ponto $x = 0$.
3. O teorema não fornece um método para encontrar um ponto de mínimo global, mesmo que suas condições sejam satisfeitas; é apenas um teorema de existência.

4.2 REVISÃO DE ALGUNS CONCEITOS BÁSICOS DE CÁLCULO

4.2.1 Notações de vetor e matriz

Como a maioria dos problemas de otimização envolve funções de diversas variáveis, essas condições usam ideias de cálculo vetorial. É importante estar familiarizado com esses materiais para entender as condições de otimização.

Vetor Gradiente

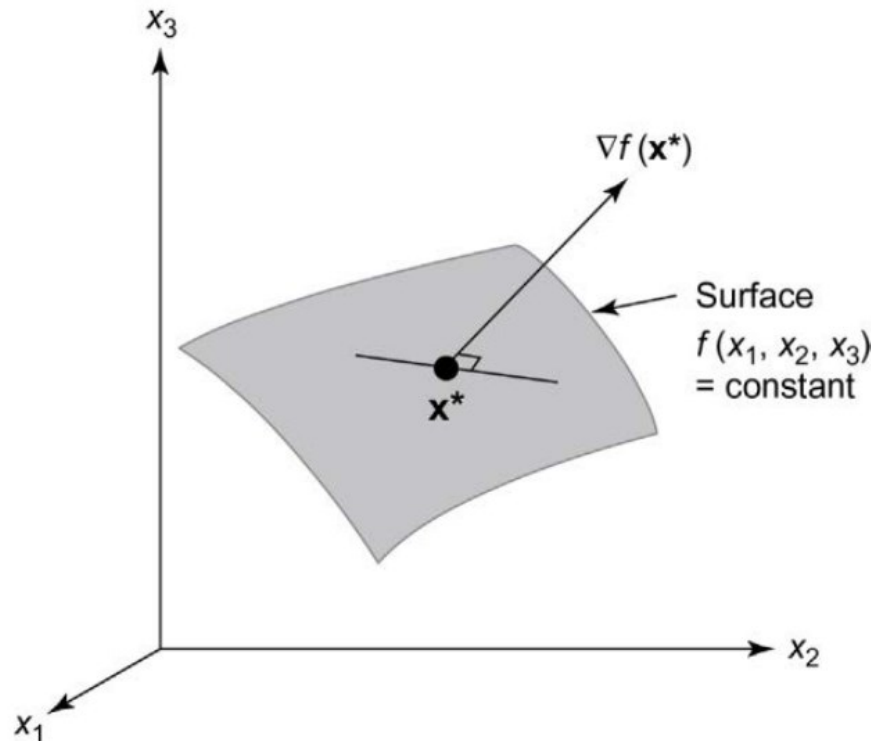
Vetor Gradiente:

Vetor contendo a derivada parcial de uma função em termos de cada uma das variáveis de projeto.

$$\mathbf{c} = \nabla f(\mathbf{x}^*) = \begin{bmatrix} \frac{\partial f(\mathbf{x}^*)}{\partial x_1} \\ \frac{\partial f(\mathbf{x}^*)}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x}^*)}{\partial x_n} \end{bmatrix} = \left[\frac{\partial f(\mathbf{x}^*)}{\partial x_1} \frac{\partial f(\mathbf{x}^*)}{\partial x_2} \dots \frac{\partial f(\mathbf{x}^*)}{\partial x_n} \right]^T$$

Vetor Gradiente

1. Geometricamente, o gradiente é normal ao plano tangente de $f(\mathbf{x}^*)$, para uma função de três variáveis;
2. O gradiente aponta na direção de maior crescimento de f .



Vetor Gradiente

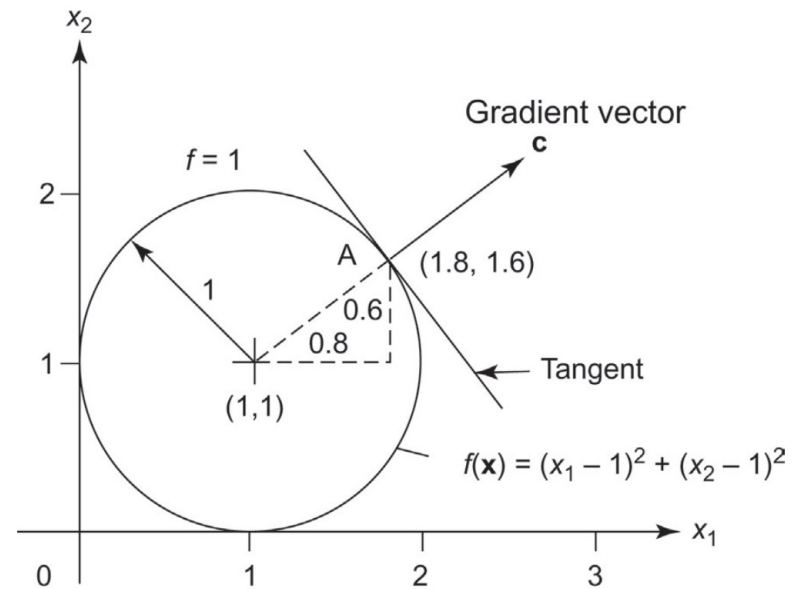
EXEMPLO 1: Calcule o vetor gradiente para a função $f(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 1)^2$ no ponto $\mathbf{x}^* = (1,8 ; 1,6)$.

Vetor Gradiente

EXEMPLO 1: Calcule o vetor gradiente para a função $f(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 1)^2$ no ponto $\mathbf{x}^* = (1,8 ; 1,6)$.

$$\frac{\partial f}{\partial x_1}(1,8,1,6) = 2(x_1 - 1) = 2(1,8 - 1) = 1,6$$

$$\frac{\partial f}{\partial x_2}(1,8,1,6) = 2(x_2 - 1) = 2(1,6 - 1) = 1,2$$



Matriz Hessiana

Matriz Hessiana:

É a matriz (simétrica) formada pelas derivadas de segunda ordem de f em termos das variáveis de projeto.

$$\mathbf{H} = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right] = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Matriz Hessiana

EXEMPLO 2: Para a seguinte função, calcule o vetor gradiente e a matriz Hessiana no ponto (1; 2).

$$f(x) = x_1^3 + x_2^3 + 2x_1^2 + 3x_2^2 - x_1x_2 + 2x_1 + 4x_2$$

Matriz Hessiana

EXEMPLO 2: Para a seguinte função, calcule o vetor gradiente e a matriz Hessiana no ponto (1; 2).

$$f(x) = x_1^3 + x_2^3 + 2x_1^2 + 3x_2^2 - x_1x_2 + 2x_1 + 4x_2$$

$$\frac{\partial f}{\partial x_1} = 3x_1^2 + 4x_1 - x_2 + 2; \quad \frac{\partial f}{\partial x_2} = 3x_2^2 + 6x_2 - x_1 + 4$$

Matriz Hessiana

EXEMPLO 2: Para a seguinte função, calcule o vetor gradiente e a matriz Hessiana no ponto (1; 2).

$$f(x) = x_1^3 + x_2^3 + 2x_1^2 + 3x_2^2 - x_1x_2 + 2x_1 + 4x_2$$

$$\frac{\partial f}{\partial x_1} = 3x_1^2 + 4x_1 - x_2 + 2; \quad \frac{\partial f}{\partial x_2} = 3x_2^2 + 6x_2 - x_1 + 4$$

$$\frac{\partial^2 f}{\partial x_1^2} = 6x_1 + 4; \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = -1; \quad \frac{\partial^2 f}{\partial x_2 \partial x_1} = -1; \quad \frac{\partial^2 f}{\partial x_2^2} = 6x_2 + 6.$$

$$\mathbf{H}(x) = \begin{bmatrix} 6x_1 + 4 & -1 \\ -1 & 6x_2 + 6 \end{bmatrix}$$

$$\mathbf{H}(1, 2) = \begin{bmatrix} 10 & -1 \\ -1 & 18 \end{bmatrix}$$

Expansão de Taylor

- A ideia da expansão de Taylor é fundamental para o desenvolvimento de conceitos de projeto otimizados e métodos numéricos, por isso é explicada aqui.
- Usando a expansão de Taylor, uma função pode ser aproximada por polinômios em uma vizinhança de qualquer ponto em função de f e suas derivadas.

Expansão de Taylor

Considere primeiro uma função $f(x)$ de uma variável. Assim, a Expansão de Taylor para $f(x)$ em torno do ponto x^* :

$$f(x) = f(x^*) + \frac{df(x^*)}{dx}(x - x^*) + \frac{1}{2} \frac{d^2f(x^*)}{dx^2}(x - x^*)^2 + R$$

Onde **R** é o termo restante que é menor em magnitude do que os termos anteriores se x estiver suficientemente próximo de x^* .

Expansão de Taylor

$$f(x) = f(x^*) + \frac{df(x^*)}{dx}(x - x^*) + \frac{1}{2} \frac{d^2f(x^*)}{dx^2}(x - x^*)^2 + R$$

Se fizermos $x - x^* = d$ (uma pequena mudança no ponto x^*), a expansão de Taylor torna-se um polinômio quadrático em d :

$$f(x^* + d) = f(x^*) + \frac{df(x^*)}{dx}d + \frac{1}{2} \frac{d^2f(x^*)}{dx^2}d^2 + R$$

Expansão de Taylor

A expansão de Taylor pode ser generalizado para funções de n variáveis. Nesse caso, $\mathbf{x}$, $\mathbf{x}^*$ e ∇f são vetores n -dimensionais e $\mathbf{H}$ é a matriz Hessiana $n \times n$.

$$f(\mathbf{x}^* + \mathbf{d}) = f(\mathbf{x}^*) + \nabla f^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \mathbf{H} \mathbf{d} + R$$

Frequentemente, uma mudança na função é desejada quando $\mathbf{x}$ está em alguma vizinhança de $\mathbf{x}^*$. Definindo a mudança como $\Delta f = f(\mathbf{x}) - f(\mathbf{x}^*)$:

$$\Delta f = \nabla f^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \mathbf{H} \mathbf{d} + R$$

Expansão de Taylor

EXEMPLO 3: Obtenha a expansão de Taylor de segunda ordem para $f(\mathbf{x}) = 3x_1^3x_2$ no ponto $\mathbf{x}^* = (1; 1)$.

Expansão de Taylor

O gradiente da função $f(\mathbf{x})$ no ponto $\mathbf{x}^* = (1; 1)$:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 9x_1^2 x_2 \\ 3x_1^3 \end{bmatrix} = \begin{bmatrix} 9 \\ 3 \end{bmatrix}$$

A matriz Hessiana:

$$\frac{\partial^2 f}{\partial x_1^2} = 18x_1 x_2; \quad \frac{\partial^2 f}{\partial x_2 x_1} = 9x_1^2; \quad \frac{\partial^2 f}{\partial x_1 x_2} = 9x_1^2; \quad \frac{\partial^2 f}{\partial x_2^2} = 0$$

$$\mathbf{H} = \begin{bmatrix} 18x_1 x_2 & 9x_1^2 \\ 9x_1^2 & 0 \end{bmatrix} = \begin{bmatrix} 18 & 9 \\ 9 & 0 \end{bmatrix}$$

Expansão de Taylor

Resolvendo:

$$\bar{f} = 3 + \begin{bmatrix} 9 \\ 3 \end{bmatrix}^T \begin{bmatrix} (x_1 - 1) \\ (x_2 - 1) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} (x_1 - 1) \\ (x_2 - 1) \end{bmatrix}^T \begin{bmatrix} 18 & 9 \\ 9 & 0 \end{bmatrix} \begin{bmatrix} (x_1 - 1) \\ (x_2 - 1) \end{bmatrix}$$

Simplificando a expressão, obtemos a expansão de Taylor para $f(\mathbf{x})$ sobre o ponto $(1; 1)$:

$$\bar{f}(\mathbf{x}) = 9x_1^2 + 9x_1x_2 - 18x_1 - 6x_2 + 9$$

Expansão de Taylor

EXEMPLO 4: Obtenha a expansão de Taylor de primeira ordem para: $f(\mathbf{x}) = x_1^2 + x_2^2 - 4x_1 - 2x_2 + 4$ no ponto $\mathbf{x}^* = (1; 2)$.

Expansão de Taylor

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} (2x_1 - 4) \\ (2x_2 - 2) \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

$$\bar{f}(\mathbf{x}) = 1 + [-2 \ 2] \begin{bmatrix} (x_1 - 1) \\ (x_2 - 2) \end{bmatrix} = -2x_1 + 2x_2 - 1$$

Para ver com que precisão a expansão se aproxima de $f(\mathbf{x})$ na vizinhança de $(1; 2)$, calculamos as funções no ponto $(1.1; 2.2)$, uma variação de 10% no ponto. Vemos que a função aproximada subestima a função real em 4%. Um erro desta magnitude é bastante aceitável em muitas aplicações. No entanto, observe que os erros serão diferentes para diferentes funções e podem ser maiores para funções altamente não lineares.

$$\bar{f}(\mathbf{x}) = 1.20 \text{ and } f(\mathbf{x}) = 1.25.$$

Formas Quadráticas e Matrizes Definidas

A forma quadrática é uma função não linear especial que possui apenas termos de 2ª ordem (seja o quadrado de uma variável ou o produto de duas variáveis); por exemplo, a seguinte função de 3 variáveis.

$$F(\mathbf{x}) = x_1^2 + 2x_2^2 + 3x_3^2 + 2x_1x_2 - 2x_2x_3 + 4x_3x_1$$

$$F(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n p_{ij} x_i x_j$$

onde p_{ij} são constantes relacionadas aos coeficientes de vários termos

Formas Quadráticas e Matrizes Definidas

A forma quadrática pode ser escrita na notação matricial. Seja $\mathbf{P} = [p_{ij}]$ uma matriz $n \times n$ e $\mathbf{x} = (x_1, x_2, \dots, x_n)$ um vetor n -dimensional. Então a forma quadrática pode ser:

$$F(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x}$$

$\mathbf{P}$ é chamada de matriz da forma quadrática $F(\mathbf{x})$. Os elementos de $\mathbf{P}$ são obtidos a partir dos coeficientes dos termos da função $F(\mathbf{x})$;

$\mathbf{P}$ pode assumir várias formas, Porém uma delas é simétrica ($\mathbf{A}$);

$$\mathbf{A} = \frac{1}{2}(\mathbf{P} + \mathbf{P}^T) \quad \text{or} \quad a_{ij} = \frac{1}{2}(p_{ij} + p_{ji}), \quad i, j = 1 \text{ to } n$$

Formas Quadráticas e Matrizes Definidas

EXEMPLO 5: Identifique a matriz A associada à forma quadrática:

$$F(x_1, x_2, x_3) = 2x_1^2 + 2x_1x_2 + 4x_1x_3 - 6x_2^2 - 4x_2x_3 + 5x_3^2$$

Formas Quadráticas e Matrizes Definidas

EXEMPLO 5: Identifique a matriz A associada à forma quadrática:

$$F(x_1, x_2, x_3) = 2x_1^2 + 2x_1x_2 + 4x_1x_3 - 6x_2^2 - 4x_2x_3 + 5x_3^2$$

$$F(\mathbf{x}) = [x_1 \ x_2 \ x_3] \begin{bmatrix} 2 & 2 & 4 \\ 0 & -6 & -4 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 2 \\ 1 & -6 & -2 \\ 2 & -2 & 5 \end{bmatrix}$$

A forma quadrática $F(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ pode ser positiva, negativa ou zero para qualquer $\mathbf{x} \neq 0$. A seguir estão as formas possíveis para a função $F(\mathbf{x})$ e a matriz simétrica associada $\mathbf{A}$:

1. Definida positiva. $F(\mathbf{x}) > 0$ para todo $\mathbf{x} \neq 0$. Autovalores > 0 .
2. Semidefinida positiva. $F(\mathbf{x}) \geq 0$ para todo $\mathbf{x} \neq 0$. Autovalores ≥ 0 .
3. Definida negativa. $F(\mathbf{x}) < 0$ para todo $\mathbf{x} \neq 0$. Autovalores < 0 .
4. Semidefinida negativa. $F(\mathbf{x}) \leq 0$ para todo $\mathbf{x} \neq 0$. Autovalores ≤ 0 .
5. Indefinido. A forma quadrática é chamada indefinida se for positiva para alguns valores de $\mathbf{x}$ e negativa para alguns outros.

Diferenciação de uma Forma Quadrática

Frequentemente, queremos encontrar o gradiente e a matriz Hessiana para a forma quadrática:

$$\frac{\partial F(\mathbf{x})}{\partial x_i} = 2 \sum_{j=1}^n a_{ij} x_j; \quad \text{or} \quad \nabla F(\mathbf{x}) = 2\mathbf{A}\mathbf{x}$$

$$\frac{\partial^2 F(\mathbf{x})}{\partial x_j \partial x_i} = 2a_{ij}; \quad \text{or} \quad \mathbf{H} = 2\mathbf{A}$$

Formas Quadráticas e Matrizes Definidas

EXEMPLO 6: Calcule a Hessiana e a Matriz A da seguinte forma quadrática:

$$F(\mathbf{x}) = 2x_1^2 + 2x_1x_2 + 4x_1x_3 - 6x_2^2 - 4x_2x_3 + 5x_3^2$$

Formas Quadráticas e Matrizes Definidas

$$F(\mathbf{x}) = 2x_1^2 + 2x_1x_2 + 4x_1x_3 - 6x_2^2 - 4x_2x_3 + 5x_3^2$$

$$\nabla F(\mathbf{x}) = \begin{bmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \\ \frac{\partial F}{\partial x_3} \end{bmatrix} = \begin{bmatrix} (4x_1 + 2x_2 + 4x_3) \\ (2x_1 - 12x_2 - 4x_3) \\ (4x_1 - 4x_2 + 10x_3) \end{bmatrix}$$

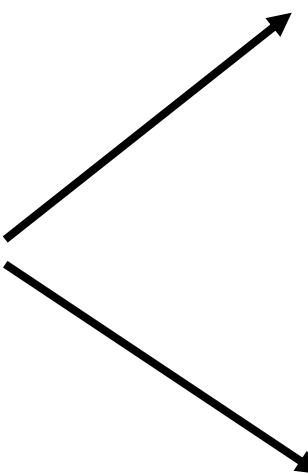
$$\begin{aligned} \frac{\partial^2 F}{\partial x_1^2} &= 4, \quad \frac{\partial^2 F}{\partial x_1 \partial x_2} = 2, \quad \frac{\partial^2 F}{\partial x_1 \partial x_3} = 4 \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} &= 2, \quad \frac{\partial^2 F}{\partial x_2^2} = -12, \quad \frac{\partial^2 F}{\partial x_2 \partial x_3} = -4 \\ \frac{\partial^2 F}{\partial x_3 \partial x_1} &= 4, \quad \frac{\partial^2 F}{\partial x_3 \partial x_2} = -4, \quad \frac{\partial^2 F}{\partial x_3^2} = 10 \end{aligned}$$

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 2 \\ 1 & -6 & -2 \\ 2 & -2 & 5 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 4 & 2 & 4 \\ 2 & -12 & -4 \\ 4 & -4 & 10 \end{bmatrix}$$

Formas Quadráticas e Matrizes Definidas

Atenção: Note com é mais simples calcular a matriz $\mathbf{A}$ (para formas quadráticas) e em seguida calcular o gradiente e a Hessiana.

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 2 \\ 1 & -6 & -2 \\ 2 & -2 & 5 \end{bmatrix}$$


The diagram shows two arrows originating from the right side of the matrix $\mathbf{A}$. The upper arrow points to the equation $\nabla F(\mathbf{x}) = 2\mathbf{A}\mathbf{x}$, and the lower arrow points to the equation $\mathbf{H} = 2\mathbf{A}$.

$$\nabla F(\mathbf{x}) = 2\mathbf{A}\mathbf{x}$$
$$\mathbf{H} = 2\mathbf{A}$$

4.4 CONCEITOS DE CONDIÇÕES NECESSÁRIAS E SUFICIENTES

4.4 NECESSÁRIAS E SUFICIENTES

1. Os pontos ótimos devem satisfazer as condições necessárias. Os pontos que não os satisfazem não podem ser ótimos. As condições necessárias de otimalidade depende de derivadas de primeira e segunda ordem.
2. Um ponto que satisfaça as condições necessárias não precisa ser ótimo; ou seja, pontos não ótimos também podem satisfazer as condições necessárias.
3. Um ponto candidato que satisfaça uma condição suficiente é de fato um ponto ótimo. Essas condições envolvem derivadas parciais de ordem superior ou de segunda ordem da função.
4. Se a condição de suficiência não puder ser usada ou não for satisfeita, podemos não ser capazes de tirar nenhuma conclusão sobre a otimalidade do ponto candidato.

4.5 CONDIÇÕES DE OTIMIDADE: PROBLEMA SEM RESTRIÇÕES

4.5 PROBLEMA SEM RESTRIÇÕES

Nesta seção, discutiremos as condições necessárias e suficientes para problemas de otimização **SEM RESTRIÇÕES** definidos como “Minimizar $f(\mathbf{x})$ sem nenhuma restrição em $\mathbf{x}$ ”;

Tais problemas surgem com pouca frequência em aplicações práticas de engenharia. Porém, uma estratégia numérica para resolver um problema com restrição é convertê-lo em uma sequência de problemas sem restrições;

4.5 PROBLEMA SEM RESTRIÇÕES

O conceito básico para derivar condições de **ótimo local** é assumir que estamos em um ponto mínimo $\mathbf{x}^*$ e então examinar sua vizinhança para estudar as propriedades da função e suas derivadas.

4.5 PROBLEMA SEM RESTRIÇÕES

Seja $\mathbf{x}^*$ um ponto de mínimo local para $f(\mathbf{x})$. Para investigar sua vizinhança, seja $\mathbf{x}$ qualquer ponto próximo a $\mathbf{x}^*$. Seja $\mathbf{d} = \mathbf{x} - \mathbf{x}^*$ e $\Delta f = f(\mathbf{x}) - f(\mathbf{x}^*)$.

Se $f(\mathbf{x})$ tem um mínimo local em $\mathbf{x}^*$, qualquer perturbação levará a um acréscimo no valor da função, ou seja, $\Delta f = f(\mathbf{x}) - f(\mathbf{x}^*) \geq 0$.

Como estamos na vizinhança de f podemos aproximar a função por uma expansão de Taylor de 1ª ordem:

~~$$\Delta f(x) = f'(x^*)d + \frac{1}{2} f''(x^*)d^2 + R$$~~

~~$$\Delta f = \nabla f(\mathbf{x}^*)^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \mathbf{H}(\mathbf{x}^*) \mathbf{d} + R$$~~

4.5 PROBLEMA SEM RESTRIÇÕES

Considerando somente o termo de 1ª ordem:

$$\Delta f(x) = f'(x^*)d + \cancel{\frac{1}{2} f''(x^*)d^2} + R$$

$$\Delta f = f'(x^*)d_1 \geq 0 \quad \text{Ocorrerá sempre que:} \quad \begin{aligned} \nabla f(x^*) &= 0 \\ f'(x^*) &= 0 \end{aligned}$$

- **Condição Necessária de Primeira Ordem.**
- **Válido também para pontos de máximos locais.**

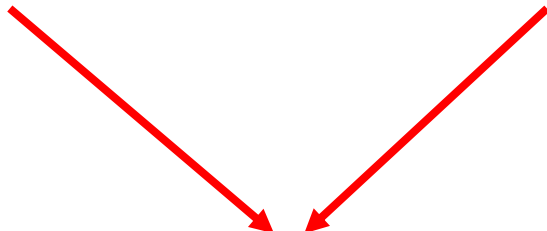
4.5 PROBLEMA SEM RESTRIÇÕES

Pontos Estacionários:

Como os pontos que satisfazem a Equação abaixo podem ser mínimos ou máximos locais, eles são chamados de pontos estacionários:

$$f'(x^*) = 0$$

$$\nabla f(x^*) = 0$$



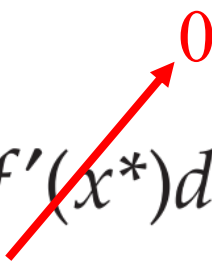
Condição Necessária de Primeira Ordem.

4.5 PROBLEMA SEM RESTRIÇÕES

Condição suficiente:

Quais dos pontos estacionários são realmente mínimos para a função?

Como os pontos estacionários satisfazem a condição necessária:

$$\Delta f(x) = \cancel{f'(x^*)d} + \frac{1}{2} f''(x^*)d^2 + R$$


$$\Delta f = \frac{1}{2} \mathbf{d}^T \mathbf{H}(\mathbf{x}^*) \mathbf{d} + \cancel{R}$$

$$\Delta f(x) = \frac{1}{2} f''(x^*)d^2 + \cancel{R}$$

Desconsiderando os termos de ordem superior R.

4.5 PROBLEMA SEM RESTRIÇÕES

Condição suficiente:

$$\Delta f(x) = \frac{1}{2} f''(x^*) d^2 > 0$$

Ocorrerá sempre que: $H(x^*)$ positiva definida

$$f''(x^*) > 0$$

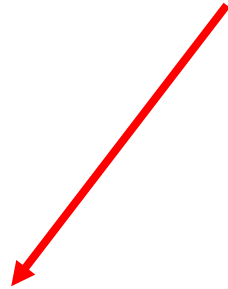
- **Condição SUFICIENTE.**

4.5 PROBLEMA SEM RESTRIÇÕES

Condição Necessária de Segunda Ordem:

Se $f''(x^*) = 0$, não podemos concluir que x^* não é um ponto mínimo.

$$f''(x^*) \geq 0$$



- **Condição Necessária de 2ª Ordem.**

4.5 PROBLEMA SEM RESTRIÇÕES

As condições de otimização para uma função de várias variáveis estão resumidas:

Problema: Encontre $\mathbf{x}$ para minimizar $f(\mathbf{x})$

- **Condição necessária de 1ª ordem:** $\nabla f = 0$ ou $f' = 0$. Qualquer ponto que satisfaça esta condição é chamado de ponto estacionário; pode ser um máximo local, mínimo local ou nenhum dos dois (ponto de inflexão);
- **Condição necessária de 2ª ordem** para um mínimo local: $f'' \geq 0$ ou $\mathbf{H}$ positivo semi definido (autovalores ≥ 0);
- **Condição necessária de 2ª ordem** para um máximo local: $f'' \leq 0$ ou $\mathbf{H}$ semi definido negativo (autovalores ≤ 0);
- **Condição suficiente de 2ª ordem** para um mínimo local: $f'' > 0$ ou $\mathbf{H}$ positivo definido (autovalores > 0);
- **Condição suficiente de 2ª ordem** para um máximo local: $f'' < 0$ $\mathbf{H}$ deve ser negativo definido (autovalores < 0);

4.5 PROBLEMA SEM RESTRIÇÕES

Note que as condições anteriores envolvem derivadas de $f(\mathbf{x})$ e não o valor da função. Portanto, se adicionarmos uma constante a $f(\mathbf{x})$, a solução $\mathbf{x}^*$ do problema de minimização permanece inalterado. Da mesma forma, se multiplicarmos $f(\mathbf{x})$ por qualquer constante positiva, o ponto mínimo $\mathbf{x}^*$ permanece inalterado, mas o valor $f(\mathbf{x}^*)$ é alterado.

Multiplicar $f(\mathbf{x})$ por uma constante negativa altera o mínimo em $\mathbf{x}^*$ para um máximo. Podemos usar essa propriedade para converter problemas de maximização em problemas de minimização multiplicando $f(\mathbf{x})$ por -1 ;

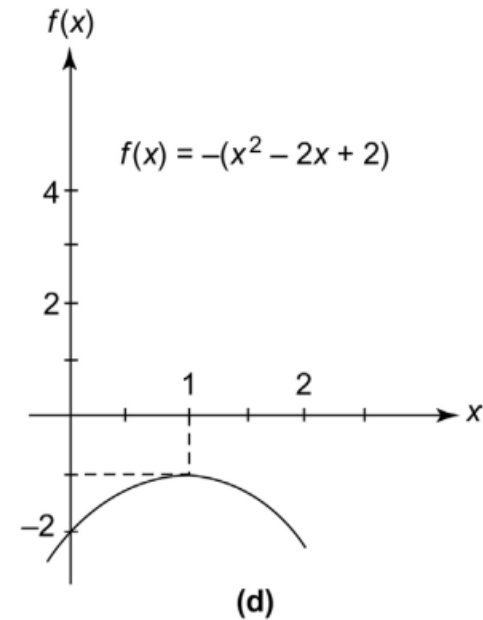
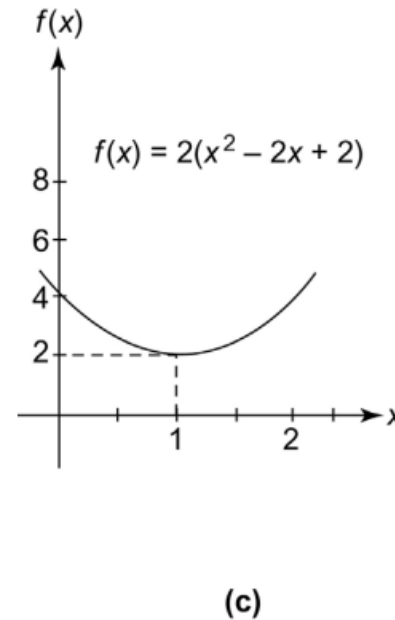
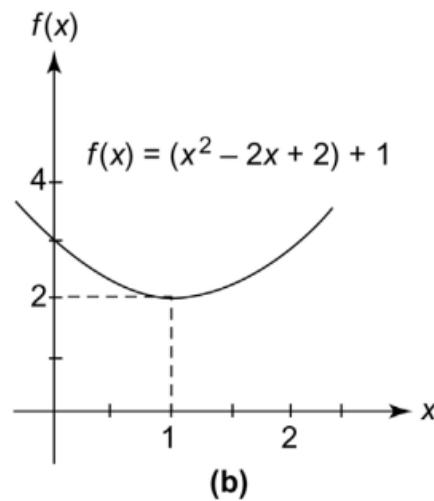
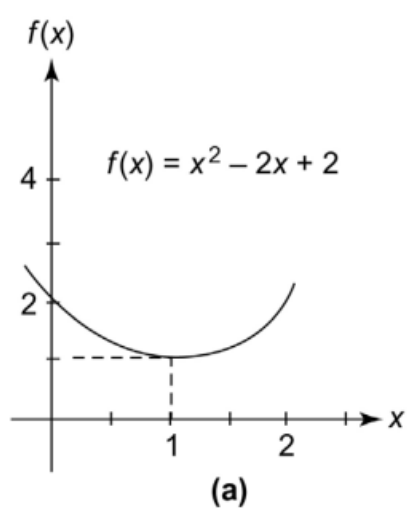
4.5 PROBLEMA SEM RESTRIÇÕES

EXEMPLO 1: Discuta o efeito das variações anteriores para a função:

$$f(x) = x^2 - 2x + 2.$$

4.5 PROBLEMA SEM RESTRIÇÕES

A Fig. (a) representa a função, que tem um mínimo em $\mathbf{x}^* = 1$. A Fig. b, c, d mostra o efeito de adicionar uma constante à função $[f(\mathbf{x}) + 1]$, multiplicando $f(\mathbf{x})$ por um número negativo $[-2f(\mathbf{x})]$. Em todos casos, o ponto estacionário permanece inalterado.



4.5 PROBLEMA SEM RESTRIÇÕES

EXEMPLO 2: Encontre pontos mínimos locais para a função:

$$f(\mathbf{x}) = x_1^2 + 2x_1x_2 + 2x_2^2 - 2x_1 + x_2 + 8$$

4.5 PROBLEMA SEM RESTRIÇÕES

As condições necessárias ($\nabla f = 0$) para o problema são:

$$\frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} (2x_1 + 2x_2 - 2) \\ (2x_1 + 4x_2 + 1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Resolvendo as equações simultaneamente, obtemos o ponto estacionário como $\mathbf{x}^* = (2,5; -1,5)$.

4.5 PROBLEMA SEM RESTRIÇÕES

Para verificar se o ponto estacionário é um mínimo local, avaliamos $\mathbf{H}$ em $\mathbf{x}^*$ diferenciando novamente os componentes do gradiente:

$$\frac{\partial^2 f}{\partial x_1^2} = 2; \quad \frac{\partial^2 f}{\partial x_2 \partial x_1} = 2; \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = 2; \quad \frac{\partial^2 f}{\partial x_2^2} = 4$$

$$\mathbf{H}(2.5, -1.5) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix}$$

4.5 PROBLEMA SEM RESTRIÇÕES

Para verificar a forma da Hessiana, calculamos autovalores de **H**:

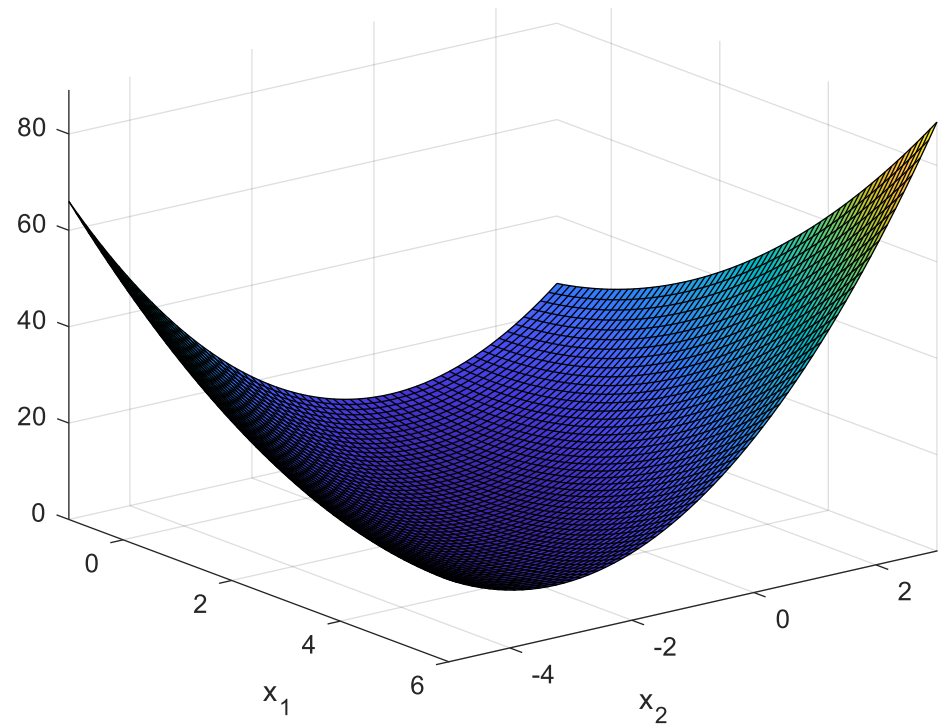
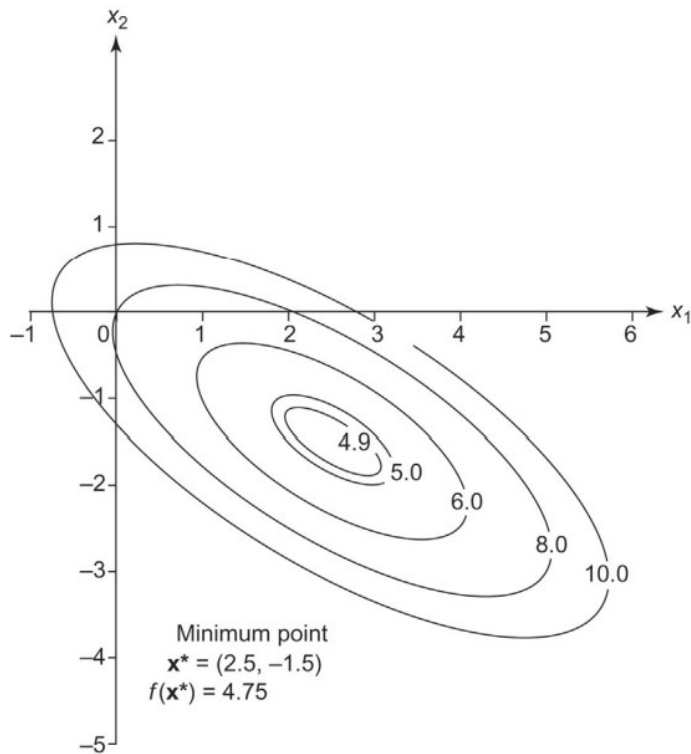
$$\begin{vmatrix} 2-\lambda & 2 \\ 2 & 4-\lambda \end{vmatrix} = 0, \quad \text{or} \quad (2-\lambda)(4-\lambda) - 2 \times 2 = 0$$

$\lambda^2 - 6\lambda + 4 = 0$ which gives eigenvalues $\lambda_1 = 5.236$, $\lambda_2 = 0.764$

Uma vez que ambos os autovalores são positivos, a Hessiana **H** é positivo definido. **Condição suficiente de 2ª** ordem para um mínimo local.

4.5 PROBLEMA SEM RESTRIÇÕES

A Figura mostra algumas curvas de isocusto para a função do problema. Vê-se que o ponto $(2.5; -1.5)$ é o mínimo para a função.



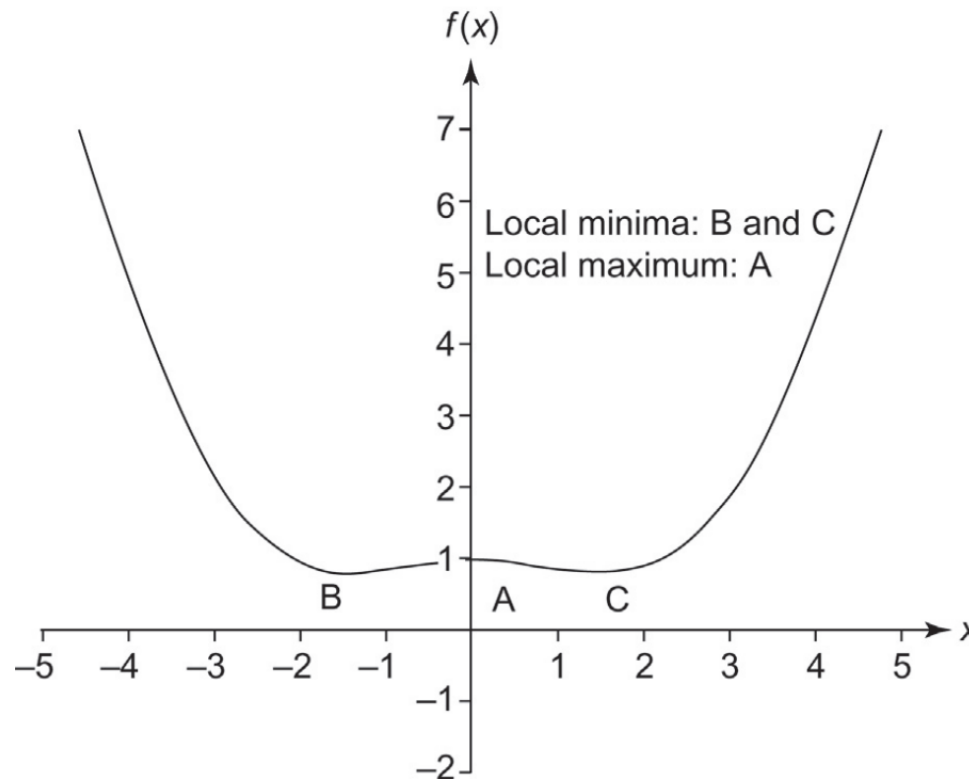
4.5 PROBLEMA SEM RESTRIÇÕES

EXEMPLO 3: Encontre pontos estacionários para a seguinte função e verifique as condições de suficiência para eles:

$$f(x) = \frac{1}{3}x^2 + \cos x$$

4.5 PROBLEMA SEM RESTRIÇÕES

Vê-se que existem três pontos estacionários: $x = 0$ (**ponto A**), x entre 1 e 2 (**ponto C**), e x entre -1 e -2 (**ponto B**). O ponto $x = 0$ é um máximo local para a função e os outros dois são mínimos locais.



4.5 PROBLEMA SEM RESTRIÇÕES

A condição necessária de 1ª ordem ($\nabla f = 0$) é:

$$f'(x) = \frac{2}{3}x - \sin x = 0$$

Vê-se que $x = 0$ satisfaz f' , então é um ponto estacionário. Devemos encontrar outras raízes. Usando métodos numéricos descobrimos que $x^* = 1,496$ e $x^* = -1,496$ satisfazem $f'(x) = 0$. Portanto, esses são pontos estacionários adicionais.

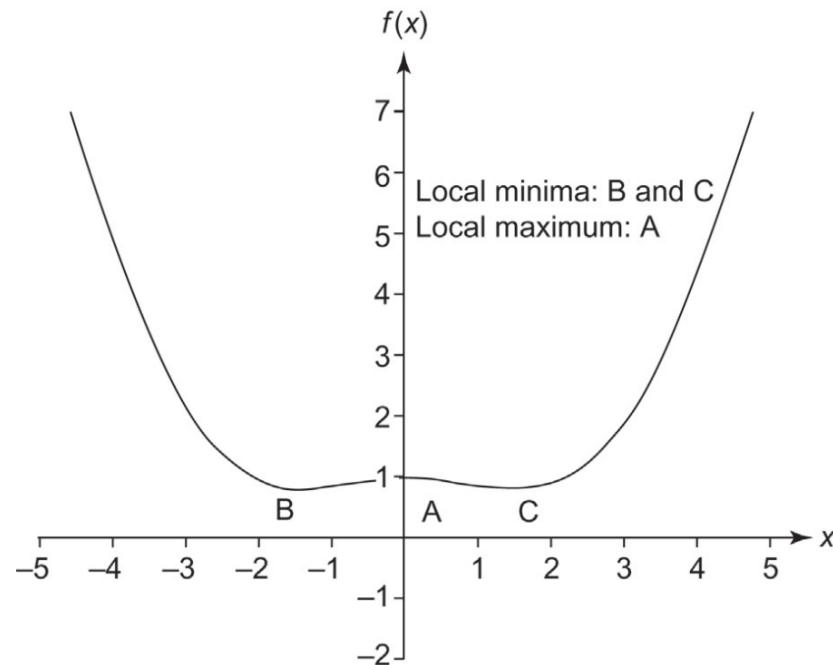
4.5 PROBLEMA SEM RESTRIÇÕES

Para determinar se eles são mínimos locais, máximos ou pontos de inflexão, devemos determinar f'' nos pontos estacionários e usar as condições suficientes.

1. $x^* = 0$; $f''(0) = -1/3 < 0$, então este é um máximo local com $f(0) = 1$.
2. $x^* = 1,496$; $f''(1,496) = 0,592 > 0$, então este é um mínimo local com $f(1,496) = 0,821$.
3. $x^* = -1,496$; $f''(-1,496) = 0,592 > 0$, então este é um mínimo local com $f(-1,496) = 0,821$.

4.5 PROBLEMA SEM RESTRIÇÕES

Observe que $x^* = 1,496$ e $x^* = -1,496$ são na verdade pontos **mínimos globais** para a função, embora a função seja ilimitada e o conjunto não seja fechado. Embora as condições do Teorema de Weierstrass não sejam atendidas, a função possui pontos de mínimos globais. Observe também que não há ponto máximo global para a função, pois a função é ilimitada e x pode ter qualquer valor.



4.5 PROBLEMA SEM RESTRIÇÕES

EXEMPLO 4: Encontre um ponto de mínimo local para a função:

$$f(\mathbf{x}) = x_1 + \frac{(4 \times 10^6)}{x_1 x_2} + 250x_2$$

4.5 PROBLEMA SEM RESTRIÇÕES

As condições necessárias ($\nabla f = 0$) para o problema são:

$$\frac{\partial f}{\partial x_1} = 0; \quad 1 - \frac{(4 \times 10^6)}{x_1^2 x_2} = 0$$

$$\frac{\partial f}{\partial x_2} = 0; \quad 250 - \frac{(4 \times 10^6)}{x_1 x_2^2} = 0$$

$$x_1^2 x_2 - (4 \times 10^6) = 0; \quad 250 x_1 x_2^2 - (4 \times 10^6) = 0$$

Como nem x_1 nem x_2 podem ser zero, a equação anterior dá $x_1^* = 1000$ e $x_2^* = 4$ é um ponto estacionário para a função $f(\mathbf{x})$.

4.5 PROBLEMA SEM RESTRIÇÕES

Para verificar se o ponto estacionário é um mínimo local, avaliamos $\mathbf{H}$ em $\mathbf{x}^*$ diferenciando novamente os componentes do gradiente:

$$\frac{\partial^2 f}{\partial x_1^2} = 2 \times \frac{4 \times 10^6}{x_1^3 x_2}; \quad \frac{\partial^2 f}{\partial x_2 \partial x_1} = \frac{4 \times 10^6}{x_1^2 x_2^2}; \quad \frac{\partial^2 f}{\partial x_2^2} = 2 \times \frac{4 \times 10^6}{x_1 x_2^3}$$

$$\mathbf{H} = \frac{(4 \times 10^6)}{x_1^2 x_2^2} \begin{bmatrix} \frac{2x_2}{x_1} & 1 \\ 1 & \frac{2x_1}{x_2} \end{bmatrix}; \quad \mathbf{H}(1000, 4) = \frac{(4 \times 10^6)}{(4000)^2} \begin{bmatrix} 0.008 & 1 \\ 1 & 500 \end{bmatrix}$$

4.5 PROBLEMA SEM RESTRIÇÕES

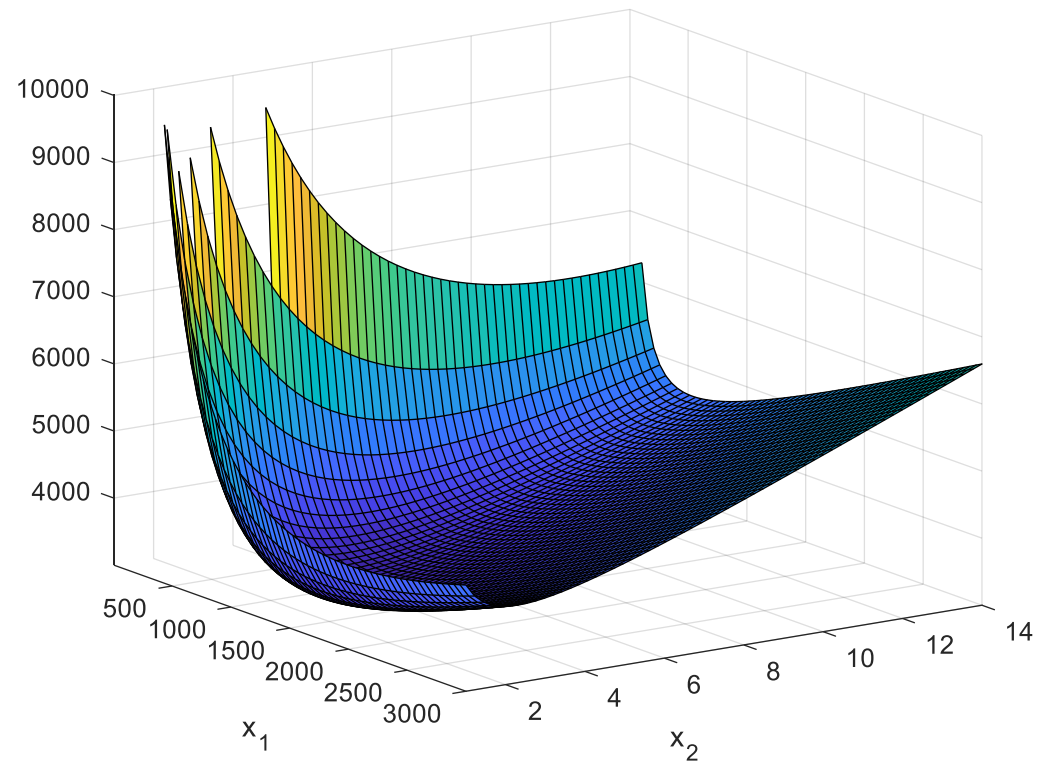
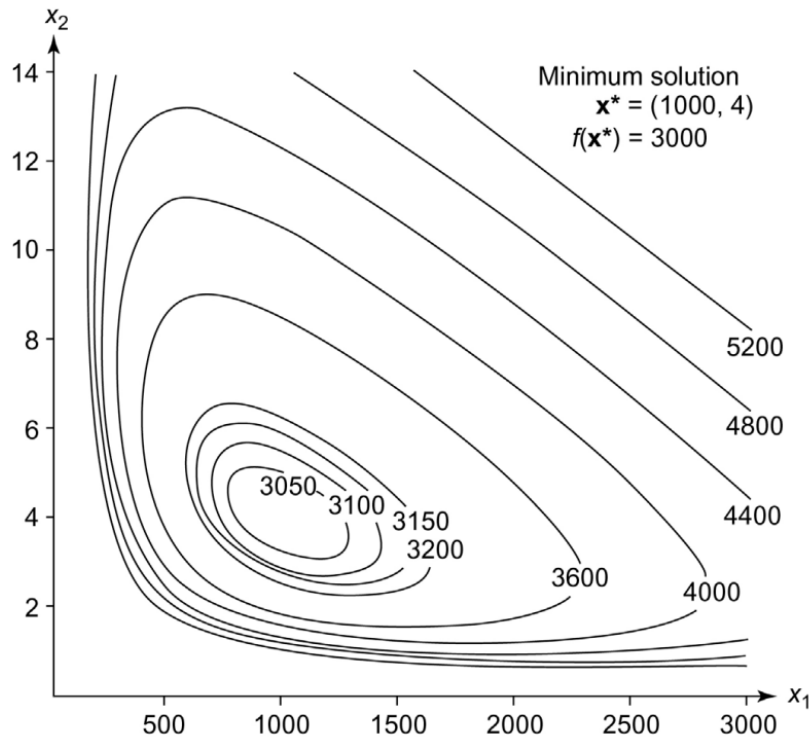
Para verificar a forma da Hessiana, calculamos autovalores de $\mathbf{H}$:

$$\begin{vmatrix} 0.002 - \lambda & 0.25 \\ 0.25 & 125 - \lambda \end{vmatrix} = (0.002 - \lambda)(125 - \lambda) - 0.25^2 = 0$$

Uma vez que ambos os autovalores são positivos ($\lambda_1 = 0,0015$ e $\lambda_2 = 125$), a Hessiana $\mathbf{H}$ é positivo definido. **Condição suficiente de 2ª** ordem para um mínimo local. Portanto: $\mathbf{x}^* = (1000, 4)$ satisfaz a condição de suficiência para um ponto de mínimo local com $f(\mathbf{x}^*) = 3000$.

4.5 PROBLEMA SEM RESTRIÇÕES

A Figura mostra algumas curvas de isocusto para a função do problema. Vê-se que o ponto $(1000; 4)$ é o mínimo para a função.



4.6 CONDIÇÕES DE OTIMIDADE: PROBLEMA COM RESTRIÇÕES DE IGUALDADE

4.6.1 Multiplicadores de Lagrange

Ponto Regular: Considere o problema de otimização restrita de minimizar $f(\mathbf{x})$ sujeito às restrições $h_i(\mathbf{x}) = 0$, $i = 1$ a p . Um ponto $\mathbf{x}^*$ satisfazendo as restrições $h(\mathbf{x}^*) = 0$ é chamado ponto regular do conjunto viável se $f(\mathbf{x}^*)$ for diferenciável e os vetores de gradiente de todas as restrições no ponto $\mathbf{x}^*$ forem linearmente independentes.

- Independência linear significa que não há dois gradientes paralelos entre si e nenhum gradiente pode ser expresso como uma combinação linear dos outros.

4.6 PROBLEMA COM RESTRIÇÕES

As restrições de igualdade ($g_i=0$) estão sempre ativas para qualquer projeto viável, enquanto uma restrição de desigualdade ($g_j>0$) pode não estar ativa em um ponto viável.

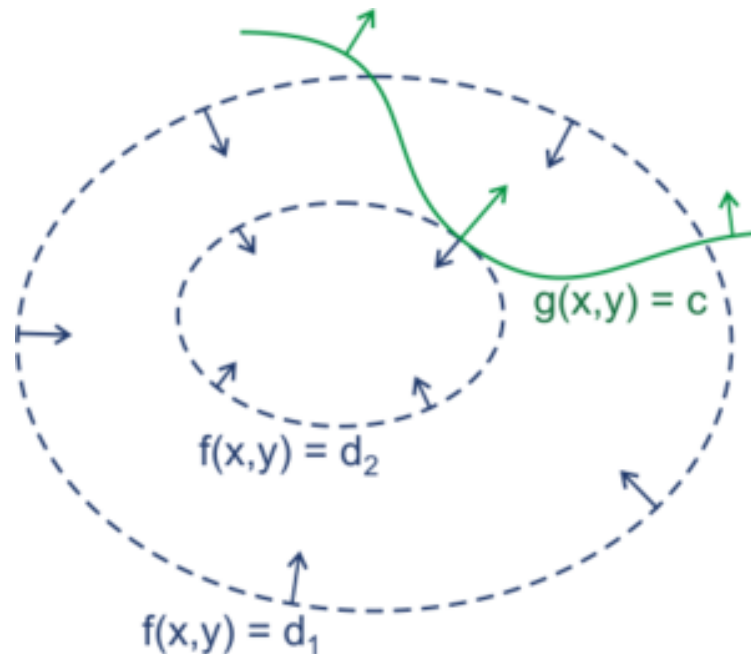
As condições necessárias para um problema com restrição de igualdade estão contidas no teorema do multiplicador de Lagrange.

TABLE 4.2 General Design Optimization Model

Design variable vector	$\mathbf{x} = (x_1, x_2, \dots, x_n)$
Cost function	$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n) \quad (4.35)$
Equality constraints	$h_i(\mathbf{x}) = 0; \quad i = 1 \text{ to } p \quad (4.36)$
Inequality constraints	$g_i(\mathbf{x}) \leq 0; \quad i = 1 \text{ to } m \quad (4.37)$

4.6 PROBLEMA COM RESTRIÇÕES

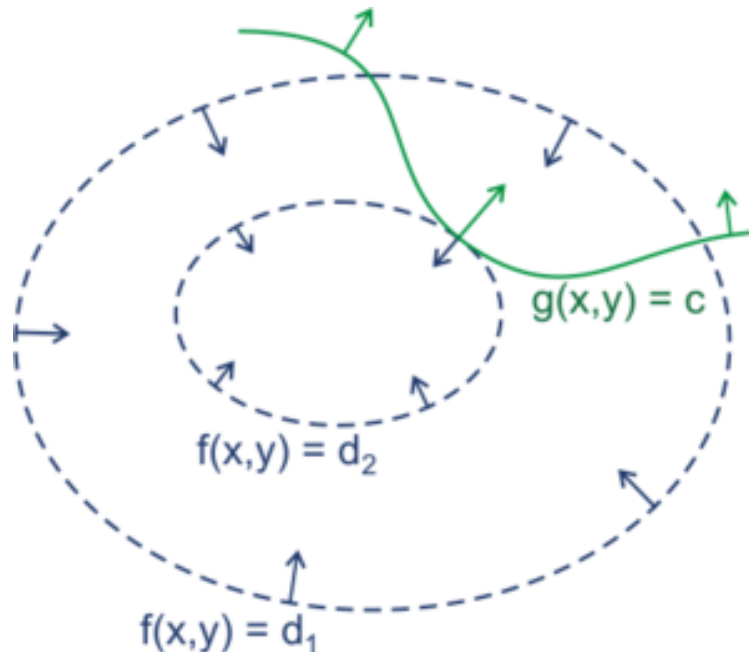
A ideia principal é procurar por pontos nos quais as linhas de contorno de f (*função custo*) e g (*restrições*), sejam tangentes entre si. Isso é o mesmo que encontrar pontos nos quais os vetores gradiente de $f(\mathbf{x})$ e $g(\mathbf{x})$ sejam paralelos entre si, ou seja, o gradiente de uma certa função, chamada de LAGRANGEANA, deve ser igual ao vetor nulo.



4.6 PROBLEMA COM RESTRIÇÕES

Quando dois vetores apontam na mesma direção significa que podemos multiplicar um por alguma constante (λ_0 ou ν) para obter o outro. Por exemplo, para 2 variáveis:

$$\nabla f(x_0, y_0) = \lambda_0 \nabla g(x_0, y_0), \lambda_0 \in \mathbb{R}$$



4.6 PROBLEMA COM RESTRIÇÕES

Nos anos 1700, *Joseph Louis Lagrange* escreveu uma nova função especial que engloba todas as variáveis de entrada de f e g e a constante (λ_0 ou ν). A função é denominado de **Lagraneano** e λ_0 de multiplicador de Lagrange.

$$\mathcal{L}(x_0, y_0, \lambda) = f(x_0, y_0) + \lambda(g(x_0, y_0) - c)$$

Teorema do Multiplicador de Lagrange

A função Lagrange é denotada como L e impõe que para $\mathbf{x}^*$ ser candidato a mínimo (ponto regular) de f , as seguintes condições sejam satisfeitas.

Minimize $f(\mathbf{x})$ sujeita a restrição de igualdade: $h_i(\mathbf{x}) = 0, i = 1$ to p

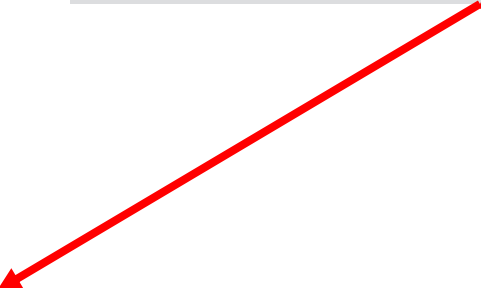
$$\frac{\partial f(\mathbf{x}^*)}{\partial x_i} + \sum_{j=1}^p v_j^* \frac{\partial h_j(\mathbf{x}^*)}{\partial x_i} = 0; i = 1 \text{ to } n$$

$$h_j(\mathbf{x}^*) = 0; j = 1 \text{ to } p$$

Teorema do Multiplicador de Lagrange

É conveniente escrever essas condições em termos de uma função de Lagrange, definida como:

$$\begin{aligned} L(\mathbf{x}, \mathbf{v}) &= f(\mathbf{x}) + \sum_{j=1}^p v_j h_j(\mathbf{x}) \\ &= f(\mathbf{x}) + \mathbf{v}^T \mathbf{h}(\mathbf{x}) \end{aligned}$$


$$\frac{\partial L(\mathbf{x}^*, \mathbf{v}^*)}{\partial x_i} = 0; \quad i = 1 \text{ to } n$$


$$\frac{\partial L(\mathbf{x}^*, \mathbf{v}^*)}{\partial v_j} = 0 \Rightarrow h_j(\mathbf{x}^*) = 0;$$

Teorema do Multiplicador de Lagrange

Condição NECESSÁRIA: Um ponto satisfazendo as condições do teorema é um candidato a ponto mínimo.

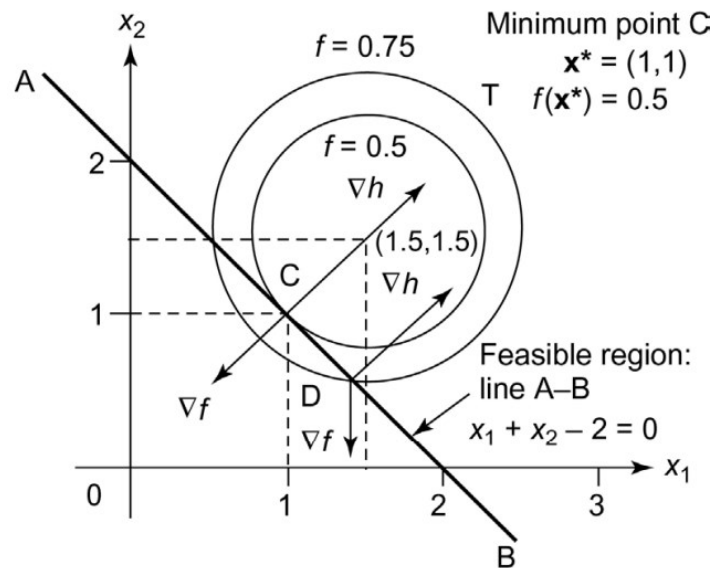
Frequentemente, as condições necessárias do teorema do multiplicador de Lagrange levam a um conjunto não linear de equações que não podem ser resolvidas analiticamente. Nesses casos, devemos usar o método de Newton-Raphson, para resolver. Vários pacotes de software comercial, como Excel e MATLAB devem ser utilizados.

4.5 PROBLEMA SEM RESTRIÇÕES

EXEMPLO 1: Minimizar a função f sujeita a restrição h :

$$f(x_1, x_2) = (x_1 - 1.5)^2 + (x_2 - 1.5)^2$$

$$h(x_1, x_2) = x_1 + x_2 - 2 = 0$$



4.5 PROBLEMA SEM RESTRIÇÕES

$$L(\mathbf{x}, \mathbf{v}) = f(\mathbf{x}) + \sum_{j=1}^p v_j h_j(\mathbf{x})$$

$$L(\mathbf{x}, v) = (x_1 - 1,5)^2 - (x_2 - 1,5)^2 + v(x_1 + x_2 - 2)$$

$$\left\{ \begin{array}{l} \frac{\partial L(x_1^*, x_2^*)}{\partial x_1} = 0, \quad = 2(x_1 - 1,5) + v = 0 \\ \frac{\partial L(x_1^*, x_2^*)}{\partial x_2} = 0 \quad = 2(x_2 - 1,5) + v = 0 \\ \frac{\partial L(\mathbf{x}^*, \mathbf{v}^*)}{\partial v_j} = 0 \quad = (x_1 + x_2 - 2) = 0 \end{array} \right.$$

Resolvendo o sistema de equação.

$$x_1^* = 1$$

$$x_2^* = 1$$

$$v^* = 1$$

4.5 PROBLEMA SEM RESTRIÇÕES

$$f(x_1, x_2) = (x_1 - 1.5)^2 + (x_2 - 1.5)^2 \quad h(x_1, x_2) = x_1 + x_2 - 2 = 0$$

$$\nabla f(1, 1) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad \nabla h(1, 1) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

**Obedece a condição necessária!
É um candidato a mínimo!**

$$\nabla f(\mathbf{x}^*) = -v \nabla h(\mathbf{x}^*)$$

4.7 CONDIÇÕES DE OTIMIDADE: PROBLEMA COM RESTRIÇÕES DE DESIGUALDADE

RESTRIÇÕES DE DESIGUALDADE

Podemos transformar uma restrição de desigualdade em uma igualdade adicionando uma variável de folga s_i :

$$s_i = g_i \leq 0 \Rightarrow g_i + s_i = 0$$

- Se $s_i^2 = 0$ a restrição é ATIVA (o ponto ótimo está localizado em uma das restrições, $g_i(\mathbf{x}^*) = 0$).
- Se $s_i^2 > 0$ a restrição é INATIVA (o ponto ótimo não fica sobre uma das restrições, $g_i(\mathbf{x}^*) < 0$).
- Se $s_i^2 < 0$ ($g_i(\mathbf{x}^*) > 0$) a restrição é VIOLADA e o ponto ótimo é inviável.

RESTRIÇÕES DE DESIGUALDADE

Por conveniência fazemos $g_i + s_i^2 = 0$ e então usamos o teorema dos multiplicadores de Lagrange para gerar as condições de otimalidade.

- Os multiplicadores de Lagrange das restrições de desigualdade ($\mathbf{u}$) devem ser sempre $u_i \geq 0$, $i=1, \dots, m$.
- Se s_i é ATIVA então $u_i > 0$.
- Se s_i é INATIVA então $u_i = 0$.

Obs.: Ou u_i ou s_i será igual a zero. Dificilmente, $u_i = s_i = 0$.

RESTRIÇÕES DE DESIGUALDADE

Esse Exemplo ilustra o uso das condições necessárias para um problema com restrição de desigualdade.

EXEMPLO 1: Minimizar a função f sujeita a restrição g :

$$f(x_1, x_2) = (x_1 - 1.5)^2 + (x_2 - 1.5)^2$$

$$g(x) = x_1 + x_2 - 2 \leq 0$$

RESTRIÇÕES DE DESIGUALDADE

Solução: $f(x_1, x_2) = (x_1 - 1.5)^2 + (x_2 - 1.5)^2$ $g(x) = x_1 + x_2 - 2 \leq 0$

1º Passo: Construir a função Lagrangeana considerando a folga s^2 .

$$L = (x_1 - 1.5)^2 + (x_2 - 1.5)^2 + u(x_1 + x_2 - 2 + s^2)$$

onde u é o multiplicador de Lagrange para a restrição de desigualdade.

RESTRIÇÕES DE DESIGUALDADE

2º Passo: As condições necessárias do teorema de Lagrange fornecem (tratando x_1 , x_2 , u e s como incógnitas):

$$\frac{\partial L}{\partial x_1} = 2(x_1 - 1.5) + u = 0$$

$$\frac{\partial L}{\partial x_2} = 2(x_2 + 1.5) + u = 0$$

$$\frac{\partial L}{\partial u} = x_1 + x_2 - 2 + s^2 = 0$$

$$\frac{\partial L}{\partial s} = 2us = 0$$

Note que as equações são não lineares. Portanto, eles podem ter várias raízes.

Uma solução pode ser obtida definindo s como zero para satisfazer a chamada condição de comutação $2us = 0$.

RESTRIÇÕES DE DESIGUALDADE

3º Passo: Resolver as equações restantes para $s = 0$:

$$\frac{\partial L}{\partial x_1} = 2(x_1 - 1.5) + u = 0$$

$$\frac{\partial L}{\partial x_2} = 2(x_2 + 1.5) + u = 0$$

$$\frac{\partial L}{\partial u} = x_1 + x_2 - 2 + s^2 = 0$$

Resolvendo as equações restantes, considerando $s = 0$, obtemos:

$$x_1^* = x_2^* = 1;$$

$$u^* = 1 > 0;$$

$$s = 0.$$

Este é um ponto estacionário de L , portanto é um candidato a ponto mínimo.

RESTRIÇÕES DE DESIGUALDADE

4º Passo: Resolver as equações restantes para $u = 0$:

$$\frac{\partial L}{\partial x_1} = 2(x_1 - 1.5) + u = 0$$

$$\frac{\partial L}{\partial x_2} = 2(x_2 + 1.5) + u = 0$$

$$\frac{\partial L}{\partial u} = x_1 + x_2 - 2 + s^2 = 0$$

Resolvendo as equações restantes, considerando $u = 0$, obtemos:

$$x_1^* = x_2^* = 1,5;$$

$$s^2 = -1;$$

$$u^* = 0.$$

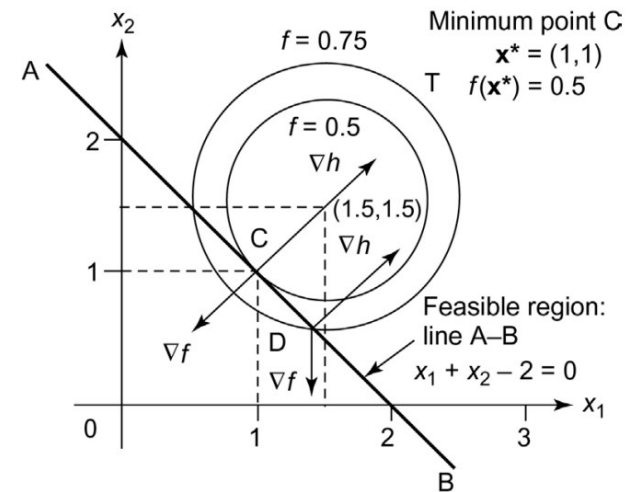
Esta não é uma solução válida, pois Se $s_i^2 < 0$ a restrição é VIOLADA e o ponto ótimo é inviável.

RESTRIÇÕES DE DESIGUALDADE

5º Passo: Determinar os gradientes das funções de custo e restrição no ponto candidato (1, 1):

$$\nabla f = \begin{bmatrix} 2(x_1 - 1.5) \\ 2(x_2 - 1.5) \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}; \quad \nabla g = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\nabla f(\mathbf{x}^*) = -v \nabla h(\mathbf{x}^*)$$



Esses gradientes estão ao longo da mesma linha, mas em direções opostas, conforme mostrado na Figura. Observe também que qualquer pequeno movimento do ponto C aumenta a função de custo ou leva o projeto para a região inviável. Assim, o ponto (1, 1) é de fato um ponto de mínimo local.

RESTRIÇÕES EM GERAL

As condições necessárias para o problema com restrições de igualdade e desigualdade escritas na forma padrão, pode ser resumido no que é comumente conhecidas como as condições necessárias de primeira ordem de Karush–Kuhn–Tucker (KKT):

RESTRIÇÕES EM GERAL

Condições de otimização de Karush-Kuhn-Tucker (KKT):

Seja $\mathbf{x}^*$ um ponto regular do conjunto viável que é um mínimo local para $f(\mathbf{x})$, sujeito a $h_i(\mathbf{x}) = 0; i = 1$ a $p; g_j(\mathbf{x}) \leq 0; j = 1$ a m . Então existem multiplicadores de Lagrange $\mathbf{v}^*$ (um vetor p) e $\mathbf{u}^*$ (um vetor m) tais que a função Lagrangeana é estacionária em relação a x_j, v_i, u_j e s_j no ponto $\mathbf{x}^*$.

RESTRIÇÕES EM GERAL

1. Função lagrangiana para o problema escrito na forma padrão:

$$\begin{aligned} L(\mathbf{x}, \mathbf{v}, \mathbf{u}, \mathbf{s}) &= f(\mathbf{x}) + \sum_{i=1}^p v_i h_i(\mathbf{x}) + \sum_{j=1}^m u_j (g_j(\mathbf{x}) + s_j^2) \\ &= f(\mathbf{x}) + \mathbf{v}^T \mathbf{h}(\mathbf{x}) + \mathbf{u}^T (\mathbf{g}(\mathbf{x}) + \mathbf{s}^2) \end{aligned}$$

2. Condições de gradiente:

$$\frac{\partial L}{\partial x_k} = \frac{\partial f}{\partial x_k} + \sum_{i=1}^p v_i^* \frac{\partial h_i}{\partial x_k} + \sum_{j=1}^m u_j^* \frac{\partial g_j}{\partial x_k} = 0; \quad k = 1 \text{ to } n$$

$$\frac{\partial L}{\partial v_i} = 0 \Rightarrow h_i(\mathbf{x}^*) = 0; \quad i = 1 \text{ to } p$$

$$\frac{\partial L}{\partial u_j} = 0 \Rightarrow (g_j(\mathbf{x}^*) + s_j^2) = 0; \quad j = 1 \text{ to } m$$

RESTRIÇÕES EM GERAL

3. Verificação de viabilidade para desigualdades:

$$s_j^2 \geq 0; \text{ or equivalently } g_j \leq 0; \quad j = 1 \text{ to } m$$

4. Condições de comutação:

$$\frac{\partial L}{\partial s_j} = 0 \Rightarrow 2u_j^* s_j = 0; \quad j = 1 \text{ to } m$$

RESTRIÇÕES EM GERAL

5. Não negatividade dos multiplicadores de Lagrange para desigualdades:

$$u_j^* \geq 0; \quad j = 1 \text{ to } m$$

6. Verificação de regularidade: Gradientes das restrições ativas devem ser linearmente independentes. Nesse caso, os multiplicadores de Lagrange para as restrições são únicos.

4.8 Efeito da Alteração dos Limites de Restrição

***sensibilidade do custo $f(x)$ em relação as restrições**

Alteração dos Limites de Restrição

Vamos assumir que o problema de minimização foi resolvido com $h_i(\mathbf{x}) = 0$ e $g_j(\mathbf{x}) \leq 0$, ou seja, com os valores limites atuais para as restrições como zero.

Queremos saber o que acontece com o valor ótimo da função de custo se uma restrição é relaxada ou uma restrição precisa ser reforçada.

Alteração dos Limites de Restrição

Para discutir mudanças na função de custo devido a mudanças nos limites de restrição, consideramos o problema modificado:

Minimize

$$f(\mathbf{x})$$

subject to

$$h_i(\mathbf{x}) = b_i; \quad i = 1 \text{ to } p$$

$$g_j(\mathbf{x}) \leq e_j; \quad j = 1 \text{ to } m$$

onde b_i e e_j são pequenas variações na vizinhança de zero. É claro que o ponto ótimo para o problema perturbado é função dos vetores b e e e pode ser escrita como $\mathbf{x}^* = \mathbf{x}^*(b, e)$, logo $f^* = f^*(b, e)$.

Alteração dos Limites de Restrição

Teorema de sensibilidade de variação de restrição:

Seja $f(\mathbf{x})$, $h_i(\mathbf{x})$, $i = 1$ to p , e $g_j(\mathbf{x})$, $j = 1$ to m tenham duas derivadas contínuas. Seja $\mathbf{x}^*$ um ponto regular que, juntamente com os multiplicadores v_i^* e u_j^* , satisfaz tanto as condições necessárias KKT quanto as condições suficientes (apresentadas em um capítulo posterior); Pode-se mostrar que em alguma vizinhança de $\mathbf{b} = \mathbf{0}$, $\mathbf{e} = \mathbf{0}$:

$$\frac{\partial f^*}{\partial b_i} = \frac{\partial f(\mathbf{x}^*(\mathbf{0}, \mathbf{0}))}{\partial b_i} = -v_i^*; \quad i = 1 \text{ to } p$$

$$\frac{\partial f^*}{\partial e_j} = \frac{\partial f(\mathbf{x}^*(\mathbf{0}, \mathbf{0}))}{\partial e_j} = -u_j^*; \quad j = 1 \text{ to } m$$

Alteração dos Limites de Restrição

Teorema de sensibilidade de variação de restrição:

Usando o teorema, podemos estimar mudanças na função de custo se decidirmos ajustar o lado direito das restrições na vizinhança de zero. Para tanto, pode-se utilizar a expansão de Taylor. Expansão de Taylor de primeira ordem para a função de custo $f(b_i, e_j)$ sobre o ponto $b_i = 0$ e $e_j = 0$ é dado como:

$$f(b_i, e_j) = f^*(0, 0) + \frac{\partial f^*(0, 0)}{\partial b_i} b_i + \frac{\partial f^*(0, 0)}{\partial e_j} e_j$$

$$f(b_i, e_j) = f^*(0, 0) - v_i^* b_i - u_j^* e_j$$

onde $f(0, 0)$ foi o valor ótimo da função de custo que foi obtido com $b_i = 0$ e $e_j = 0$.

Alteração dos Limites de Restrição

Teorema de sensibilidade de variação de restrição:

uma mudança de primeira ordem na função de custo df devido a pequenas mudanças em b_i e e_j é dada como:

$$\delta f^* = f(b_i, e_j) - f^*(0, 0) = -v_i^* b_i - u_j^* e_j$$

Para valores dados de b_i e e_j , podemos estimar o novo valor da função de custo. Além disso, vê-se que os multiplicadores dão o benefício de afrouxar uma restrição ou a penalidade associada a aumentá-la; o relaxamento aumenta o conjunto viável, enquanto o estreitamento o contrai.

4.9. OTIMIZAÇÃO GLOBAL

OTIMIZAÇÃO GLOBAL

1. Se a função custo $f(\mathbf{x})$ é contínua em um conjunto viável fechado e limitado, então o Teorema de Weierstrass garante a existência de um mínimo global. Portanto, se calcular todos os pontos mínimos locais para a função, o ponto que dá o menor valor para a função de custo pode ser selecionado como um mínimo global para a função (busca exaustiva do espaço de projeto viável).
2. Se o problema de otimização é convexo, então qualquer mínimo local também é um mínimo global. Além disso, as condições necessárias do KKT são necessárias e suficientes para o ponto mínimo.

OTIMIZAÇÃO GLOBAL

CONJUNTO CONVEXO:

Um conjunto convexo S é uma coleção de pontos (vetores $\mathbf{x}$) com a seguinte propriedade: se P_1 e P_2 são quaisquer pontos em S , então todo o segmento de linha P_1 – P_2 também está em S . Isso é necessário e condição suficiente para convexidade do conjunto S :

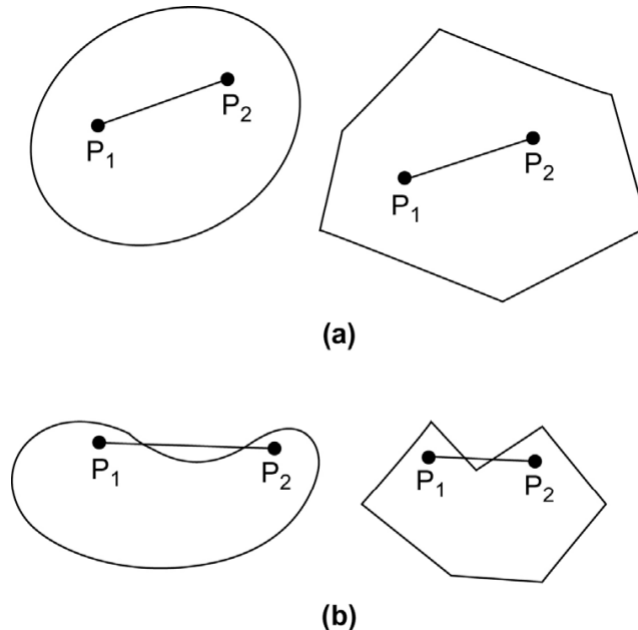


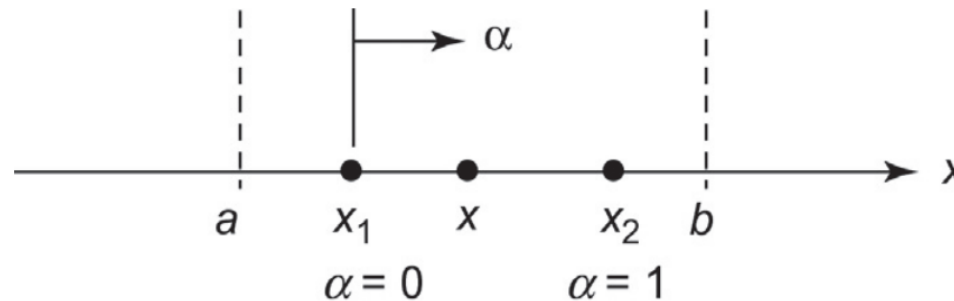
FIGURE 4.19 (a) Convex sets. (b) Nonconvex sets.

OTIMIZAÇÃO GLOBAL

CONJUNTO CONVEXO:

Para mostrar que é um conjunto convexo, sejam x_1 e x_2 dois pontos no intervalo. O segmento de linha entre os pontos pode ser escrito como:

$$x = \alpha x_2 + (1 - \alpha)x_1; \quad 0 \leq \alpha \leq 1$$



OTIMIZAÇÃO GLOBAL

CONJUNTO CONVEXO:

Para mostrar que é um conjunto convexo, sejam x_1 e x_2 dois pontos no intervalo. O segmento de linha entre os pontos pode ser escrito como:

$$x = \alpha x_2 + (1 - \alpha)x_1; \quad 0 \leq \alpha \leq 1$$

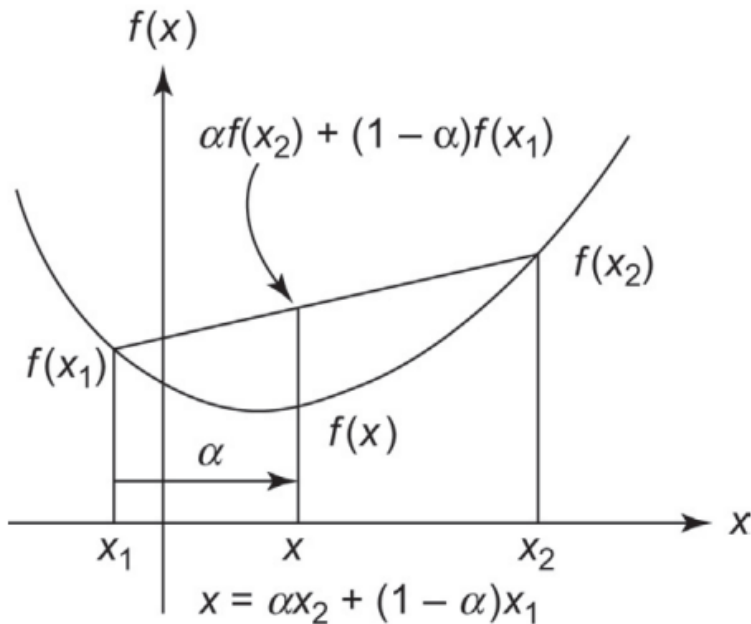
Em geral, para o espaço n -dimensional, o segmento de linha entre quaisquer dois pontos $\mathbf{x}^{(1)}$ e $\mathbf{x}^{(2)}$ é escrito como:

$$\mathbf{x} = \alpha \mathbf{x}^{(2)} + (1 - \alpha)\mathbf{x}^{(1)}; \quad 0 \leq \alpha \leq 1$$

OTIMIZAÇÃO GLOBAL

FUNÇÕES CONVEXAS:

A função f é convexa no conjunto S convexo se o gráfico da função ficar abaixo da linha ligando dois pontos quaisquer de f :



OTIMIZAÇÃO GLOBAL

FUNÇÕES CONVEXAS:

Uma função $f(\mathbf{x})$ definida em um conjunto convexo S é convexa se satisfaz a desigualdade, para quaisquer dois pontos $\mathbf{x}^{(1)}$ e $\mathbf{x}^{(2)}$ em S .

$$f(\alpha \mathbf{x}^{(2)} + (1 - \alpha) \mathbf{x}^{(1)}) \leq \alpha f(\mathbf{x}^{(2)}) + (1 - \alpha) f(\mathbf{x}^{(1)}) \quad \text{for } 0 \leq \alpha \leq 1$$

Essa equação é difícil de usar na prática porque temos que verificar um número infinito de pares de pontos. Felizmente, o seguinte teorema fornece uma maneira mais fácil de verificar a convexidade de uma função.

OTIMIZAÇÃO GLOBAL

TEOREMA PARA A VERIFICAÇÃO DA CONVEXIDADE DE UMA FUNÇÃO:

Uma função de n variáveis $f(x_1, x_2, \dots, x_n)$ definida em um conjunto convexo S é convexa se e somente se a matriz H hessiana da função é positiva semi-definida (**autovalores ≥ 0**) ou positiva definida (**autovalores > 0**) em todos os pontos do conjunto S .

OTIMIZAÇÃO GLOBAL

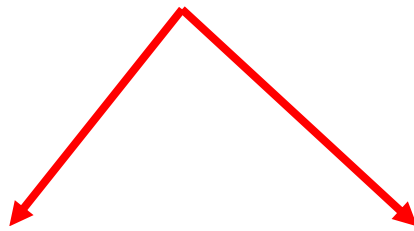
EXEMPLO 1: VERIFICAR A CONVEXIDADE DE UMA FUNÇÃO:

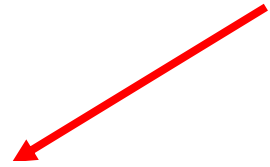
$$f(x) = x_1^2 + x_2^2 - 1$$

OTIMIZAÇÃO GLOBAL

EXEMPLO 1: VERIFICAR A CONVEXIDADE DE UMA FUNÇÃO:

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


$$\nabla f = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

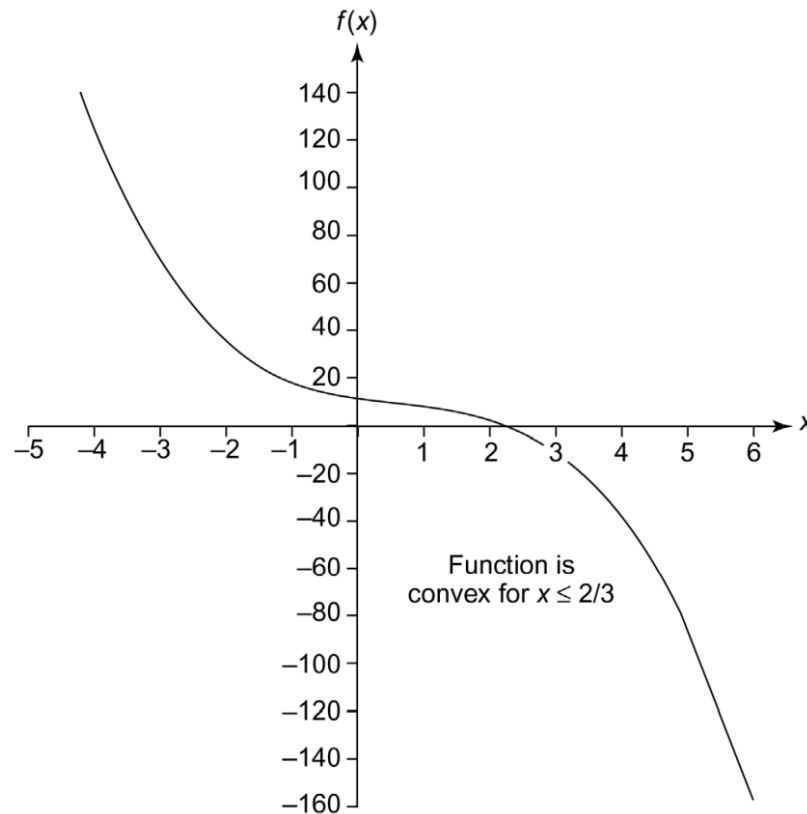

$$\lambda_1 = 2, \lambda_2 = 2$$

**H é positiva definida.
Portanto, f é uma
função convexa.**

OTIMIZAÇÃO GLOBAL

EXEMPLO 2: VERIFICAR A CONVEXIDADE DE UMA FUNÇÃO:

$$f(x) = 10 - 4x + 2x^2 - x^3$$



OTIMIZAÇÃO GLOBAL

EXEMPLO 2: VERIFICAR A CONVEXIDADE DE UMA FUNÇÃO:

$$f(x) = 10 - 4x + 2x^2 - x^3$$



$$d^2f/dx^2 = 4 - 6x$$



$$d^2f/dx^2 \geq 0$$



$$4 - 6x \geq 0$$

A função é convexa para $x \leq 2/3$ e côncava para $x \geq 2/3$.

OTIMIZAÇÃO GLOBAL

PROBLEMA DE PROGRAMAÇÃO CONVEXA:

Seja o conjunto viável S definido pelas restrições do problema geral de otimização, como:

$$S = \{x \mid h_i(x) = 0, i = 1 \text{ to } p; \quad g_j(x) \leq 0, j = 1 \text{ to } m\}$$

Então S é um conjunto convexo se as funções g_j são convexas e as funções h_i são lineares.

OTIMIZAÇÃO GLOBAL

PROBLEMA DE PROGRAMAÇÃO CONVEXA:

Se todas as funções de restrição de desigualdade ($g_j(\mathbf{x}) \leq 0$) para um problema de projeto ótimo forem convexas e todas as restrições de igualdade forem lineares ($h_i(\mathbf{x}) = 0$), então o conjunto viável S é convexo. Se a função de custo ($f(\mathbf{x})$) também for convexa sobre o conjunto S , temos o que é conhecido como problema de programação convexa.

OTIMIZAÇÃO GLOBAL

PROBLEMA DE PROGRAMAÇÃO CONVEXA:

Se todas as funções de restrição de desigualdade ($g_j(\mathbf{x}) \leq 0$) para um problema de projeto ótimo forem convexas e todas as restrições de igualdade forem lineares ($h_i(\mathbf{x}) = 0$), então o conjunto viável S é convexo. Se a função de custo ($f(\mathbf{x})$) também for convexa sobre o conjunto S , temos o que é conhecido como problema de programação convexa.

Tais problemas têm uma propriedade muito útil, que é que as condições necessárias de KKT também são **SUFICIENTES** e **qualquer mínimo local também é um mínimo global**.

OTIMIZAÇÃO GLOBAL

PROBLEMA DE PROGRAMAÇÃO CONVEXA:

É importante notar que o teorema não diz que x^* não pode ser um ponto de mínimo global se as funções do problema falharem no teste de convexidade. Caso falhe, teremos que usar algum outro procedimento, como a busca exaustiva.

Note também que o teorema não diz que o mínimo global é único; ou seja, pode haver vários pontos mínimos no conjunto viável, todos com o mesmo valor de função de custo.

OTIMIZAÇÃO GLOBAL

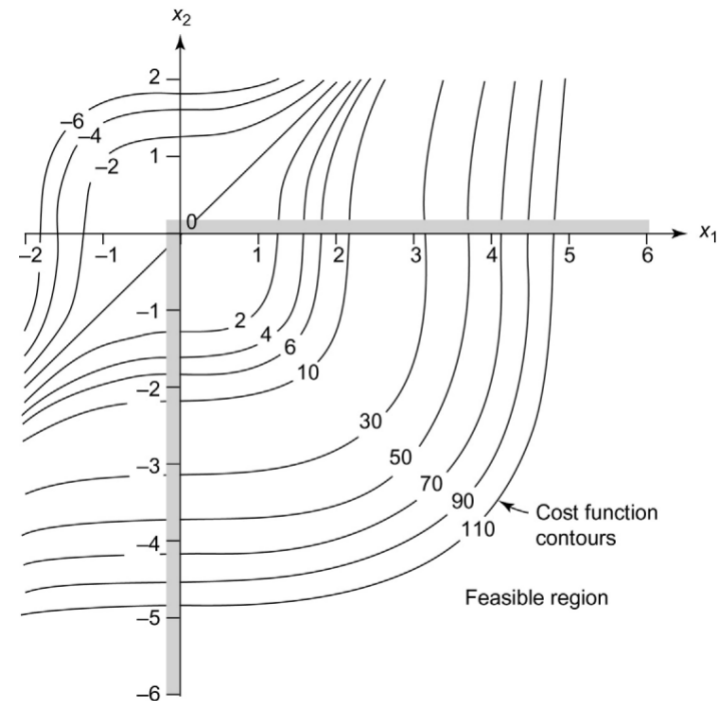
EXEMPLO 1: VERIFICAR A CONVEXIDADE DE UMA FUNÇÃO:

Minimize

$$f(x_1, x_2) = x_1^3 - x_2^3$$

subject to

$$x_1 \geq 0, \quad x_2 \leq 0$$



OTIMIZAÇÃO GLOBAL

EXEMPLO 1: VERIFICAR A CONVEXIDADE DE UMA FUNÇÃO:

Minimize

$$f(x_1, x_2) = x_1^3 - x_2^3$$

subject to

$$x_1 \geq 0, \quad x_2 \leq 0$$



A segunda derivada das restrições é igual a ZERO. Logo, autovalores nulos. Portanto, g_1 e g_2 são funções convexas. O Hessiano de f é dado como:

$$\mathbf{H} = \begin{bmatrix} 6x_1 & 0 \\ 0 & -6x_2 \end{bmatrix}$$

O Hessiano é positivo semidefinido ou positivo definido sobre o domínio definido pelas restrições ($x_1 \geq 0$, $x_2 \leq 0$). Portanto, a função de custo é convexa

OTIMIZAÇÃO GLOBAL

EXEMPLO 1: VERIFICAR A CONVEXIDADE DE UMA FUNÇÃO:

Minimize

$$f(x_1, x_2) = 9x_1^2 - 18x_1x_2 + 13x_2^2 - 4$$

subject to

$$x_1^2 + x_2^2 + 2x_1 \geq 16$$

OTIMIZAÇÃO GLOBAL

EXEMPLO 1: VERIFICAR A CONVEXIDADE DE UMA FUNÇÃO:

Minimize

$$f(x_1, x_2) = 9x_1^2 - 18x_1x_2 + 13x_2^2 - 4$$

subject to

$$x_1^2 + x_2^2 + 2x_1 \geq 16$$



Para verificar a convexidade do problema, precisamos escrever a restrição na forma padrão como:

$$g(x) = -x_1^2 - x_2^2 - 2x_1 + 16 \leq 0$$

OTIMIZAÇÃO GLOBAL

EXEMPLO 1: VERIFICAR A CONVEXIDADE DE UMA FUNÇÃO:

$$g(x) = -x_1^2 - x_2^2 - 2x_1 + 16 \leq 0$$

A MATRIZ HESSIANA DE $g(x)$:

$$\mathbf{H} = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix}$$

Os autovalores do Hessiano são -2 e -2 . Como $\mathbf{H}$ é definido negativo, então $g(x)$ é côncavo. Portanto, o problema não pode ser classificado como um problema de programação convexa e a otimalidade global da solução não pode ser garantida.

5. Otimização não linear: SOLUÇÃO NUMÉRICA

SOLUÇÃO NUMÉRICA

1. São necessários devido o grande número de variáveis de projeto e restrições;
2. As funções custo (f), e as restrições (g ; h) podem ser funções implícitas.

SOLUÇÃO NUMÉRICA

1. Até agora, discutimos a formulação de um problema de projeto ótimo, otimização gráfica e condições de otimização. O método gráfico é aplicável somente para problemas de duas variáveis;
2. Condições de Otimalidade tornam-se difíceis de usar quando o número de variáveis e/ou o número de restrições é maior que três. Portanto, métodos numéricos, que podem lidar com muitas variáveis e restrições, bem como procurar diretamente pontos ótimos, foram desenvolvidos;

SOLUÇÃO NUMÉRICA

MÉTODOS NUMÉRICOS:

O propósito de introduzir um programa de otimização numérica antes de apresentar a teoria é permitir que os alunos comecem a trabalhar em projetos que envolvam problemas de otimização do mundo real. Além disso, os alunos podem usar esses programas para verificar suas soluções de lição de casa.

SOLUÇÃO NUMÉRICA

MÉTODOS NUMÉRICOS:

Os métodos NUMÉRICOS são iterativos e cada iteração pode exigir um grande número de cálculos, dependendo da aplicação. Os passos básicos para implementar um iteração de um algoritmo de otimização para problemas suaves são as seguintes:

SOLUÇÃO NUMÉRICA

Os algoritmos podem ser classificados em:

1. Ordem zero (Método de Nelder-Mead: função *fminsearch* do matlab). Este método não usa gradientes da função de custo e, portanto, a continuidade e a diferenciabilidade da função não são necessárias para encontrar um mínimo local da função. A ideia básica do método;
2. 1º Ordem: uso de f , g e h e seus gradientes;
3. 2º Ordem: uso de f , g e h , seus gradientes e matriz hessiana.

SOLUÇÃO NUMÉRICA

A ideia básica dos algoritmos é dar um chute inicial ($\mathbf{x}^0$) e a partir dele progredir/caminhar até a solução do problema ($\mathbf{x}^*$).

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta \mathbf{x}^{(k)}; \quad k = 0, 1, 2, \dots$$

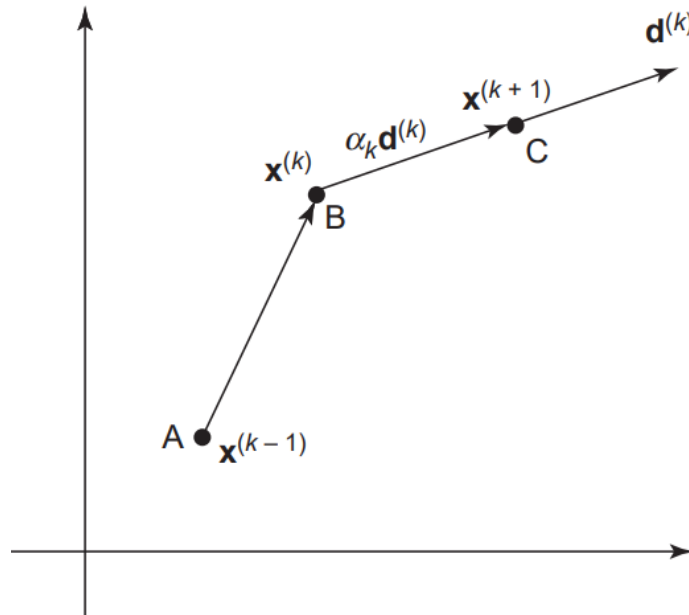
A sequência acima é repetida até os critérios de otimalidade serem satisfeitos ou uma solução aceitável seja encontrada.

SOLUÇÃO NUMÉRICA

O que caracteriza os diferentes algoritmos de otimização é a forma que o incremento $\Delta \mathbf{x}^{(k)}$ é calculado. Para tal, tem-se que:

$$\Delta \mathbf{x}^{(k)} = \alpha_k \mathbf{d}^{(k)}$$

onde $\alpha_k > 0$ é um tamanho de passo na direção de busca $\mathbf{d}^{(k)}$. Assim, a melhoria do projeto envolve a resolução de $\mathbf{d}^{(k)}$ e o cálculo de α_k .



SOLUÇÃO NUMÉRICA

A Direção de decida para um mínimo ocorre quando há uma diminuição de valor da função f .

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

$$f(\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}) < f(\mathbf{x}^{(k)})$$

Aproximando o lado esquerdo por uma expansão de Taylor, tem-se que $\nabla f(\mathbf{x}^{(k)}) \cdot \mathbf{d}^{(k)} < 0$. Isso quer dizer que o ângulo entre ∇f e $\mathbf{d}$ deve estar entre 90° e 270° .

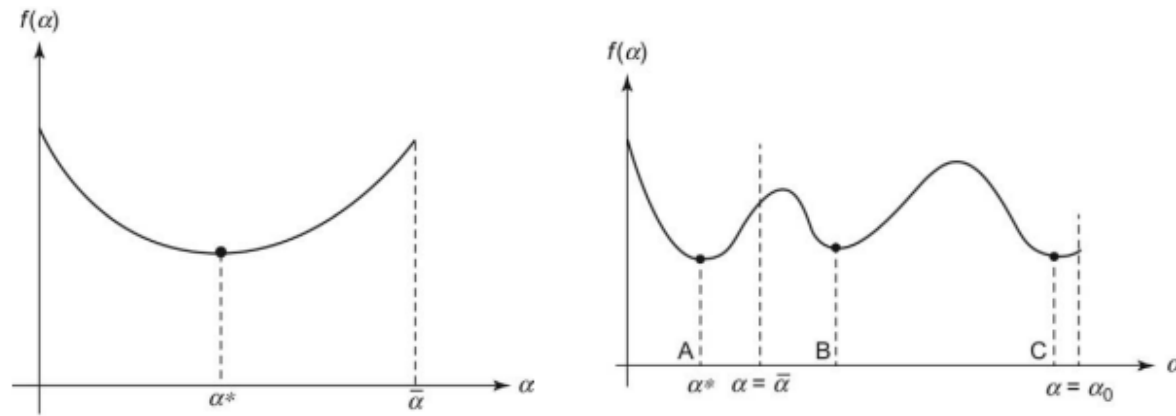
$$f(\mathbf{x}^{(k)}) + \alpha_k (\mathbf{c}^{(k)} \cdot \mathbf{d}^{(k)}) < f(\mathbf{x}^{(k)})$$

$$(\mathbf{c}^{(k)} \cdot \mathbf{d}^{(k)}) < 0$$

Onde $\mathbf{c}^{(k)} = \nabla f(\mathbf{x}^{(k)})$.

SOLUÇÃO NUMÉRICA

O Processo de busca do passo α_k é iterativo. Geralmente, é baseado na comparação de valores da função em diversos pontos na direção de busca $\mathbf{d}^{(k)}$. *Em seguida resolve-se o problema em 1D.*



E pode ser resolvido por diferentes métodos: ***Golden section Search*** (um dos melhores métodos na classe de métodos de redução de intervalo) ou Gradiente conjugado. O **método do gradiente conjugado** é uma modificação muito simples e eficaz do método steepest-descent. O algoritmo é muito simples de programar e funciona muito bem para problemas gerais de minimização irrestrita.

$$\mathbf{d}^{(k)} = -\mathbf{c}^{(k)} + \beta_k \mathbf{d}^{(k-1)}$$

$$\beta_k = \left(\frac{\|\mathbf{c}^{(k)}\|}{\|\mathbf{c}^{(k-1)}\|} \right)^2 = \frac{(\mathbf{c}^{(k)} \cdot \mathbf{c}^{(k)})}{(\mathbf{c}^{(k-1)} \cdot \mathbf{c}^{(k-1)})}$$

SOLUÇÃO NUMÉRICA

Para problemas **SEM RESTRIÇÕES**, pode-se usar também, outras técnicas, tais como:

- interpolação polinomial para cálculo do tamanho do passo;
- pesquisa de linha inexata (regra de Armijo, Condições Wolfe, Teste de Goldstein);
- método de Newton que usa o Hessiano da função de custo em otimização numérica (com Marquardt Modification, para acelerar o processor);
- Métodos aproximados de segunda ordem (métodos de quase Newton: com BFGS ou DFP, para acelerar o processor),
- Métodos de transformação que transformam um problema restrito em um problema irrestrito, de modo que os métodos de otimização irrestritos possam ser usados para resolver problemas restritos.

SOLUÇÃO NUMÉRICA

Para problemas **SEM RESTRIÇÕES**, pode-se usar também, outras técnicas, tais como:

- métodos de pesquisa direta: métodos que não requerem derivadas das funções em sua estratégia de busca. Isso significa que os métodos podem ser usados para problemas em que as derivadas são caras para calcular ou não estão disponíveis devido à falta de diferenciabilidade das funções. No entanto, a convergência dos métodos pode ser provada se as funções forem assumidas como contínuas e diferenciáveis. os métodos tratam apenas problemas de otimização sem restrições; no entanto, as restrições podem ser incluídas usando a abordagem da função de penalidade.
1. Método simplex Nelder-Mead (não confundir com o método Simplex de programação linear);
 2. Método Hooke-Jeeves.

SOLUÇÃO NUMÉRICA

Para problemas **COM RESTRIÇÕES**:

- A maioria dos métodos segue a abordagem em duas fases: 1) a determinação da direção de busca e 2) determinação do tamanho do passo.

SOLUÇÃO NUMÉRICA

Para problemas **COM RESTRIÇÕES**, pode-se usar:

O mais comumente utilizados são:

1. O **ALGORITMO DE PROGRAMAÇÃO LINEAR SEQUENCIAL (SLP)**: Para isso deve-se linearizar em cada iteração as funções de custo e de restrições (método limitado pela falta de robustez);
2. **PROGRAMAÇÃO QUADRÁTICA SEQUENCIAL (SQP)**: ex.: método do gradiente projetado, o método de direcções factíveis (FD) e o método de gradiente reduzido generalizado (GRG).
3. Método restrito **STEEPEST-DESCENT (CSD)**;
4. Entre outros;

SOLUÇÃO NUMÉRICA

Um algoritmo geral para a otimização não linear é:

Passo 1: Estimar $\mathbf{x}^{(0)}$ e fazer $k=0$;

Passo 2: Calcular a direção de busca $\mathbf{d}^{(k)}$;

Passo 3: Verificar a convergência do algoritmo. Se convergiu, pare, se não, continue.

Passo 4: Calcular o passo α_k .

Passo 5: Calcular:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)}; \quad k = 0, 1, 2, \dots \quad \Delta\mathbf{x}^{(k)} = \alpha_k \mathbf{d}^{(k)}$$

Fazer $k=k+1$ e ir para o passo 2.

SOLUÇÃO NUMÉRICA

1. Vários programas de computador comerciais, como o Excel Solver, o MATLAB *Optimization Tool Box* (comando: *optimtool*) e muitos outros estão disponíveis para resolver um problema de otimização;
2. O EXCEL/MATLAB podem ser usados para resolver problemas de programação linear e não linear.

5.1 SOLUÇÃO NUMÉRICA: EXCEL SOLVER

EXCEL SOLVER

O “Solver” é a ferramenta disponível no Excel para resolver uma equação não linear, um sistema de equações lineares/não lineares e problemas de otimização.

Você pode escolher qualquer um dos três métodos de solução na caixa de diálogo Parâmetros do Solver:

- 1. Gradiente Reduzido Generalizado (GRG) Não Linear:** Use para problemas simples não lineares.
- 2. LP Simplex:** Use para problemas lineares.
- 3. Evolucionário:** Use para problemas complexos.

EXCEL SOLVER

Gradiente Reduzido Generalizado (GRG) :

Carpentier e Abadie (1966) generalizaram o método do Gradiente Projetado, para solução de problemas de programação matemática com função objetivo e restrições não-lineares. O Método ficou conhecido como Método do Gradiente Reduzido Generalizado.

Quadro 1 - Notação utilizada.

Nome	Notação	Básico	Não básico
Variáveis	$\mathbf{x}$	$\mathbf{x}_B$	$\mathbf{x}_N$
Gradiente	$\partial f / \partial \mathbf{x}$	$\partial f / \partial \mathbf{x}_B$	$\partial f / \partial \mathbf{x}_N$
Jacobiano	$\partial \mathbf{g} / \partial \mathbf{x}$	$\partial \mathbf{g} / \partial \mathbf{x}_B$	$\partial \mathbf{g} / \partial \mathbf{x}_N$
Direção	$\mathbf{d}$	$\mathbf{d}_B$	$\mathbf{d}_N$
B: conjunto dos índices das variáveis básicas; $ B =m$ N: conjunto dos índices das variáveis não-básicas; $ N =n-m$ $\mathbf{x}^k$, $\mathbf{d}^k$ são os valores de $\mathbf{x}$ e $\mathbf{d}$ na k-ésima iteração			

EXCEL SOLVER

Algoritmo do Método do Gradiente Reduzido Generalizado (GRG) não linear:

Marco Antônio Rahal Sacoman, OTIMIZAÇÃO DE PROJETOS UTILIZANDO GRG, SOLVER E EXCEL. COBENGE 2012.

Passo 1	Encontrar uma primeira solução viável $\mathbf{x}^0$. Considerar $\mathbf{x}^k$ a k-ésima solução encontrada pelo algoritmo.
Passo 2	Calcular o jacobiano $\partial \mathbf{g} / \partial \mathbf{x}^k$ no ponto $\mathbf{x}^k$ e separar as variáveis em $\mathbf{x}_B^k \in \mathbf{R}^m$ e $\mathbf{x}_N^k \in \mathbf{R}^{n-m}$, de forma a satisfazer as hipóteses de não-degenerescência: H1) $x_i \in P, \forall i \in B$; H2) $\partial \mathbf{g} / \partial \mathbf{x}_B^k$ é não-singular.
Passo 3	Calcular a direção de deslocamento das variáveis não-básicas, como segue: a) calcular os multiplicadores de Lagrange $\mathbf{u} = -\frac{\partial f}{\partial \mathbf{x}_B^k} \cdot \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}_B^k} \right]^{-1}$ b) calcular o gradiente reduzido $\nabla_N f = \frac{\partial f}{\partial \mathbf{x}_N^k} + \mathbf{u} \cdot \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}_N^k} \right]$ c) calcular o gradiente reduzido projetado $\forall j \in N, p_j = \begin{cases} \nabla_j f & \text{se } \nabla_j f < 0 \text{ e } x_j^k = a_j \\ 0 & \text{se } \nabla_j f > 0 \text{ e } x_j^k = b_j \\ \nabla_j f & \text{senão} \end{cases}$ Se $p_N = \mathbf{0}$, PARAR; senão, fazer $\mathbf{d}_N = \mathbf{p}_N$.
Passo 4	Considerar a condição de otimalidade $\mathbf{g}' \cdot \mathbf{d} = \mathbf{0}$ e calcular a direção de deslocamento das variáveis básicas. Então, $\frac{\partial \mathbf{g}}{\partial \mathbf{x}_B^k} \cdot \mathbf{d}_B + \frac{\partial \mathbf{g}}{\partial \mathbf{x}_N^k} \cdot \mathbf{d}_N = \mathbf{0}$ e, a partir da relação, calcular $\mathbf{d}_B$ $\mathbf{d}_B = -\left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}_B^k} \right]^{-1} \cdot \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}_N^k} \right] \cdot \mathbf{d}_N$
Passo 5	Melhorar a solução, como segue: a) encontrar um valor positivo θ que maximize $f(\mathbf{x} + \theta \cdot \mathbf{d})$ b) deslocar as variáveis, tanto não-básicas como básicas, segundo as direções calculadas, ou seja, calcular $\tilde{\mathbf{x}}_N = \mathbf{x}_N^k + \theta \cdot \mathbf{d}_N$ e $\tilde{\mathbf{x}}_B = \mathbf{x}_B^k + \theta \cdot \mathbf{d}_B$, encontrando $\tilde{\mathbf{x}} = (\tilde{\mathbf{x}}_B, \tilde{\mathbf{x}}_N)$ que, em geral, não é viável. Então, c) resolver um sistema de m equações não-lineares a m incógnitas, para modificação de suas variáveis básicas $\mathbf{g}(\mathbf{x}_B, \tilde{\mathbf{x}}_N) = \mathbf{0}$, aplicando um método pseudo-Newton: • calcular, iterativamente, a partir de $\tilde{\mathbf{x}}_B$, a solução $\mathbf{x}_B^{t+1} = \mathbf{x}_B^t - \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}_B^k} \right]^{-1} \cdot \mathbf{g}(\mathbf{x}_B^t, \tilde{\mathbf{x}}_N)$ • considerar $\tilde{\mathbf{x}}_B$ a solução encontrada e o ponto obtido $\mathbf{x}^{k+1} = (\tilde{\mathbf{x}}_B, \tilde{\mathbf{x}}_N)$ pode ser tal que: • se $\mathbf{x}^{k+1} \in P$, mas $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$, tentar encontrar um novo ponto, reduzindo θ; • se $\mathbf{x}^{k+1} \in P$, mas $f(\mathbf{x}^{k+1}) > f(\mathbf{x}^k)$, tentar encontrar uma solução melhor, aumentando θ; • se $\mathbf{x}^{k+1} \notin P$, efetuar uma troca de base. Retornar ao Passo 2.

EXCEL SOLVER

SIMPLEX:

O SIMPLEX é usado para problemas de projeto ótimo **com função custo e funções de restrição LINEARES** (*problema de programação linear*). O método se baseia no método de eliminação de Gauss-Jordan, realizando uma espécie de “escalonamento”.

O método Simplex trabalha apenas com restrições do problema cujas desigualdades sejam do tipo " $\leq$ " e seus coeficientes independentes sejam maiores ou iguais a 0. Portanto, é preciso padronizar as restrições para atender aos requisitos antes de iniciar o algoritmo Simplex.

EXCEL SOLVER

SIMPLEX:

Consiste em:

- i) Conhecer uma solução básica viável inicial;
- ii) Testar se a solução é ótima;
- iii) Melhorar a solução a partir de um conjunto de regras e repetir o processo até que uma solução ótima seja obtida..

EXCEL SOLVER

SIMPLEX:

Pode ser resolvido, manualmente, pelo método tabular (Para mais informações ver: <https://www.youtube.com/watch?v=w047PccELc4>) ou (<https://www.youtube.com/watch?v=7qOdbo-xPaA>).

TABLE 8.9 Selection of Pivot Column and Pivot Row for Example 8.7

Basic ↓	x_1	x_2	x_3	x_4	x_5	b	Ratio: $b_i / a_{i1}; a_{i1} > 0$
1. x_3	4	3	1	0	0	12	$\frac{12}{4} = 3$
2. x_4	2	1	0	1	0	4	$\frac{4}{2} = 2 \leftarrow$ smallest
3. x_5	1	2	0	0	1	4	$\frac{4}{1} = 1$
4. Cost function	-2	-1	0	0	0	f	

The selected pivot element is boxed. Selected pivot row and column are highlighted. x_1 should become basic (pivot column). x_4 row has the smallest ratio, and so x_4 should become nonbasic (pivot row).

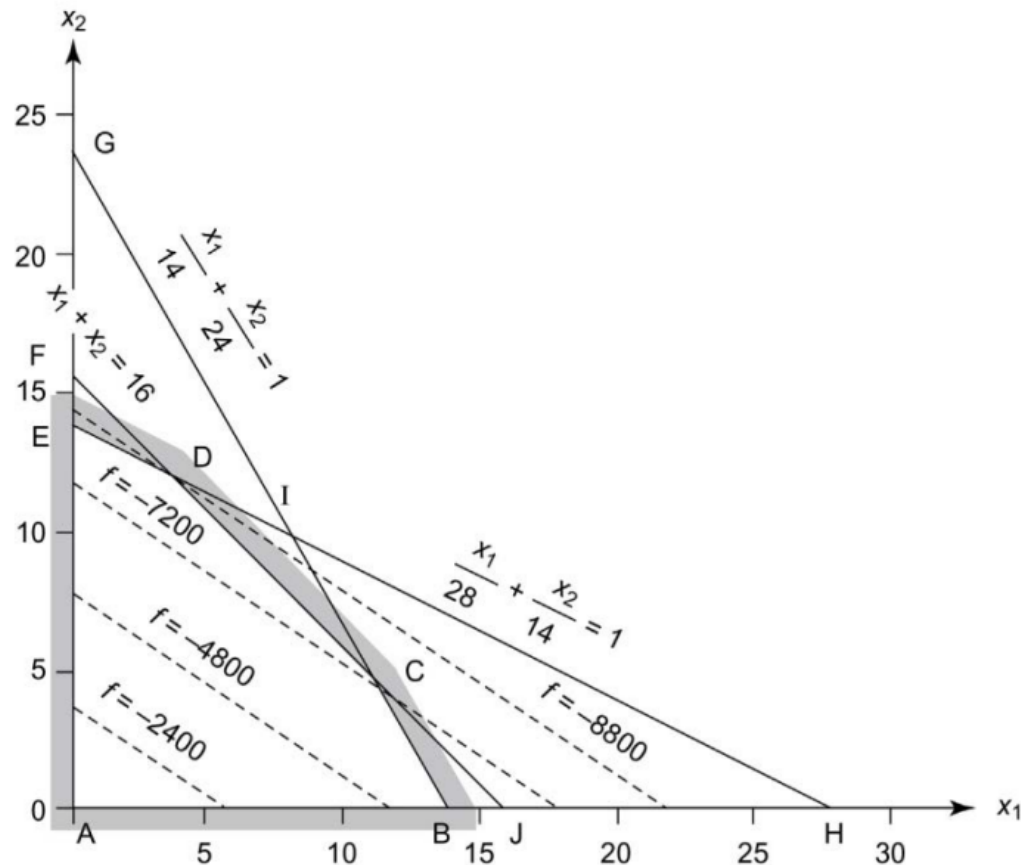
TABLE 8.10 Second Tableau for Example 8.7 Making x_1 a Basic Variable

Basic ↓	x_1	x_2	x_3	x_4	x_5	b
1. x_3	0	1	1	-2	0	4
2. x_1	1	0.5	0	0.5	0	2
3. x_5	0	1.5	0	-0.5	1	2
4. Cost function	0	0	0	1	0	$f + 4$

EXCEL SOLVER

SIMPLEX:

Toda solução ótima está associado a um ponto extremo (Ex.: No caso 2D: um dos pontos **A**, **B**, **C**, **D** ou **E** na Figura a baixo).



EXCEL SOLVER

Evolucionário: O algoritmo Evolutivo é mais robusto do que o GRG Não-linear porque é mais provável encontrar uma solução ótima global. No entanto, esse método de resolução também é **MUITO lento**.

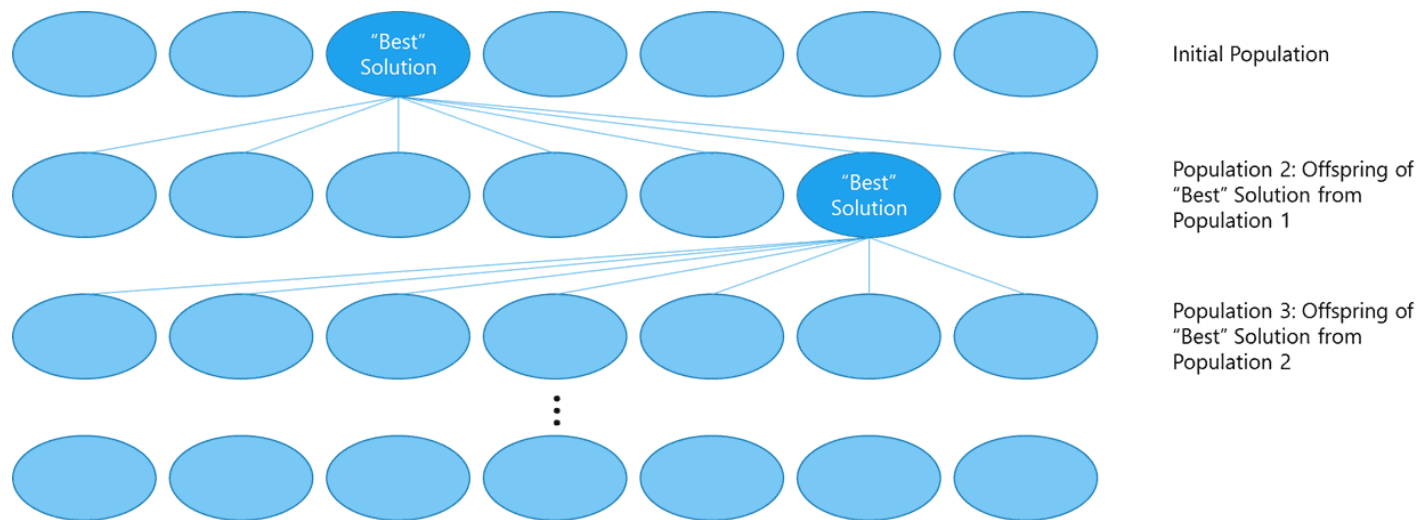
Em resumo, o solucionador começa com uma “população” aleatória de conjuntos de valores de entrada. Os conjuntos de valores de entrada que resultam em uma solução mais próxima do valor alvo são selecionados para criar uma segunda população de “filhos”. A prole é uma “mutação” daquele melhor conjunto de valores de entrada da primeira população. A segunda população é então avaliada e um vencedor é escolhido para criar a terceira população.

Isso continua até que haja muito pouca mudança na função objetivo de uma população para outra.

EXCEL SOLVER

Evolucionário:

O que torna esse processo tão demorado é que cada membro da população deve ser avaliado individualmente. Além disso, as “gerações” subsequentes são preenchidas aleatoriamente em vez de usar derivadas.



EXCEL SOLVER

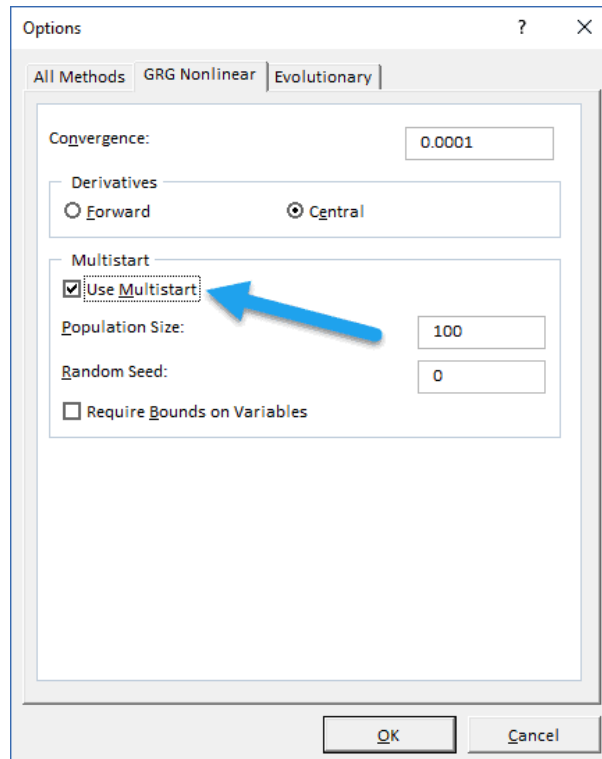
Evolucionário:

O Excel oferece algum controle sobre o algoritmo por meio da janela de opções do Solver. Por exemplo, você pode escolher a taxa de mutação e o tamanho da população para encurtar potencialmente a solução. No entanto, isso tem retornos decrescentes porque reduzir o tamanho da população e/ou aumentar a taxa de mutação pode exigir ainda mais populações para atingir a convergência..

EXCEL SOLVER

GRG MULTISTART:

Uma opção para utilizar a velocidade do algoritmo GRG Nonlinear e a robustez do algoritmo Evolutionary é o ***GRG Nonlinear Multistart***. Você pode habilitar esta opção através da janela Solver Options, na aba GRG Nonlinear.



https://engineerexcel-com.translate.goog/excel-solver-solving-method-choose/?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt-BR&_x_tr_pto=wapp#:~:text=The%20Evolutionary%20method%20is%20based,of%20sets%20of%20input%20values

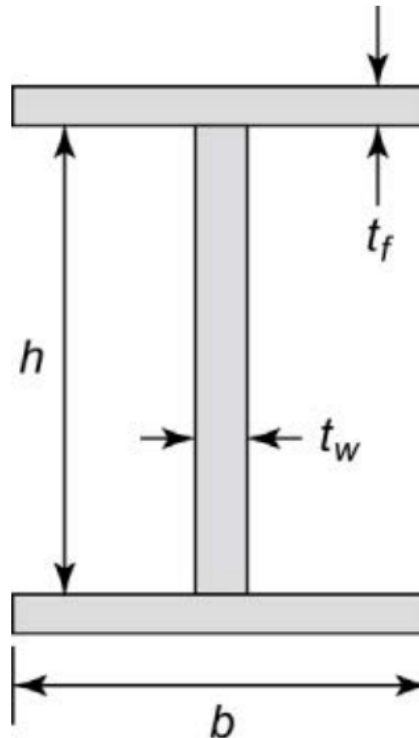
EXCEL SOLVER

GRG MULTISTART:

O algoritmo cria uma população distribuída aleatoriamente de valores iniciais que são avaliados usando o algoritmo não linear GRG tradicional. Partindo várias vezes de diferentes condições iniciais, há uma chance muito maior de que a solução encontrada seja o ótimo global.

EXCEL SOLVER

Formular e Resolver o problema de dimensionamento ótimo de uma viga com perfil I, usando o Excel Solver.



EXCEL SOLVER

Etapa 1: descrição do projeto/problema:

EXEMPLO 1: PROJETO DE UMA VIGA PERFIL I:

Apresentaremos o projeto de uma viga de chapa soldada para uma ponte rodoviária para minimizar seu custo (Arora *et al.*, 1997). Foi determinado que o custo do ciclo de vida da viga está relacionado com a sua massa total. Como a massa é proporcional ao volume do material, o objetivo deste projeto é projetar uma viga de volume mínimo e, ao mesmo tempo, atender aos requisitos das especificações da *American Association of State Highway and Transportation Officials* (AASHTO, 2002). A carga permanente para a viga consiste no peso do pavimento e no peso próprio da viga. A carga variável consiste na carga uniforme equivalente e nas cargas concentradas com base na condição de carregamento do caminhão HS-20(MS18).

EXCEL SOLVER

Etapa 2: Coleta de Dados e Informações:

Os dados de material e carregamento e outros parâmetros para a viga de placa são especificados a seguir:

L = span, 25 m

E = modulus of elasticity, 210 GPa

σ_y = yield stress, 262 MPa

σ_a = allowable bending stress, $0.55\sigma_y = 144.1$ MPa

τ_a = allowable shear stress, $0.33\sigma_y = 86.46$ MPa

σ_t = allowable fatigue stress, 255/2 MPa (factor of safety used = 2)

D_a = allowable deflection, $L/800$ m

P_m = concentrated load for moment, 104 kN

P_s = concentrated load for shear, 155 kN

LLIF = live load impact factor, $1 + \frac{50}{(L + 125)}$

EXCEL SOLVER

Etapa 2: Coleta de Dados e Informações:

Observe que o fator de impacto da carga variável depende do comprimento do vão L . Para $L = 25$ m, esse fator é calculado como 1,33, e assume-se que as cargas P_m e P_s já incorporam esse fator. As variáveis dependentes para o problema, avaliadas usando as dimensões transversais e outros dados, são definidas como:

Cross-sectional area:

$$A = (ht_w + 2bt_f), \text{ m}^2$$

Moment of inertia:

$$I = \frac{1}{12}t_w h^3 + \frac{2}{3}bt_f^3 + \frac{1}{2}bt_f h(h + 2t_f), \text{ m}^4$$

Uniform load for the girder:

$$w = (19 + 77A), \text{ kN} \cdot \text{m}$$

EXCEL SOLVER

Etapa 2: Coleta de Dados e Informações:

Bending moment:

$$M = \frac{L}{8}(2P_m + wL), \text{ kN} \cdot \text{m}$$

Bending stress:

$$\sigma = \frac{M}{1000I}(0.5h + t_f), \text{ MPa}$$

Flange buckling stress limit:

$$\sigma_f = 72,845 \left(\frac{t_f}{b} \right)^2, \text{ MPa}$$

Web crippling stress limit:

$$\sigma_w = 3,648,276 \left(\frac{t_w}{h} \right)^2, \text{ MPa}$$

Shear force:

$$S = 0.5(P_s + wL), \text{ kN}$$

Deflection:

$$D = \frac{L^3}{384 \times 10^6 EI}(8P_m + 5wL), \text{ m}$$

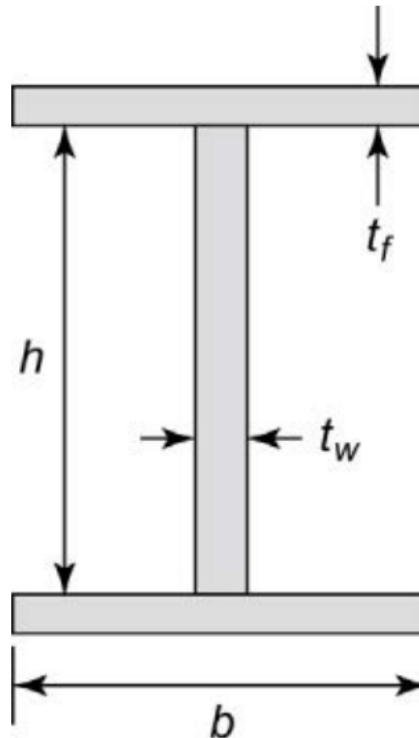
Average shear stress:

$$\tau = \frac{S}{1000ht_w}, \text{ MPa}$$

EXCEL SOLVER

Etapa 3: Definição de Variáveis de Projeto:

As dimensões da seção transversal da viga com perfil I são tratadas como quatro variáveis de projeto para o problema.



EXCEL SOLVER

Etapa 4: Critério de Otimização:

O objetivo é minimizar o volume de material da viga:

$$\text{Vol} = AL = (ht_w + 2bt_f)L, \text{ m}^3$$

EXCEL SOLVER

Etapa 5: Formulação das Restrições:

As seguintes restrições para a viga de placa são definidas:

Bending stress:

$$\sigma \leq \sigma_a$$

Flange buckling:

$$\sigma \leq \sigma_f$$

Web crippling:

$$\sigma \leq \sigma_w$$

Shear stress:

$$\tau \leq \tau_a$$

Deflection:

$$D \leq D_a$$

Fatigue stress:

$$\sigma \leq \sigma_t$$

Size constraints:

$$0.30 \leq h \leq 2.5 \text{ m}; \quad 0.30 \leq b \leq 2.5 \text{ m}$$

EXCEL SOLVER

SOLUÇÃO: Abrir o arquivo no excel

PROJETO ÓTIMO DE VIGASI USANDO O EXCEL SOLVER

1. VARIÁVEIS DE PROJETO					
NOME	LIMITE INFERIOR	SIMBOLO	VALOR	LIMITE SUPERIOR	UNIDADE
altura da alma	0,3	h	0,3	2,5	m
espessura da alma	0,01	tw	0,01	0,1	m
comprimento da mesa	0,3	b	0,3	2,5	m
espessura da mesa	0,01	tf	0,01	0,1	m

2. PARAMETROS			
NOME	SIMBOLO	VALOR	UNIDADE
Comprimento do vão	L	25	m
Módulo de elasticidade	E	210	GPa
Tensão de escoamento	sigma_y	262	MPa
Tensão de flexão admissível	sigma_a	144,1	MPa
Tensão de cisalhamento admissível	tau_a	86,46	MPa
Tensão de flambagem limite da mesa	sigma_f	80,93889	MPa
Tensão de flambagem limite da alma	sigma_w	4053,64	MPa
Tensão de fadiga admissível	sigma_t	127,5	MPa
Carga de momento concentrada	Pm	104	kN
carga de cisalhamento concentrada	Ps	155	kN
Deflexão admissível	Da	0,03125	m

EXCEL SOLVER

SOLUÇÃO 1: A solução ótima para a viga laje é obtida a partir do ponto (0,3, 0,3, 0,01,0,01) da seguinte forma:.

$$h = 2.0755 \text{ m}; b = 0.3960 \text{ m}; t_f = 0.0156 \text{ m}; t_w = 0.0115 \text{ m}; \text{Vol} = 0.90563 \text{ m}^3$$

SOLUÇÃO 2: A solução ótima para a viga laje é obtida a partir do ponto (1,5; 0,3; 0,01,0,01) da seguinte forma:.

$$h = 2.0670 \text{ m}, b = 0.3000 \text{ m}, t_f = 0.0208 \text{ m}, t_w = 0.0115 \text{ m}, \text{Vol} = 0.90507 \text{ m}^3$$

EXCEL SOLVER

Essas duas soluções são um pouco diferentes; no entanto, os valores da função de custo são quase os mesmos. Isso indica que a superfície da função de custo é plana perto do ponto ótimo. Portanto, dependendo do ponto de partida, o processo de solução converge para diferentes pontos ótimos com o mesmo valor de função de custo.

Perguntas do tipo “e se” podem ser investigadas e insights sobre o comportamento do sistema podem ser obtidos. Por exemplo:

1. O que acontece se a restrição de deflexão ou distorção da trama for omitida da formulação?
2. E se o comprimento do vão for alterado?
3. E se algumas propriedades do material mudarem?
4. E se uma variável de projeto receber um valor fixo?
5. E se os limites das variáveis forem alterados?

5.2 SOLUÇÃO NUMÉRICA: Ferramentas de otimização do MATLAB

Ferramentas de otimização do MATLAB

O MATLAB pode ser considerado uma linguagem de programação de alto nível para computação numérica, análise de dados e gráficos para aplicações em diversos campos.

As ferramentas de otimização do MATLAB podem resolver problemas de otimização irrestritos e restritos. Além disso, possui um algoritmo para resolver problemas de otimização não suave. Alguns dos algoritmos de otimização implementados no MATLAB são Resumidos:

Ferramentas de otimização do MATLAB

RESUMO DAS FERRAMENTAS DE OTIMIZAÇÃO DO MATLAB:

TABLE 7.1 Optimization Toolbox Functions

Problem type	Formulation	MATLAB function
<i>Minimização de uma variável em intervalo fixo</i>	Find $\mathbf{x} \in [\mathbf{x}_L \ \mathbf{x}_U]$ to minimize $f(\mathbf{x})$	fminbnd
<i>Minimização irrestrita</i>	Find $\mathbf{x}$ to minimize $f(\mathbf{x})$	fminunc fminsearch
<i>Minimização restrita: Minimiza uma função sujeita a desigualdades e igualdades lineares, desigualdades e igualdades não lineares e limites nas variáveis</i>	Find $\mathbf{x}$ to minimize $f(\mathbf{x})$ subject to $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{Nx} = \mathbf{e}$ $g_i(\mathbf{x}) \leq 0, i = 1 \text{ to } m$ $h_j = 0, j = 1 \text{ to } p$ $x_{iL} \leq x_i \leq x_{iU}$	fmincon
<i>Programação linear: minimizar uma função linear sujeito a desigualdades e igualdades lineares</i>	Find $\mathbf{x}$ to minimize $f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$ subject to $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{Nx} = \mathbf{e}$	linprog
<i>Programação quadrática: Minimize uma função quadrática sujeita a desigualdades e igualdades lineares</i>	Find $\mathbf{x}$ to minimize $f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x}$ subject to $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{Nx} = \mathbf{e}$	quadprog

Ferramentas de otimização do MATLAB

fminbnd: usa o algoritmo é baseado no *Golden section Search* e na interpolação parabólica;

fminunc: usa o algoritmo quasi-newton com um procedimento de busca de linha cúbica. Este método quase-Newton usa a fórmula BFGS para atualizar a aproximação da matriz Hessiana;

fminsearch: usa o método Nelder-Mead simplex;

fmincon: implementa quatro algoritmos diferentes: ponto interior, SQP, conjunto ativo e região de confiança reflexiva;

linprog: Algoritmo Dual-Simplex;

quadprog: usa o algoritmo “*interior-point-convex*” ou o algoritmo “*trust-region-reflective*”.

Ferramentas de otimização do MATLAB

EXEMPLO 1: MINIMIZAÇÃO DE UM PROBLEMA COM UMA ÚNICA VARIÁVEL NÃO RESTRINGIDA.

Find x to

Minimize

$$f(x) = 2 - 4x + e^x, -10 \leq x \leq 10$$

Ferramentas de otimização do MATLAB

Solução: Arquivo 1/2

```
% All comments start with %  
% File name: Example12_1.m  
% Problem: minimize  $f(x) = 2 - 4x + \exp(x)$   
clear all  
% Set lower and upper bound for the design variable  
Lb = -10; Ub = 10;  
% Invoke single variable unconstrained optimizer fminbnd;  
% The argument ObjFunction12_1 refers to the M-file that  
% contains expression for the objective function  
[x, FunVal, ExitFlag, Output] = fminbnd('ObjFunction12_1', Lb, Ub)
```

Solução: Arquivo 2/2

```
% File name: ObjFunction12_1.m  
% Example 12.1 Single variable unconstrained minimization  
function f = ObjFunction12_1(x)  
f = 2 - 4*x + exp(x)
```

Ferramentas de otimização do MATLAB

EXEMPLO 2: Minimização irrestrita multivariável.

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Ferramentas de otimização do MATLAB

Solução: Arquivo 1/2

```
% File name: Example12_2
% Rosenbruck valley function with analytical gradient of
% the objective function
clear all
x0 = [-1.0 1.0]'; % Set starting values
max_ava = 300; % maximum number of function evaluations
% Invoke unconstrained optimization routines
% 1. Nelder-Mead simplex method,
% Set options: medium scale problem, maximum number of function evaluations
options = optimset('LargeScale', 'off', 'MaxFunEvals', max_ava);
[solucao1, FunValue1, ExitFlag1, Output1] = fminsearch ('ObjAndGrad12_2', x0, options)
% 2. BFGS method, , default option
% Set options: medium scale problem, maximum number of function evaluations,
% gradient of objective function
options = optimset('LargeScale', 'off', 'MaxFunEvals', max_ava, 'GradObj', 'on');
[solucao2, FunValue2, ExitFlag2, Output2] = fminunc ('ObjAndGrad12_2', x0, options)
% 3. DFP method, , HessUpdate = dfp
% Set options: medium scale optimization, maximum number of function evaluation,
% gradient of objective function, DFP method
options = optimset('LargeScale', 'off', 'MaxFunEvals', max_ava, 'GradObj', 'on', 'HessUpdate', 'dfp');
[solucao3, FunValue3, ExitFlag3, Output3] = fminunc ('ObjAndGrad12_2', x0, options)
```

Solução: Arquivo 2/2

```
% File name: ObjAndGrad12_2.m
% Rosenbrock valley function
function [f, df] = ObjAndGrad12_2(x)
% Re-name design variable x
x1 = x(1); x2 = x(2); %
% Evaluate objective function
f = 100*(x2 - x1^2)^2 + (1 - x1)^2;
% Evaluate gradient of the objective function
df(1) = -400*(x2-x1^2)*x1 - 2*(1-x1);
df(2) = 200*(x2-x1^2);
```

Ferramentas de otimização do MATLAB

OBSERVAÇÕES EXEMPLO 2: A solução ótima para o problema é conhecida como $\mathbf{x}^* = (1,0; 1,0)$ com $f(\mathbf{x}^*) = 0$ (Schittkowski, 1987).

- *fminsearch* usa o método de busca Simplex de Nelder-Mead, que não requer o gradiente analítico da função objetivo.
- Uma vez que o *fminunc* requer o gradiente e com a opção *LargeScale* definida como 'off', ele usa o método quase-Newton BFGS com uma linha quadrática e cúbica mista como procedimentos de busca.
- A fórmula DFP, que aproxima a matriz Hessiana inversa, pode ser selecionada definindo a opção *HessUpdate* como 'dfp'. O método de descida mais íngreme pode ser selecionado definindo a opção *HessUpdate* como 'steepdesc'.
- *fminsearch* é geralmente menos eficiente que *fminunc*. No entanto, pode ser eficaz para problemas para os quais a avaliação do gradiente é cara ou impossível.

Ferramentas de otimização do MATLAB

Problemas de projeto ótimo restritos:

Se as expressões analíticas de gradiente forem programadas nos arquivos M da função objetivo e das funções de restrição, elas serão declaradas por meio do comando “*options*”. Caso contrário, o *fmincon* usa cálculos numéricos de gradiente com base no método de diferenças finitas.

Ferramentas de otimização do MATLAB

EXEMPLO 3: Problema de minimização restrita usando *fmincon* da caixa de ferramentas de otimização do MATLAB.

Solve the problem to

$$\text{minimize } f(\mathbf{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

subject to the constraints

$$g_1(\mathbf{x}) = 100 - (x_1 - 5)^2 - (x_2 - 5)^2 \leq 0$$

$$g_2(\mathbf{x}) = -82.81 - (x_1 - 6)^2 - (x_2 - 5)^2 \leq 0$$

$$13 \leq x_1 \leq 100, \quad 0 \leq x_2 \leq 100$$

Ferramentas de otimização do MATLAB

Solução: Arquivo 1/3

```
% File name: Example12_3
% Constrained minimization with gradient expressions available
% Calls ObjAndGrad12_3 and ConstAndGrad12_3
clear all
% Set options; medium scale, maximum number of function evaluation,
% gradient of objective function, gradient of constraints, tolerances
options = optimset ('LargeScale', 'off', 'GradObj', 'on', 'GradConstr', 'on', 'TolCon', 1e-8, 'TolX', 1e-8);
% Set bounds for variables
Lb = [13; 0]; Ub = [100; 100];
% Set initial design
x0 = [20.1; 5.84];
% Invoke fmincon; four [ ] indicate no linear constraints in the problem
[x, FunVal, ExitFlag, Output] = fmincon('ObjAndGrad12_3', x0, [], [], [], [], Lb, Ub, 'ConstAndGrad12_3', options)
```

Solução: Arquivo 2/3

```
% File name: ObjAndGrad12_3.m
function [f, gf] = ObjAndGrad12_3(x)
% f returns value of objective function; gf
returns objective function gradient
% Re-name design variables x
x1 = x(1); x2 = x(2);
% Evaluate objective function
f = (x1-10)^3 + (x2-20)^3;
% Compute gradient of objective function
if nargin > 1
gf(1,1) = 3*(x1-10)^2;
gf(2,1) = 3*(x2-20)^2;
end
```

Solução: Arquivo 3/3

```
% File name: ConstAndGrad12_3.m
function [g, h, gg, gh] = ConstAndGrad12_3(x)
% g returns inequality constraints; h returns equality constraints
% gg returns gradients of inequalities; each column contains a gradient
% gh returns gradients of equalities; each column contains a gradient
% Re-name design variables
x1 = x(1); x2 = x(2);
% Inequality constraints
g(1) = 100-(x1-5)^2-(x2-5)^2;
g(2) = -82.81+ (x1-6)^2 + (x2-5)^2;
% Equality constraints (none)
h = [ ];
% Gradients of constraints
if nargin > 2
gg(1,1) = -2*(x1-5);
gg(2,1) = -2*(x2-5);
gg(1,2) = 2*(x1-6);
gg(2,2) = 2*(x2-5);
gh = [ ];
end
```

Ferramentas de otimização do MATLAB

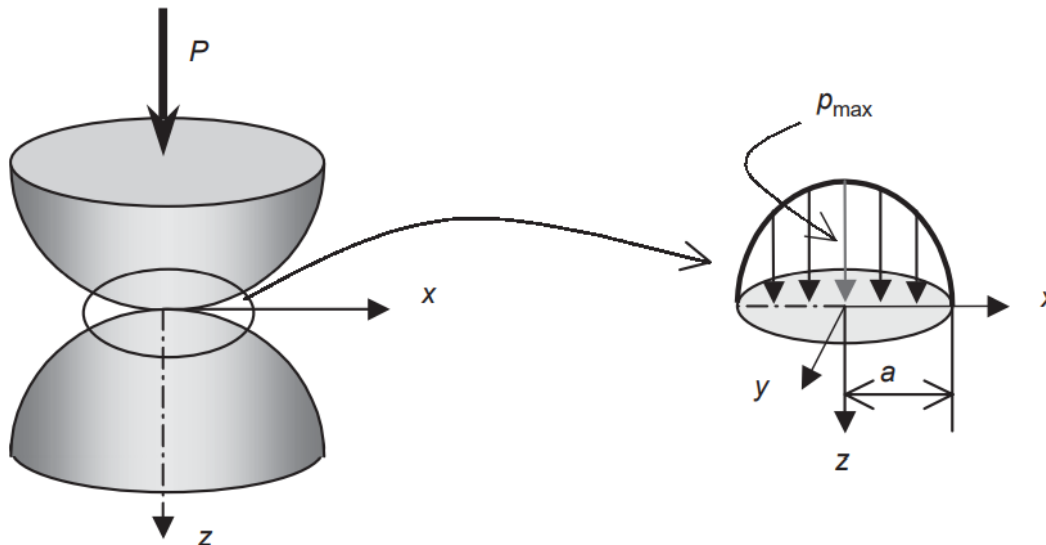
OBSERVAÇÕES EXEMPLO 3: A solução ótima para o problema é conhecida como $\mathbf{x}^* = (14,095; 0,84296)$ com $f(\mathbf{x}^*) = -6961,8$ (Schittkowski, 1987).

- *A saída da função é dada como restrições ativas: $x = (14,095, 0,843)$, $FunVal = -6,9618e+003$, $ExitFlag = 1 > 0$ (ou seja, o mínimo foi encontrado),*

Ferramentas de otimização do MATLAB

EXEMPLO 4: Localização da tensão de cisalhamento máxima para dois corpos esféricos em contato.

Definição do projeto/problema: Existem muitas aplicações práticas em que dois corpos esféricos entram em contato um com o outro, conforme mostrado na Figura. Deseja-se determinar a tensão de cisalhamento máxima e sua localização ao longo do eixo z para um determinado valor da razão de Poisson do material, $\nu = 0,3$.



Ferramentas de otimização do MATLAB

Coleta de dados e informações: A tensão de cisalhamento ao longo do eixo z é calculada usando as tensões principais como (Norton, 2000)

$$\sigma_{xz} = \frac{p_{\max}}{2} \left(\frac{1-2\nu}{2} + \frac{(1+\nu)\alpha}{\sqrt{1+\alpha^2}} - \frac{3}{2} \frac{\alpha^3}{\sqrt[3]{1+\alpha^2}} \right)$$

$\alpha = z/a$ e a é o raio da área de contato conforme mostrado na figura. A pressão máxima ocorre no centro da área de contato a e é dado como:

$$p_{\max} = \frac{3P}{2\pi a^2}$$

É bem conhecido que o pico de tensão de cisalhamento não ocorre na superfície de contato, mas sim a uma pequena distância abaixo da superfície. A localização abaixo da superfície da tensão de cisalhamento máxima acredita-se ser um fator significativo na falha por fadiga superficial chamada pitting.

Ferramentas de otimização do MATLAB

Identificação/Definição das Variáveis de Projeto: α é a única variável de projeto para o problema.

Identificação de um critério a ser otimizado: O objetivo é localizar um ponto ao longo do eixo z onde a tensão de cisalhamento é máxima. Transformando para a forma de minimização padrão e normalizando em relação a p_{max} , o problema se torna: encontre α para minimizar:

$$f(\alpha) = -\frac{\sigma_{xz}}{p_{max}}$$

Identificação de Restrições: Não há restrições para o problema exceto limites na variável α tomada como $0 \leq \alpha \leq 5$.

Solução: A solução exata para o problema é dada como:

$$\left. \frac{\sigma_{xz}}{p_{max}} \right|_{\max} = \frac{1}{2} \left(\frac{1-2\nu}{2} + \frac{2}{9} (1+\nu) \sqrt{2(1+\nu)} \right) \quad \text{at} \quad \alpha = \sqrt{\frac{2+2\nu}{7-2\nu}}$$

Ferramentas de otimização do MATLAB

Solução Numérica:

Este é um problema de otimização de variável única com apenas limites inferior e superior na variável. Portanto, a função *fminbnd* no *Optimization Toolbox* pode ser usada para resolver o problema.

A solução ótima é obtida como $\alpha = 0,6374$, $\text{FunVal} = -0,3329$ [$\alpha^* = 0,6374$, $f(\alpha^*) = -0,3329$] que corresponde à solução exata.

Ferramentas de otimização do MATLAB

Solução: Arquivo 1/2

```
% File name: sphcont_opt.m
% Design variable: ratio of the max shear stress location to
% size of the contact patch
% Find location of the maximum shear stress along the z-axis
clear all
% Set lower and upper bound for the design variable
Lb = 0; Ub = 5;
% Plot normalized shear stress distribution along the z-axis in spherical contact
z = [Lb: 0.1: Ub]';
n = size (z);
for i = 1: n
    outz(i) = -sphcont_objf(z(i));
end
plot(z, outz); grid
xlabel ('normalized depth z/a');
ylabel ('normalized shear stress');
% Invoke the single-variable unconstrained optimizer
[alpha, FunVal, ExitFlag, Output] = fminbnd ('sphcont_objf', Lb, Ub)
```

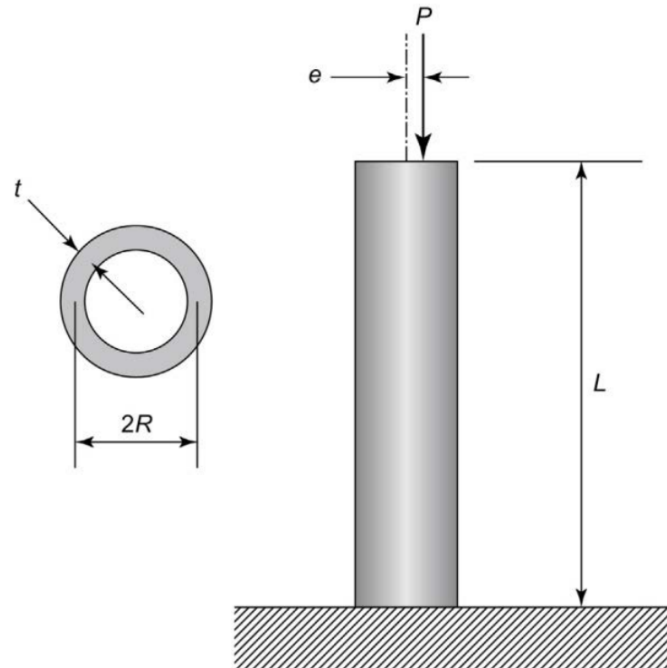
Solução: Arquivo 2/2

```
% File name = sphcont_objf.m
% Location of max shear stress along z-axis for spherical contact problem
function f = sphcont_objf(alpha)
% f = - shear stress/max pressure
nu = 0.3; % Poisson's ratio
f = -0.5*( (1-2*nu)/2 + (1+nu)*alpha/sqrt(1+alpha^2) - 1.5*( alpha/sqrt(1+alpha^2) )^3 );
```

Ferramentas de otimização do MATLAB

EXEMPLO 5: Projeto de Coluna com Massa Mínima.

Definição do projeto/problema: Muitas vezes, Os Pilares são submetidos a cargas excêntricas. O problema é projetar uma coluna tubular de massa mínima que seja submetida a uma carga excêntrica, conforme mostrado na Figura. A seção transversal da coluna é um tubo circular oco com R e t , sendo o raio médio e espessura da parede, respectivamente.



Ferramentas de otimização do MATLAB

Coleta de dados e informações: Os dados do problema são:

Load:	$P = 100 \text{ kN}$
Length:	$L = 5 \text{ m}$
Young's modulus:	$E = 210 \text{ GPa}$
Allowable stress:	$\sigma_a = 250 \text{ MPa}$
Eccentricity (2% of radius):	$e = 0.02R, \text{ m}$
Allowable deflection:	$\Delta = 0.25 \text{ m}$
Mass density:	$\rho = 7850 \text{ kg/m}^3$
Cross-sectional area:	$A = 2\pi R t, \text{ m}^2$
Moment of inertia:	$I = \pi R^3 t, \text{ m}^4$
Distance to the extreme fiber:	$c = R + \frac{1}{2}t, \text{ m}$

Uma análise da estrutura produz as seguintes equações de projeto:

$$\text{Normal stress: } \sigma = \frac{P}{A} \left[1 + \frac{ec}{k^2} \sec \left(\frac{L}{k} \sqrt{\frac{P}{EA}} \right) \right], \quad k^2 = \frac{I}{A}$$

$$\text{Buckling load: } P_{cr} = \frac{\pi^2 EI}{4L^2}$$

$$\text{Deflection: } \delta = e \left[\sec \left(L \sqrt{\frac{P}{EI}} \right) - 1 \right]$$

Ferramentas de otimização do MATLAB

Identificação/Definição das Variáveis de Projeto: R e t são as variáveis de projeto para o problema.

Identificação de um critério a ser otimizado: O objetivo é minimizar a massa da coluna que é dada como:

$$f(\mathbf{x}) = \rho LA = (7850)(5)(2\pi Rt), \text{ kg}$$

Identificação de Restrições: As restrições para o problema estão no desempenho da estrutura, raio máximo em relação à espessura e limites do raio e da espessura:

Stress constraint:	$\sigma \leq \sigma_a$
Buckling load constraint:	$P \leq P_{cr}$
Deflection constraint:	$\delta \leq \Delta$
Radius/thickness constraint:	$\frac{R}{t} \leq 50$
Bounds on the variables:	$0.01 \leq R \leq 1, \quad 0.005 \leq t \leq 0.2$

Ferramentas de otimização do MATLAB

Solução:

Vamos redefinir as variáveis de projeto e outros parâmetros para o MATLAB como:

$$x_1 = R, \quad x_2 = t$$

$$c = x_1 + \frac{1}{2}x_2, \quad e = 0.02x_1$$

$$A = 2\pi x_1 x_2, \quad I = \pi x_1^3 x_2, \quad k^2 = \frac{I}{A} = \frac{x_1^2}{2}$$

Todas as restrições são normalizadas e reescritas usando variáveis de projeto redefinidas. Portanto, o problema de otimização é apresentado na forma padrão da seguinte forma:

Ferramentas de otimização do MATLAB

Solução:

Todas as restrições são normalizadas e reescritas usando variáveis de projeto redefinidas. Portanto, o problema de otimização é apresentado na forma padrão da seguinte forma:

$$\text{minimize } f(\mathbf{x}) = 2\pi(5)(7850)x_1x_2$$

subject to

$$g_1(\mathbf{x}) = \frac{P}{2\pi x_1 x_2 \sigma_a} \left[1 + \frac{2 \times 0.02(x_1 + 0.5x_2)}{x_1} \sec\left(\frac{\sqrt{2}L}{x_1} \sqrt{\frac{P}{E(2\pi x_1 x_2)}}\right) \right] - 1 \leq 0$$

$$g_2(\mathbf{x}) = 1 - \frac{\pi^2 E(2\pi x_1^3 x_2)}{4L^2 P} \leq 0$$

$$g_3(\mathbf{x}) = \frac{0.02x_1}{\Delta} \left[\sec\left(L \sqrt{\frac{P}{E(\pi x_1^3 x_2)}}\right) - 1 \right] - 1 \leq 0$$

$$g_4(\mathbf{x}) = \frac{x_1}{50x_2} - 1 \leq 0$$

$$0.01 \leq x_1 \leq 1, \quad 0.005 \leq x_2 \leq 0.2$$

Ferramentas de otimização do MATLAB

Solução: Arquivo 1/3

```
% File name = column_opt.m
clear all
% Set options
options = optimset('LargeScale', 'off', 'TolCon', 1e-8, 'TolX', 1e-8);
% Set the lower and upper bounds for design variables
Lb = [0.01 0.005]; Ub = [1 0.2];
% Set initial design
x0 = [1 0.2];
% Invoke the constrained optimization routine, fmincon
[x, FunVal, ExitFlag, Output] = fmincon('column_objf', x0, [], [], [], [], Lb, Ub, 'column_conf', options)
```

Solução: Arquivo 2/3

```
% File name = column_objf.m
% Column design
function f = column_objf (x)
% Rename design variables
x1 = x(1); x2 = x(2);
% Set input parameters
L = 5.0; % length of column (m)
rho = 7850; % density (kg/m^3)
f = 2*pi*L*rho*x1*x2; % mass of the
column
```

Solução: Arquivo 3/3

```
% File name = column_conf.m
% Column design
function [g, h] = column_conf (x)
x1 = x(1); x2 = x(2);
% Set input parameters
P = 50000; % loading (N)
E = 210e9; % Young's modulus (Pa)
L = 5.0; % length of the column (m)
Sy = 250e6; % allowable stress (Pa)
Delta = 0.25; % allowable deflection (m)
% Inequality constraints
g(1) = P/(2*pi*x1*x2)*(1 + 2*0.02*(x1+x2/2)/x1*sec(
5*sqrt(2)/x1*sqrt(P/E/(2*pi*x1*x2))))/Sy - 1;
g(2) = 1 - pi^3*E*x1^3*x2/4/L^2/P;
g(3) = 0.02*x1*( sec( L*sqrt( P/(pi*E*x1^3*x2) ) ) - 1 )/Delta - 1;
g(4) = x1/x2/50 - 1;
% Equality constraint (none)
h = [];
```

SOLUÇÃO NUMÉRICA COM O MEF

Muitos problemas de otimização na Engenharia são resolvidos com auxílio de um software acoplado a um programa em MEF.

SOLUÇÃO NUMÉRICA COM O MEF

O software de otimização deve conter pelo menos um algoritmo que satisfaça todos os requisitos ali estabelecidos.

A prova de convergência da maioria dos algoritmos é baseada em certas suposições. Estes precisam ser rigorosamente respeitados durante a implementação do algoritmo.

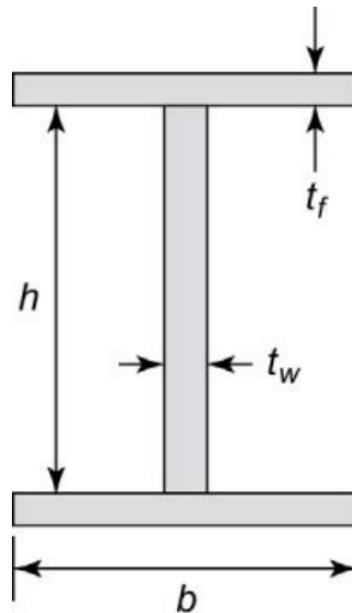
Além disso, a maioria dos algoritmos possuem algumas incertezas numéricas em suas etapas que precisam ser reconhecidas, e procedimentos adequados precisam ser desenvolvidos para sua implementação numérica.

Também é importante que o software seja bem testado em uma variedade de exemplos de dificuldade variada.

SOLUÇÃO NUMÉRICA COM O MEF

EXEMPLO 6: Resolver o problema de dimensionamento ótimo de uma viga com perfil I, modelado no programa em MEF (*ansys apdl*).

- a) Criando o próprio programa;
- b) Usando a toolbox *optimtool* do matlab.



Ferramentas de otimização do MATLAB

a) Criando o próprio programa;

Solução: Arquivo 1/4

```
% File name = column_opt.m
clear all
% Set options
options = optimset ('LargeScale', 'off', 'TolCon', 1e-3, 'TolX', 1e-3);
% Set the lower and upper bounds for design variables
Lb = [0.03 0.01 0.03 0.01]; Ub = [2.5 0.1 2.5 0.1];
% Set initial design
x0 = [2.1 0.02 0.9 0.010];
% Invoke the constrained optimization routine, fmincon

[x, FunVal, ExitFlag, Output] = fmincon('beam_objf', x0, [], [], [], [], Lb, Ub, 'beam_conf', options)
```

Solução: Arquivo 2/4

```
% File name = column_objf.m
% Column design
function f = beam_objf (x)
% Rename design variables
ap = x(1); tw = x(2); bm = x(3); tf = x(4);
% Set input parameters
L = 25; % length of column (m)

[ap tw bm tf]
f = (ap*tw+2*bm*tf)*L % mass of the column
```

Ferramentas de otimização do MATLAB

Solução: Arquivo 3/4

```
% File name = column_conf.m
% Column design
function [g, h] = beam_conf (x)
ap = x(1); tw = x(2); bm = x(3); tf = x(4);
% Set input parameters
L=25;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*****SUBSTITUIÇÃO DAS VARIÁVEIS ALEATORIAS NO
APDL*****%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fid = fopen('Otimization_beam.txt','rt+');
while feof(fid) == 0
    d = [fgetl(fid) ''];
    if (d(1) == '!')
        switch (d)
            case ('!VA1')
                posil=find(d=='=',1);
                y = [d(1:(posil+1)) num2str(bm)];
                fseek(fid,5, 'cof');
                fprintf(fid,'%c', ' ');
                fseek(fid,-30, 'cof');
                fprintf(fid,'%c',y);

            case ('!VA2')
                posil=find(d=='=',1);
                y = [d(1:(posil+1)) num2str(tf)];
                fseek(fid,5, 'cof');
                fprintf(fid,'%c', ' ');
                fseek(fid,-30, 'cof');
                fprintf(fid,'%c',y);

            case ('!VA3')
                posil=find(d=='=',1);
                y = [d(1:(posil+1)) num2str(tw)];
                fseek(fid,5, 'cof');
                fprintf(fid,'%c', ' ');
                fseek(fid,-30, 'cof');
                fprintf(fid,'%c',y);

            case ('!VA4')
                posil=find(d=='=',1);
                y = [d(1:(posil+1)) num2str(ap)];
                fseek(fid,5, 'cof');
                fprintf(fid,'%c', ' ');
                fseek(fid,-30, 'cof');
                fprintf(fid,'%c',y);

        end
    end
end
fclose(fid);
```

```
%% EXECUTA O ANSYS
%system('SET ANS_CONSEC=YES & SET
ANSYS_LOCK=OFF');
system('SET KMP_STACKSIZE=10240k & "C:\Program
Files\ANSYS
Inc\v192\ansys\bin\winx64\ANSYS192.exe" -b -i
Otimization_beam.txt -o ZFEAout.txt');
%10240
```

```
% PLOTAR O GRAFICO DESLOCAMENTO X TEMPO DE
INCENDIO
load beam_file.txt
vetor=(abs(beam_file));
maxx=max(vetor);

j=(L/800)*ones(length(vetor));
i=1:1:length(vetor);
% FORMATAÇÃO DO GRAFICO
box on; hold on;
plot(0.25*i,-100*vetor,'-b','LineWidth',1);
plot(0.25*i,-100*j,'--r','LineWidth',1);
set(gcf,'Color',[1 1 1]); %Setting a white
background for the figure
font_name = 'Times New Roman';
text_size = 16;
xlabel('pontos na viga
(m)', 'FontSize',text_size,'FontName',font_name);

ylabel('deslocamentos(cm)', 'FontSize',text_size,'F
ontName',font_name);
xlim([1 25])
ylim([-12 0])

% Inequality constraints

g(1) = maxx/(L/800)-1;

% Equality constraint (none)
h = [];
end
```

Ferramentas de otimização do MATLAB

Solução: Arquivo 4/4: arquivo em *apdl*

```
/BATCH
!
!=====
!***** START *****
!=====
!
FINISH          ! (Estes dois comandos limpam os dados atuais)
/CLEAR
!
/NOPR           !Suprime a impressao de dados de entrada interpretados
/NOLIST         !Suprime a entrada de dados listados
!
!
!=====
! ***** DADOS DO ARQUIVO *****
!=====
!
! NORSTGM=1 NAO ESCREVER OS DADOS DE GEOMETRIA
! NPROC - NUMERO DE PROCESSADORES
! NOELDB=1, NAO ESCREVER RESULTADOS NA DATABASE
!
/CONFIG,NORSTGM,1,NOELDB,1,NPROC,4
!
! MSGPOP=3 - Mostrar apenas erros ocorridos
! UNDO=2    - NAO escreve comentarios e comandos nao essenciais
! REPLOT=0  - NO AUTOMATIC REPLOT
!
/UIS, MSGPOP, 3, UNDO, 2, REPLOT,0
!
/CWD,'C:\Users\Pichau\FIRE_ANSYS'          ! DIRETORIO
/TITLE,OTIMIZATION
/FILNAME,OTIMIZATION,0                     ! TITULO DO TRABALHO
```

Ferramentas de otimização do MATLAB

b) Usando a toolbox *optimtool* do matlab:

Optimization Tool

File Help

Problem Setup and Results

Solver:

Algorithm:

Problem

Objective function:

Derivatives:

Start point:

Constraints:

Linear inequalities: A: b:

Linear equalities: Aeq: beq:

Bounds: Lower: Upper:

Nonlinear constraint function:

Derivatives:

Run solver and view results

Current iteration:

Options

Stopping criteria

Max iterations: ☒ Use default: 1000 ☐ Specify:

Max function evaluations: ☒ Use default: 3000 ☐ Specify:

X tolerance: ☒ Use default: 1e-10 ☐ Specify:

Function tolerance: ☒ Use default: 1e-6 ☐ Specify:

Constraint tolerance: ☒ Use default: 1e-6 ☐ Specify:

SQP constraint tolerance: ☒ Use default: 1e-6 ☐ Specify:

Unboundedness threshold: ☒ Use default: -1e20 ☐ Specify:

Function value check

☐ Error if user-supplied function returns Inf, NaN or complex

User-supplied derivatives

☐ Validate user-supplied derivatives

Hessian sparsity pattern: ☒ Use default: sparse(ones(numberOfVariables)) ☐ Specify:

Hessian multiply function: ☒ Use default: No multiply function ☐ Specify:

Approximated derivatives

Finite differences $f(x + r*x) - f(x)$

Type:

Relative perturbation vector: ☒ Use default: sqrt(eps)*ones(numberOfVariables,1) ☐ Specify:

Minimum perturbation $|r*x|$: ☒ Use default: 0 ☐ Specify:

Maximum perturbation $|r*x|$: ☒ Use default: Inf ☐ Specify:

Quick Reference

fmincon Solver

Find a minimum of a constrained nonlinear multivariable function using the interior-point algorithm.

Click to expand the section below corresponding to your task.

Problem Setup and Results

- [Solver and Algorithm](#)
- [Problem](#)
- [Constraints](#)
- [Run solver and view results](#)

Options

- [Stopping criteria](#)
- [Function value check](#)
- [User-supplied derivatives](#)
- [Approximated derivatives](#)
- [Hessian](#)
- [Algorithm settings](#)
- [Inner iteration stopping criteria](#)
- [Plot functions](#)
- [Output function](#)
- [Display to command window](#)

Suggested Next Steps

- [Overview of Next Steps](#)
- [When the Solver Fails](#)
- [When the Solver Might Have Succeeded](#)
- [When the Solver Succeeds](#)

More Information

- [User Guide](#)
- [Function equivalent](#)

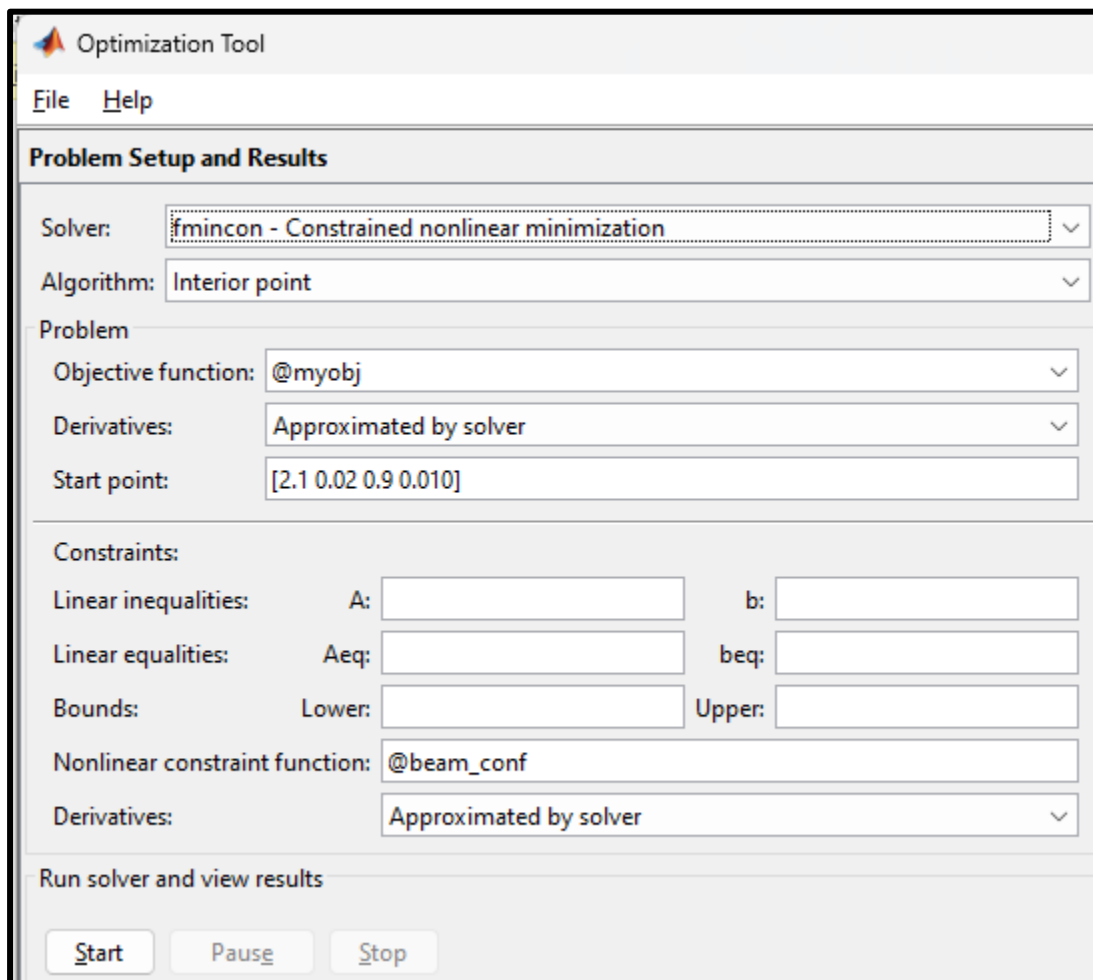
Ferramentas de otimização do MATLAB

b) Usando a toolbox *optimtool* do matlab:

O *optimtool* é uma ferramenta gráfica do MATLAB que pode ser usada para otimização de funções. Aqui estão os passos gerais para usar o *optimtool*:

1. Abra o MATLAB e digite *optimtool* no prompt de comando;
2. Selecione o tipo de otimização desejado na guia “*Solver*”.
3. Selecione a função que deseja otimizar na guia “*objective function*”.
4. Escolha o ponto inicial/de projeto na guia “Start Point”
5. Configure as restrições, se houver, na guia “*Constraints: Nonlinear constraint function*”.
6. Clique no botão “Run” para iniciar a otimização.
7. O resultado será exibido na guia “Results”.

Ferramentas de otimização do MATLAB



The image shows the MATLAB Optimization Tool window. It has a title bar with the MATLAB logo and the text "Optimization Tool". Below the title bar is a menu bar with "File" and "Help". The main area is titled "Problem Setup and Results". It contains several sections: "Solver" with a dropdown menu set to "fmincon - Constrained nonlinear minimization"; "Algorithm" with a dropdown menu set to "Interior point"; "Problem" with fields for "Objective function" (set to "@myobj"), "Derivatives" (set to "Approximated by solver"), and "Start point" (set to "[2.1 0.02 0.9 0.010]"); "Constraints" with fields for "Linear inequalities" (A: and b:), "Linear equalities" (Aeq: and beq:), "Bounds" (Lower: and Upper:), "Nonlinear constraint function" (set to "@beam_conf"), and "Derivatives" (set to "Approximated by solver"); and a "Run solver and view results" section at the bottom with "Start", "Pause", and "Stop" buttons.

Optimization Tool

File Help

Problem Setup and Results

Solver: fmincon - Constrained nonlinear minimization

Algorithm: Interior point

Problem

Objective function: @myobj

Derivatives: Approximated by solver

Start point: [2.1 0.02 0.9 0.010]

Constraints:

Linear inequalities: A: b:

Linear equalities: Aeq: beq:

Bounds: Lower: Upper:

Nonlinear constraint function: @beam_conf

Derivatives: Approximated by solver

Run solver and view results

Start Pause Stop

Ferramentas de otimização do MATLAB

b) Usando a toolbox *optimtool* do matlab:

Solução: Arquivo 1/3

```
function f = myobj(x)
% Rename design variables
ap = x(1); tw = x(2); bm = x(3); tf = x(4);
% Set input parameters
L = 25; % length of column (m)

[ap tw bm tf]
f = (ap*tw+2*bm*tf)*L % mass of the column

end
```

Ferramentas de otimização do MATLAB

Solução: Arquivo 2/3

```
% File name = column_conf.m
% Column design
function [g, h] = beam_conf (x)
ap = x(1); tw = x(2); bm = x(3); tf = x(4);
% Set input parameters
L=25;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*****SUBSTITUIÇÃO DAS VARIÁVEIS ALEATORIAS NO
APDL*****%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fid = fopen('Otimization_beam.txt','rt+');
while feof(fid) == 0
    d = [fgetl(fid) ''];
    if (d(1) == '!')
        switch (d)
            case ('!VA1')
                posil=find(d=='=',1);
                y = [d(1:(posil+1)) num2str(bm)];
                fseek(fid,5, 'cof');
                fprintf(fid,'%c', ' ');
                fseek(fid,-30, 'cof');
                fprintf(fid,'%c',y);

            case ('!VA2')
                posil=find(d=='=',1);
                y = [d(1:(posil+1)) num2str(tf)];
                fseek(fid,5, 'cof');
                fprintf(fid,'%c', ' ');
                fseek(fid,-30, 'cof');
                fprintf(fid,'%c',y);

            case ('!VA3')
                posil=find(d=='=',1);
                y = [d(1:(posil+1)) num2str(tw)];
                fseek(fid,5, 'cof');
                fprintf(fid,'%c', ' ');
                fseek(fid,-30, 'cof');
                fprintf(fid,'%c',y);

            case ('!VA4')
                posil=find(d=='=',1);
                y = [d(1:(posil+1)) num2str(ap)];
                fseek(fid,5, 'cof');
                fprintf(fid,'%c', ' ');
                fseek(fid,-30, 'cof');
                fprintf(fid,'%c',y);

        end
    end
end
fclose(fid);
```

```
%% EXECUTA O ANSYS
%system('SET ANS_CONSEC=YES & SET
ANSYS_LOCK=OFF');
system('SET KMP_STACKSIZE=10240k & "C:\Program
Files\ANSYS
Inc\v192\ansys\bin\winx64\ANSYS192.exe" -b -i
Otimization_beam.txt -o ZFEAout.txt');
%10240
```

```
% PLOTAR O GRAFICO DESLOCAMENTO X TEMPO DE
INCENDIO
load beam_file.txt
vetor=(abs(beam_file));
maxx=max(vetor);
```

```
j=(L/800)*ones(length(vetor));
i=1:1:length(vetor);
% FORMATAÇÃO DO GRAFICO
box on; hold on;
plot(0.25*i,-100*vetor,'-b','LineWidth',1);
plot(0.25*i,-100*j,'--r','LineWidth',1);
set(gcf,'Color',[1 1 1]); %Setting a white
background for the figure
font_name = 'Times New Roman';
text_size = 16;
xlabel('pontos na viga
(m)', 'FontSize',text_size,'FontName',font_name);

ylabel('deslocamentos(cm)', 'FontSize',text_size,'F
ontName',font_name);
xlim([1 25])
ylim([-12 0])
```

```
% Inequality constraints
```

```
g(1) = maxx/(L/800)-1;
```

```
% Equality constraint (none)
h = [];
end
```

Ferramentas de otimização do MATLAB

Solução: Arquivo 4/4: arquivo em *apdl*

```
/BATCH
!
!=====
!***** START *****
!=====
!
FINISH          ! (Estes dois comandos limpam os dados atuais)
/CLEAR
!
/NOPR           !Suprime a impressao de dados de entrada interpretados
/NOLIST         !Suprime a entrada de dados listados
!
!
!=====
! ***** DADOS DO ARQUIVO *****
!=====
!
! NORSTGM=1 NAO ESCREVER OS DADOS DE GEOMETRIA
! NPROC - NUMERO DE PROCESSADORES
! NOELDB=1, NAO ESCREVER RESULTADOS NA DATABASE
!
/CONFIG,NORSTGM,1,NOELDB,1,NPROC,4
!
! MSGPOP=3 - Mostrar apenas erros ocorridos
! UNDO=2    - NAO escreve comentarios e comandos nao essenciais
! REPLOT=0  - NO AUTOMATIC REPLOT
!
/UIS, MSGPOP, 3, UNDO, 2, REPLOT,0
!
/CWD,'C:\Users\Pichau\FIRE_ANSYS'          ! DIRETORIO
/TITLE,OTIMIZATION
/FILNAME,OTIMIZATION,0                     ! TITULO DO TRABALHO
```

Ferramentas de otimização do MATLAB

b) Usando a toolbox *optimtool* do matlab:

Observe que o *optimtool* é apenas uma interface gráfica para otimização de funções no MATLAB, e pode não ser a melhor escolha para todos os problemas de otimização. Para problemas mais complexos, é possível escrever código manualmente usando as funções de otimização incluídas no MATLAB.

6) Conceitos e métodos de projeto ótimo de variável discreta (Não é escopo do curso)

Projeto Ótimo de Variável Discreta

➤ Problemas de programação inteira (IP) :

Em muitas aplicações práticas, variáveis de projeto discretas e inteiras ocorrem naturalmente na formulação do problema. Variável discreta: Uma variável é chamada discreta se seu valor deve ser atribuído a partir de um determinado conjunto de valores. Variável inteira: Uma variável que pode ter apenas valores inteiros é chamada de variável inteira.

Por exemplo: o diâmetro da barra de aço deve ser selecionado entre os disponíveis;

Projeto Ótimo de Variável Discreta

➤ Problemas de programação inteira (IP) :

Vários algoritmos estão disponíveis para resolver tais problemas como o BBM (MÉTODOS BRANCH-AND-BOUND), MÉTODO DE ARREDONDAMENTO DINÂMICO, NEIGHBORHOOD SEARCH; MÉTODOS PARA VARIÁVEIS DISCRETAS LIGADAS; MÉTODO NUMÉRICO ADAPTÁVEL PARA DISCRETOOTIMIZAÇÃO DE VARIÁVEL. Problemas discretos não lineares também podem ser resolvido por procedimentos de linearização sequencial se as funções do problema forem contínuas e diferenciável.

Projeto Ótimo de Variável Discreta

➤ MÉTODO DE ARREDONDAMENTO DINÂMICO:

Uma abordagem simples para problemas do tipo é primeiro obter uma solução ótima usando uma abordagem contínua. Em seguida, usando heurísticas, as variáveis são arredondadas para o valor mais próximo. O arredondamento é uma ideia simples que tem sido usada com frequência, mas pode resultar em projetos inviáveis para problemas com um grande número de variáveis.

A principal preocupação da abordagem de arredondamento é a seleção das variáveis a serem aumentadas e as variáveis a serem diminuídas. A estratégia pode não convergir, especialmente no caso de alta não linearidade e valores discretos permitidos amplamente separados. Nesse caso, o ponto de mínimo discreto não precisa estar na vizinhança da solução contínua.

Projeto Ótimo de Variável Discreta

➤ SELEÇÃO DO MÉTODO:

Vê-se que BBM, recozimento simulado, algoritmos genéticos e outros métodos inspirados na natureza são os métodos mais gerais. Eles podem ser usados para resolver todos os tipos de problemas. No entanto, estes também são os mais caros em termos de esforço computacional. Se as funções do problema são diferenciáveis e variáveis discretas podem receber valores não discretos durante o processo de solução iterativa, então existem inúmeras estratégias para sua solução que são mais eficientes do que os três métodos que acabamos de discutir. A maioria deles envolve uma combinação de dois ou mais algoritmos. Para problemas discretos e não diferenciáveis, os métodos de recozimento simulado e inspirados na natureza são apropriados para otimização global.

7) Conceitos e métodos de otimização global

Métodos de Otimização Global

➤ Resumo:

Até agora, abordamos principalmente o problema de encontrar um mínimo local para a função de custo no conjunto viável. Neste capítulo, nos concentramos na apresentação e discussão de conceitos e métodos para as soluções ótimas globais porque, em algumas aplicações práticas, é importante encontrar tais soluções em oposição às locais.

Métodos de Otimização Global

➤ Resumo:

Não há condições matemáticas que caracterizem um ponto de mínimo global, exceto quando o problema pode ser mostrado como convexo.

Uma questão importante é como sabemos que um processo de busca numérica terminou em um ponto de mínimo global. Em geral, a resposta é que não sabemos. Devido a isso, é difícil definir um critério de parada preciso para um algoritmo computacional de otimização global.

Métodos de Otimização Global

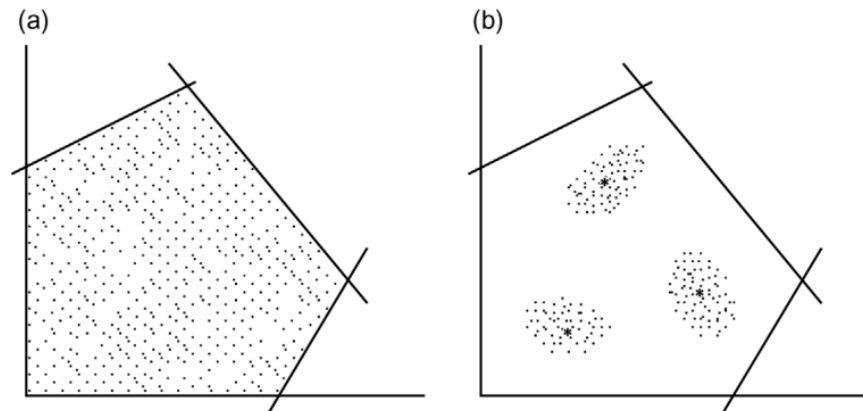
➤ Resumo:

A qualidade da solução geralmente depende de quanto tempo o algoritmo pode ser executado. É importante notar que o esforço computacional para resolver um problema de otimização global é substancial e aumenta enormemente com o aumento do número de variáveis de projeto. Assim, resolver problemas globais de otimização continua sendo um desafio tanto do ponto de vista matemático quanto computacional.

Métodos de Otimização Global

➤ Resumo:

Devido à falta de condições globais de otimalidade para problemas gerais, uma solução global para o problema pode ser obtida apenas por uma busca exaustiva do espaço de projeto. O procedimento para tal busca é especificar alguns pontos amostrais no conjunto S_b e avaliar a função de custo neles. O ponto onde a função tem o menor valor é considerado o ponto de mínimo global. Vemos que a localização e o valor do global mínimo depende do tamanho da amostra. Uma solução exata para o problema requer um número infinito de cálculos..



Métodos de Otimização Global

➤ Visão geral dos métodos:

Os métodos de otimização global podem ser divididos em duas categorias principais: determinísticos e estocásticos.

DETERMINÍSTICOS: Métodos determinísticos encontram o mínimo global por uma busca exaustiva sobre o conjunto S_b . O sucesso deste método pode ser garantido apenas para as funções que satisfaçam certas condições. Quatro métodos determinísticos são os mais comuns, a saber: cobertura, zoom, descida generalizada e tunelamento.

Métodos de Otimização Global

➤ Visão geral dos métodos:

ESTOCÁSTICOS: Todos os métodos estocásticos envolvem elementos aleatórios para determinar o ponto de mínimo global, cada um tentando reduzir a carga computacional da busca aleatória. No início, uma amostra aleatória de pontos no conjunto S_b é escolhida. Em seguida, cada método manipula os pontos de amostra de uma maneira diferente. Os métodos mais comuns são: multistart, clustering, controle de busca aleatória, aceitação-rejeição (A-R), integração estocástica, estocástica zoom e eliminação de domínio. Além destes, existem métodos inspirados na natureza que também usam elementos estocásticos em seus algoritmos.

Métodos de Otimização Global

➤ Visão geral dos métodos:

A Tabela resume as características de vários algoritmos de otimização global:

TABLE 16.1 Characteristics of Global Optimization Methods

Method	Can solve discrete problems?	General constraints?	Tries to find all x^* ?	Phases	Needs gradients?
Covering (D)	No	No	Yes	G	1
Zooming (D)	Yes ¹	Yes	No	L	1
Generalized descent (D)	No	No	No	G	Yes
Tunneling (D)	No	Yes	No	L + G	1
Multistart (S)	Yes ¹	Yes	Yes	L + G	1
Clustering (S)	Yes ¹	Yes	Yes	L + G	1
Controlled random search (S)	Yes	No	No	L + G	No
Acceptance–rejection (S)	Yes ¹	Yes	No	G	No
Stochastic integration (S)	No	No	No	G	No
Genetic (S)	Yes	No	No	G	No
Stochastic zooming (S)	Yes ¹	Yes	No	L + G	1
Domain elimination (S)	Yes ¹	Yes	Yes	L + G	1

Note: *D*, deterministic methods; *S*, stochastic methods; *G*, global phase; *L*, local phase.

¹*Depends on the local minimization procedure used.*

8) Métodos de pesquisa inspirados na natureza

Métodos Inspirados na Natureza

➤ Resumo:

Os métodos inspirados na natureza usam ideias estocásticas e números aleatórios em seus cálculos para procurar o ponto ótimo. As decisões tomadas na maioria das etapas dos algoritmos são baseadas na geração de números aleatórios. Portanto, executados em momentos diferentes, os algoritmos podem levar a uma sequência diferente de projetos e a uma solução diferente mesmo com as mesmas condições iniciais. Eles tendem a convergir para um ponto de mínimo global para a função, mas não há garantia de convergência ou otimalidade global.

Métodos Inspirados na Natureza

➤ Resumo:

Abordagens inspiradas na natureza têm sido chamadas de programação estocástica, algoritmos evolutivos, programação genética, inteligência de enxame e computação evolutiva. Eles também são chamados de métodos metaheurísticos inspirados na natureza, pois não fazem suposições sobre o problema de otimização e podem pesquisar espaços muito grandes para soluções candidatas.

Métodos Inspirados na Natureza

➤ Resumo:

Algoritmos inspirados na natureza podem superar alguns dos desafios complexos. Eles são relativamente fáceis de usar e programar, pois não requerem o uso de gradientes de custo ou funções de restrição.

Métodos Inspirados na Natureza

➤ Resumo:

As desvantagens desses algoritmos são as seguintes:

1. Eles exigem uma grande quantidade de avaliações de função, mesmo para problemas de tamanho razoável. Para problemas em que a própria avaliação de funções requer cálculos massivos, a quantidade de tempo de computação necessária para resolver o problema pode ser proibitiva.
2. Não há garantia absoluta de que uma solução global será obtida..

Métodos Inspirados na Natureza

➤ Resumo:

Como exemplo, pode-se citar quatro métodos que são relativamente populares:

1. ALGORITMOS GENÉTICOS (GA) PARA PROJETO ÓTIMO;
2. ALGORITMO DE EVOLUÇÃO DIFERENCIAL;
3. OTIMIZAÇÃO DA COLÔNIA DE FORMIGAS;
4. OTIMIZAÇÃO DE ENXAME DE PARTICULAS

9) Conceitos e métodos de projeto otimizado multiobjetivo

Problemas de Otimização Multiobjetivo

➤ Resumo:

Projetos onde se consideraram duas ou mais funções objetivas, simultaneamente, são chamados de problemas multiobjetivos, multicritérios ou de otimização vetorial; nos referimos a eles como **PROBLEMAS DE OTIMIZAÇÃO MULTIOBJETIVO**.

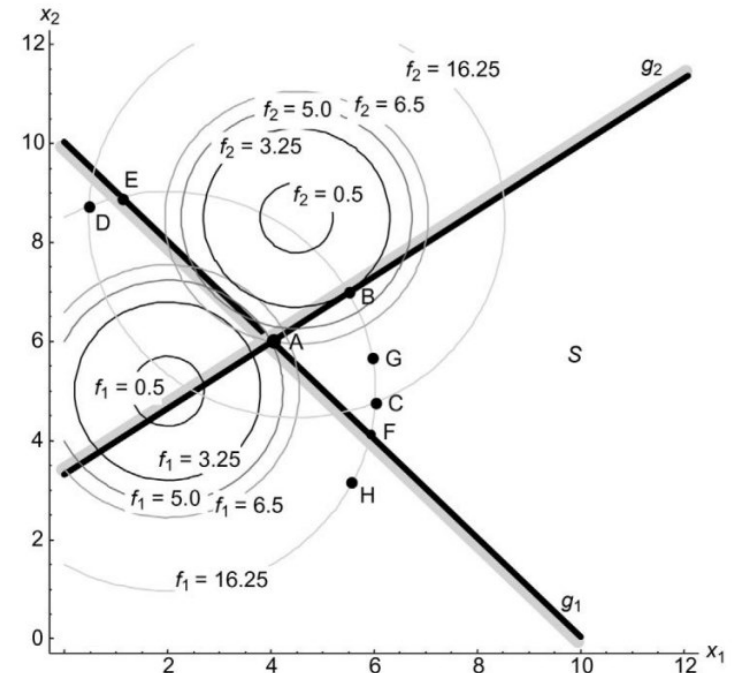
Minimize:

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))$$

subject to:

$$h_i(\mathbf{x}) = 0; \quad i = 1 \text{ to } p$$

$$g_j(\mathbf{x}) \leq 0; \quad j = 1 \text{ to } m$$



Problemas de Otimização Multiobjetivo

➤ A Toolbox de Otimização do MATLAB suporta vários métodos de otimização multiobjetivo, incluindo:

1. Algoritmos evolutivos: Algoritmos Genéticos, Algoritmos de Otimização por Enxame Particulado e Algoritmos de Colônia de Formigas.
2. Métodos híbridos: Métodos que combinam técnicas de diferentes métodos de otimização, como algoritmos genéticos e algoritmos de gradiente.
3. Métodos de divisão e conquista: Métodos que dividem o problema em subproblemas menores e os resolvem individualmente antes de combinar as soluções.
4. Métodos de algoritmos genéticos nichados: Algoritmos que exploram a fronteira de Pareto de maneira eficiente.
5. Métodos de escalonamento de pesos: Métodos que permitem aos usuários definir os pesos dos objetivos para guiar a busca de soluções.

Além disso, a Toolbox de Otimização do MATLAB fornece ferramentas para visualização da fronteira de Pareto e avaliação da qualidade das soluções encontradas.

10) Tópicos Adicionais sobre Projeto Ideal

Tópicos Adicionais

➤ Resumo:

Além dos métodos apresentados nesse curso, existem diversas pesquisas que buscam melhorar a resposta do problema (diminuindo os erros) e reduzir o custo computacional.

Alguns exemplos são: Geração de metamodelos para problemas práticos de otimização de projeto, projeto de experimentos para geração de superfície de resposta e RBDO (projeto sob incerteza - confiabilidade).

Esses tópicos, geralmente, são abordados em cursos de Mestrado e Doutorado.



UNIVERSIDADE FEDERAL
DE ALAGOAS



OBRIGADO

Prof. Dr. Alverlando Silva Ricardo

*Universidade Federal de Alagoas - Campus do Sertão
Engenharia Civil/Produção*

E-mail: alverlando.ricardo@delmiro.ufal.br

Site: alverlandoricardo.wixsite.com/professor