

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

— Otimização de Processos —

Prof. Argimiro R. Secchi

– 2001 - 2005 –

Conteúdo

1. FORMULAÇÃO DE PROBLEMAS	5
1.1 APLICAÇÕES	5
1.2 NOMENCLATURA	7
1.3 PROCEDIMENTO GERAL.....	9
1.4 DIFICULDADES ENCONTRADAS	10
2. CONCEITOS BÁSICOS.....	10
2.1 MÍNIMOS E MÁXIMOS.....	10
2.2 CONDIÇÕES PARA A OTIMALIDADE.....	11
2.2.1 <i>Otimização sem restrição</i>	11
2.2.2 <i>Otimização com restrições</i>	15
2.3 CONVEXIDADE.....	23
2.4.1 <i>Generalização de funções convexas</i>	25
2.4 FORMAS FUNCIONAIS	27
2.5 ÁLGEBRA VETORIAL	30
EXERCÍCIOS DE FIXAÇÃO	31
3. OTIMIZAÇÃO SEM RESTRIÇÃO	32
3.1 MÉTODOS DE BUSCA MONOVARIÁVEL E MULTIVARIÁVEL.....	32
3.1.1 <i>Método da seção áurea</i>	33
3.1.2 <i>Método da aproximação polinomial</i>	34
3.1.3 <i>Método de busca seccionada</i>	36
3.1.4 <i>Método de Hooke & Jeeves</i>	37
3.1.5 <i>Método de Rosenbrock</i>	38
3.1.6 <i>Método de Powell</i>	39
3.1.7 <i>Método de busca de limites</i>	42
3.1.8 <i>Método dos poliedros flexíveis</i>	42
3.1.9 <i>Métodos de busca aleatória</i>	44
3.2 MÉTODOS ANALÍTICOS (MÉTRICA VARIÁVEL)	51

3.2.1 Métodos gradientes.....	52
3.2.2 Método de Newton.....	54
3.2.3 Método do gradiente conjugado.....	55
3.2.4 Métodos da métrica variável.....	56
4. TEORIA DA DUALIDADE.....	63
4.1 PROBLEMA PRIMAL.....	63
4.2 PROBLEMA DUAL.....	64
4.3 INTERPRETAÇÃO GEOMÉTRICA DO DUAL.....	65
4.4 DUALIDADE FRACA E FORTE.....	68
5. PROGRAMAÇÃO LINEAR (LP).....	71
5.1 FUNDAMENTOS.....	72
5.2 MÉTODO SIMPLEX.....	74
5.2.1 Primeira solução viável.....	77
5.2.2 Forma padrão.....	78
5.3 MÉTODO DO PONTO INTERIOR (KARMARKAR).....	79
6. PROGRAMAÇÃO NÃO LINEAR (NLP).....	83
6.1 PROGRAMAÇÃO QUADRÁTICA (QP).....	83
6.2 RELAXAÇÃO LAGRANGEANA.....	87
6.3 FUNÇÕES PENALIDADE.....	90
6.4 PROGRAMAÇÃO LINEAR SEQUENCIAL (SLP).....	93
6.5 GRADIENTES REDUZIDOS GENERALIZADOS (GRG).....	95
6.6 PROGRAMAÇÃO QUADRÁTICA SEQUENCIAL (SQP).....	100
<i>Exercícios de fixação</i>	104
6.7 OTIMIZAÇÃO MULTI-OBJETIVO.....	105
7. PROGRAMAÇÃO INTEIRA MISTA.....	113
7.1 PROGRAMAÇÃO DINÂMICA.....	116
7.2 PROGRAMAÇÃO LINEAR INTEIRA MISTA (MILP).....	118
7.2.1 Princípio dos métodos dos planos de corte.....	120

7.2.2 Método de <i>branch and bound</i>	121
7.3 PROGRAMAÇÃO NÃO LINEAR INTEIRA MISTA (MINLP)	127
7.3.1 <i>Decomposição Generalizada de Benders (GBD)</i>	127
7.3.2 <i>Aproximações Externas (OA)</i>	129
7.3.3 <i>Aproximações Externas com Relaxamento de Igualdade (OA/ER)</i>	130
7.3.4 <i>Aproximações Externas com Relaxamento de Igualdade e Penalidade Aumentada (OA/ER/AP)</i> ..	131
7.3.5 <i>Aproximação Externa Generalizada (GOA)</i>	132
7.3.6 <i>Decomposição Cruzada Generalizada (GCD)</i>	133
7.3.7 <i>Branch and Bound Espacial (SBB)</i>	134
8. OTIMIZAÇÃO DE PROCESSOS DINÂMICOS.....	142
8.1 FORMULAÇÃO DA FUNÇÃO OBJETIVO E RESTRIÇÕES	143
8.2 PRINCÍPIO DO MÍNIMO DE PONTRYAGIN.....	144
8.3 CONTROLE ÓTIMO.....	151
EXERCÍCIOS	155
BIBLIOGRAFIA BÁSICA	158

1. Formulação de Problemas

Os componentes chaves da formulação de um problema de otimização são:

- 1) A função objetivo
- 2) O modelo do processo
- 3) As restrições

A função objetivo representa lucro, custo, energia, produção, distância, etc., em termos das variáveis de decisão do processo ou sistema em análise. O modelo do processo e as restrições descrevem as inter-relações entre estas variáveis.

1.1 Aplicações

As técnicas de otimização, que buscam a melhor solução para um problema (máximos ou mínimos de grandezas mensuráveis em seus domínios de definição), fazem-se necessárias em muitas áreas da engenharia, tais como:

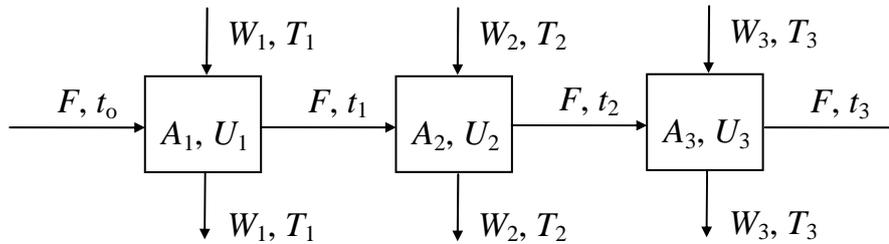
- pesquisa operacional: otimização de sistemas técnico-econômicos, controle de estoques, planejamento de produção, etc.;
- engenharia de processos: dimensionamento e otimização de processos, integração mássica e energética de processos, estimação de parâmetros, reconciliação de dados, análise de flexibilidade, etc.;
- controle de processos: identificação de sistemas, controle ótimo, controle adaptativo, controle preditivo, estimadores de estados, etc.;
- análise numérica: aproximações, regressão, solução de sistemas lineares e não-lineares, etc.

A otimização pode ser efetuada em diferentes níveis dentro de uma empresa, desde uma combinação complexa de plantas e unidades de suprimentos, passando por plantas individuais, combinações de unidades, equipamentos, peças de equipamentos, até entidades menores. Portanto, o escopo de um problema de otimização pode ser toda a empresa, uma planta, um processo, uma equipamento, uma peça de um equipamento, ou qualquer sistema intermediário entre estes. Em uma indústria típica existem três níveis que podem ser otimizados: nível gerencial, nível de projeto de processo e especificação de equipamento e nível de operação de plantas.

Existe uma série de atributos de processos que afetam os custos ou lucros, tornando-os atrativos para aplicação de otimização, tais como:

- vendas limitadas pela produção → aumento de capacidade produtiva
- vendas limitadas pelo mercado → aumento de eficiência ou produtividade
- grandes unidades de processamento → redução dos custos de produção
- elevados consumos de matéria-prima ou energia → redução do consumo
- qualidade do produto superior às especificações → operar perto das restrições
- perda de componentes valiosos para os efluentes → ajustes e re-aproveitamentos
- alto custo de mão-de-obra → automação e replanejamento das unidades.

Exemplo 1.1: No projeto de um sistema de resfriamento representado abaixo:



deseja-se dimensionar os trocadores de calor de modo a resfriar a corrente a temperatura t_0 para t_3 com o menor custo possível. Uma análise econômica levou a seguinte expressão do custo do sistema:

$$\text{custo por unidade: } c_i = a_i \sqrt{A_i} + b_i W_i$$

$$\text{custo total: } c_T = \sum_{i=1}^3 c_i$$

onde o coeficiente a_i é função do tipo de trocador de calor e do fluido refrigerante e b_i é o custo unitário do refrigerante. Uma análise técnica (balanço de energia) gerou as seguintes relações restritivas:

$$A_i = \frac{F C_p}{U_i} \ln \left(\frac{t_{i-1} - T_i}{t_i - T_i} \right) \quad \text{e} \quad W_i = \frac{F C_p}{\lambda_i} (t_{i-1} - t_i)$$

onde C_p é o calor específico da corrente, U_i é o coeficiente global de troca térmica e λ_i é o calor de vaporização do refrigerante. Portanto, dados os tipos de trocadores de calor e fluidos refrigerantes, e conhecidas as temperaturas T_i e as propriedades C_p , λ_i e U_i , o problema de levar a corrente (F, t_0) para a condição (F, t_3) da maneira mais econômica possível, consiste em minimizar c_T no projeto dos trocadores de calor.

1.2 Nomenclatura

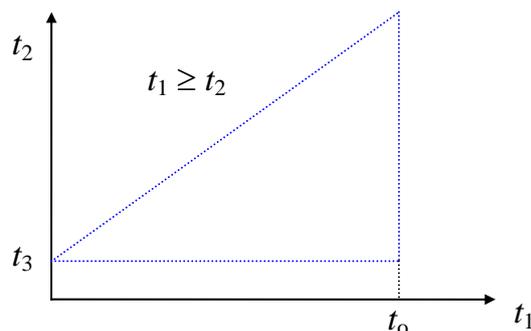
No contexto de otimização os problemas são tratados dentro das seguintes definições:

função objetivo: é a função matemática cujo máximo ou mínimo deseja-se determinar. No exemplo acima a função objetivo é o custo total.

variáveis de decisão: são as variáveis independentes que aparecem na função objetivo. Correspondem, em número, ao excesso de variáveis em relação ao número de equações (restrições de igualdade), isto é, o **grau de liberdade** do sistema. No exemplo acima tem-se 6 equações (3 para A_i e 3 para W_i) e 8 variáveis (t_1 , t_2 , A_i e W_i), portanto 2 variáveis de decisão ou grau de liberdade igual a 2.

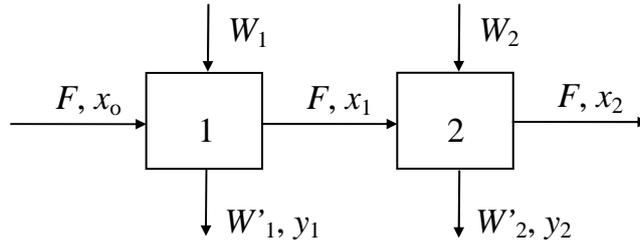
restrições: são os limites impostos ao sistema ou estabelecidos pelas leis naturais que governam o comportamento do sistema, a que estão sujeitas as variáveis de decisão. As restrições podem ser de igualdade (equações) ou de desigualdade (inequações). No exemplo acima tem-se 6 restrições de igualdade e 3 restrições de desigualdade: $t_3 \leq t_2 \leq t_1 \leq t_0$.

região de busca: ou região viável, é a região do espaço definido pelas variáveis de decisão, delimitada pelas restrições, em cujo interior ou em cuja fronteira se localiza o ótimo da função objetivo. No exemplo acima, tomando t_1 e t_2 como variáveis de decisão, a região de busca é aquela delimitada pelas restrições de desigualdade.



No exemplo acima, o problema de otimização foi gerado pela busca de um projeto ótimo do processo. Outra aplicação típica de técnicas de otimização aparece na melhoria das condições de operação de um processo em produção.

Exemplo 1.2: No processo de extração por solvente puro, ilustrado abaixo, deseja-se encontrar a condição de operação com a maior lucratividade possível.



onde W_i e W'_i são vazões mássicas de solvente, F é a vazão mássica de água, x_i é a massa de soluto por unidade de massa de água e y_i é a massa de soluto por unidade de massa de solvente. Uma análise econômica levou a seguinte expressão do lucro do sistema:

$$\begin{aligned} \text{lucro:} \quad & L = R - C \\ \text{receita:} \quad & R = P_s (W'_1 y_1 + W'_2 y_2) \\ \text{custo:} \quad & C = P_x (W_1 + W_2) \\ \text{restrição:} \quad & R > C \end{aligned}$$

onde P_s é o preço do soluto no extrato, P_x é o preço do solvente puro. Uma análise técnica gerou as seguintes relações restritivas:

$$\begin{aligned} \text{balanços de massa para o soluto:} \quad & F x_0 - W'_1 y_1 - F x_1 = 0 \\ & F x_1 - W'_2 y_2 - F x_2 = 0 \end{aligned}$$

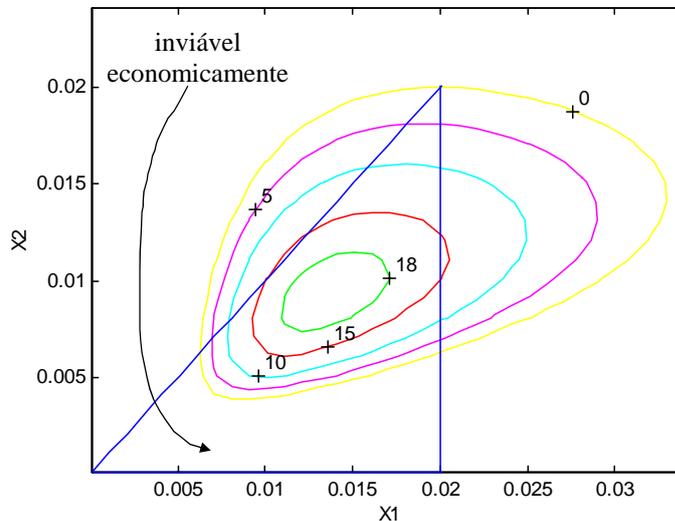
$$\begin{aligned} \text{balanços de massa para o solvente:} \quad & W_1 - W'_1 - s F = 0 \\ & W_2 + s F - W'_2 - s F = 0 \end{aligned}$$

$$\begin{aligned} \text{relações de equilíbrio:} \quad & y_1 = m x_1 \\ & y_2 = m x_2 \end{aligned}$$

onde s é a solubilidade do solvente em água (massa de solvente / massa de água) e m é a constante de equilíbrio entre as fases. Portanto, dados F , x_0 , s , m , P_s e P_x , o problema de extrair o soluto da água da maneira mais lucrativa possível, consiste em maximizar L em função das condições de operação. Este exemplo possui também 6 equações (4 equações de balanço de massa e 2 equações de equilíbrio) e 8 variáveis (W_i , W'_i , y_i , x_1 e x_2), tendo portanto 2 variáveis de decisão, ou dois graus de liberdade. Tomando x_1 e x_2 como variáveis de decisão, a região de busca para o problema de otimização está delimitada pelas seguintes restrições de desigualdade:

$$\begin{aligned} x_0 > x_1 > x_2 > 0 \\ L(x_1, x_2) > 0 \end{aligned}$$

A figura abaixo ilustra a região de busca para $F = 1,0 \times 10^4$ kg-água / h, $x_0 = 0,02$ kg-soluto / kg-água, $s = 7,0 \times 10^{-4}$ kg-solvente / kg-água, $m = 4,0$ kg-água / kg-solvente, $P_s = 0,4$ R\$ / kg-soluto e $P_x = 0,01$ R\$ / kg-solvente.



Outra aplicação muito comum, que utiliza as técnicas de otimização, é o ajuste de modelos matemáticos a dados experimentais (ajuste de curvas), onde a função objetivo utilizada, $S(x)$, é formada pelos quadrados dos desvios do modelo em relação aos dados experimentais (mínimos quadrados):

$$S(x) = \sum_{j=1}^N [y_j(x) - y_j^{\text{exp}}]^2$$

onde $y(x)$ é o modelo proposto (ou tipo de curva) para ser ajustado aos N dados experimentais y^{exp} , e x são os parâmetros de ajuste do modelo. Ou, se for utilizado o método da máxima verossimilhança:

$$S(x) = \sum_{j=1}^N \frac{1}{\sigma_j^2} [y_j(x) - y_j^{\text{exp}}]^2$$

onde σ_j^2 são as variâncias dos dados experimentais.

1.3 Procedimento geral

Não existe um único método que pode ser aplicado eficientemente para todos os problemas. O método escolhido para um caso particular depende das características da função objetivo, da natureza das restrições e do número de variáveis do problema. Contudo, os seguintes passos devem ser observados na formulação e solução de um problema de otimização:

- 1) Análise do processo e suas variáveis;
- 2) Determinação do critério para otimização e especificação da função objetivo em termos das variáveis do processo;
- 3) Desenvolvimento do modelo para o processo, relacionando as suas variáveis através de restrições de igualdade e desigualdade. Cálculo do grau de liberdade do sistema;
- 4) Se a formulação do problema é de dimensão elevada, então procurar particionar o problema em partes menores ou simplificar a função objetivo e/ou o modelo;
- 5) Aplicação de técnicas apropriadas de otimização;
- 6) Analisar a solução obtida e a sua sensibilidade frente a variações em parâmetros do modelo e suas considerações (hipóteses).

1.4 Dificuldades encontradas

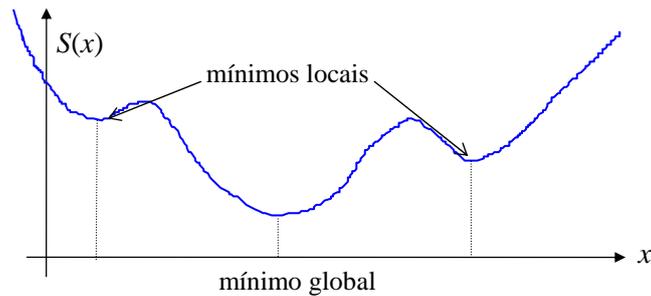
Problemas de otimização que apresentam funções objetivo e/ou restrições complicadas podem apresentar grandes dificuldades para obter a solução pelo uso de algumas técnicas de otimização. Algumas destas complicações estão listadas abaixo:

- função objetivo e/ou restrições podem apresentar descontinuidades;
- função objetivo e/ou restrições não lineares;
- função objetivo e/ou restrições podem estar definidas em termos de complicadas interações entre as variáveis, podendo ocorrer valores não únicos destas variáveis para o valor ótimo da função objetivo;
- função objetivo e/ou restrições podem exibir patamares (pouca sensibilidade à variação das variáveis) para alguns intervalos das variáveis e comportamento exponencial (muita sensibilidade) para outros intervalos;
- função objetivo pode apresentar ótimos locais.

2. Conceitos Básicos

2.1 Mínimos e máximos

Seja uma função $S: X \subseteq \mathfrak{R}^n \rightarrow \mathfrak{R}$. Diz-se que x^* é um mínimo global (ou absoluto) de S se $S(x^*) \leq S(x) \forall x \in X$, e que x^* é um mínimo local (ou relativo) de S se existe $\varepsilon > 0$, tal que $S(x^*) \leq S(x) \forall x$ tal que $\|x - x^*\| < \varepsilon$. Se as desigualdades forem estritas, isto é, $S(x^*) < S(x)$ tem-se mínimos globais e locais estritos.



Se existe um número β tal que $S(x) \geq \beta \forall x \in X$ e, para um ε suficientemente pequeno, $S(x) < \beta + \varepsilon$ para algum $x \in X$, então β é o ínfimo (ou valor inferior) de $S(x)$. Considerando os pontos $\pm\infty$, então toda função $S(x)$ tem um ínfimo e um supremo (ou valor superior) em X . Nem toda função tem mínimo (máximo), mas se ele existir deve ser finito e é obtido no ínfimo (supremo), isto é,

$$S(x^*) = \min_{x \in X} S(x) = \inf_{x \in X} S(x) \quad [S(x^*) = \max_{x \in X} S(x) = \sup_{x \in X} S(x)]$$

Por exemplo, a função $S(x) = e^x$ não tem máximo em $X = \mathfrak{R}$ e a função $S(x) = e^{-x}$ não tem mínimo em $X = \mathfrak{R}$. Contudo, o supremo de e^x é $+\infty$, e o ínfimo de e^{-x} é zero.

2.2 Condições para a otimalidade

2.2.1 Otimização sem restrição

No caso da otimização sem restrições, onde deseja-se encontrar os pontos extremos da função objetivo, tem-se as seguintes condições de otimalidade.

- Condição necessária de primeira ordem:

Para que x^* seja um mínimo (máximo) local da função $S(x)$, diferenciável em x^* , é necessário que:

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

- Condição necessária de segunda ordem:

Para que x^* seja um mínimo (máximo) local da função $S(x)$, duas vezes diferenciável em x^* , é necessário que:

$$\nabla S(x^*) = 0 \text{ e que}$$

$$H(x^*) \equiv \nabla^2 S(x^*) \text{ seja positiva (negativa) semidefinida}$$

onde $H(x^*)$ é chamada de matriz Hessiana.

Observa-se que estas condições são apenas necessárias porque os termos de primeira e segunda ordem podem estar nulos, deixando ainda dúvida sobre a natureza de x^* .

• Condição suficiente:

Seja $S(x)$ duas vezes diferenciável em x^* tal que:

$$\nabla S(x^*) = 0 \text{ e}$$

$H(x^*)$ seja positiva (negativa) definida

então x^* é um mínimo (máximo) local estrito de S .

Pode-se analisar a condição da matriz Hessiana, $H(x^*)$, pelas seguintes formas:

1) Pela sua contribuição no termo de segunda ordem da expansão em série de Taylor em torno do ponto ótimo.

$$S(x) - S(x^*) = \frac{1}{2}(x - x^*)^T H(x^*)(x - x^*) + \dots$$

2) Pelos sinais dos valores característicos de $H(x^*)$.

Decompondo a matriz Hessiana em seus valores e vetores característicos:

$$H(x^*) = V \Lambda V^{-1}$$

onde V é a matriz dos vetores característicos (nas colunas) e Λ é a matriz dos valores característicos (na diagonal). Definindo $z(x) = V^{-1}(x - x^*)$ e lembrando que sendo a matriz Hessiana simétrica então $V^{-1} = V^T$ (matriz ortogonal) e $(x - x^*)^T V = z^T$. Desta forma a expansão em série de Taylor pode ser escrita como:

$$S(x) - S(x^*) = \frac{1}{2} z^T \Lambda z + \dots = \frac{1}{2} \sum_{i=1}^n \lambda_i z_i^2 + \dots$$

3) Pelos sinais dos determinantes das primeiras menores principais de $H(x^*)$ (critério de Sylvester).

A *menor* M_{ij} de uma matriz H é definida como a matriz obtida pela remoção da i -ésima linha e da j -ésima coluna de H . Uma *menor principal* de ordem k é uma matriz obtida pela remoção de quaisquer $n - k$ colunas e suas linhas correspondentes de uma matriz de ordem n . A *primeira menor principal* de ordem k de uma matriz H , denotada por $M_k(H)$, é obtida pela remoção das últimas $n - k$ colunas e linhas da matriz H . Observa-se que os determinantes das primeiras menores principais de ordem $1, 2, \dots, n$ da matriz Λ são, respectivamente: $\lambda_1, \lambda_1 \lambda_2, \dots, \lambda_1 \lambda_2 \lambda_3 \dots \lambda_n$.

Na tabela a seguir apresenta-se as relações entre a matriz Hessiana e as três formas de analisar a sua condição.

$H(x^*)$	Taylor	λ	$\Delta_k = \det(M_k)$
positiva semidefinida	$x^T H x \geq 0$	≥ 0	≥ 0
positiva definida	$x^T H x > 0$	> 0	> 0
negativa semidefinida	$x^T H x \leq 0$	≤ 0	$(-1)^k \Delta_k \geq 0$
negativa definida	$x^T H x < 0$	< 0	$(-1)^k \Delta_k > 0$
não definida	$x^T H x \begin{matrix} > 0 \\ < 0 \end{matrix}$	$\begin{matrix} > 0 \\ < 0 \end{matrix}$	\neq dos acima

Desta forma, pode-se afirmar que x^* é:

- ponto de mínimo local se $H(x^*)$ for positiva definida, $S(x) > S(x^*)$
- ponto de máximo local se $H(x^*)$ for negativa definida, $S(x) < S(x^*)$
- ponto de sela se $H(x^*)$ for não definida, ora $S(x) > S(x^*)$ e ora $S(x) < S(x^*)$

Nas situações onde $H(x^*)$ é semidefinida deve-se ainda investigar os termos de ordem superior da expansão em série de Taylor.

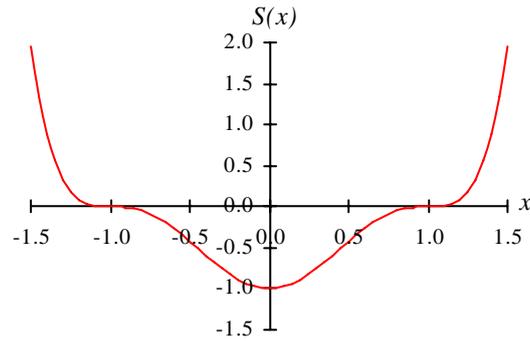
Exemplo 2.1: $S(x) = (x^2 - 1)^3$

$$\nabla S(x) = 6x(x^2 - 1)^2 \rightarrow \nabla S(x) = 0 \quad \begin{cases} x_1 = 0 \\ x_2 = 1 \\ x_3 = -1 \end{cases}, \text{ satisfazem a condição necessária de}$$

primeira ordem;

$H(x) = (x^2 - 1)(30x^2 - 6) \rightarrow H(x_1) = 6; H(x_2) = 0; H(x_3) = 0$ satisfazem a condição necessária de segunda ordem; contudo somente x_1 satisfaz a condição suficiente. Neste caso (univariável) x_2 e x_3 são pontos de inflexão, como pode ser visto no gráfico abaixo, ou avaliando o valor da derivada terceira de $S(x)$ nestes pontos:

$$\nabla^3 S(x) = 24x(5x^2 - 3) \rightarrow \nabla^3 S(x_2) = 48 \neq 0 \text{ e } \nabla^3 S(x_3) = -48 \neq 0.$$



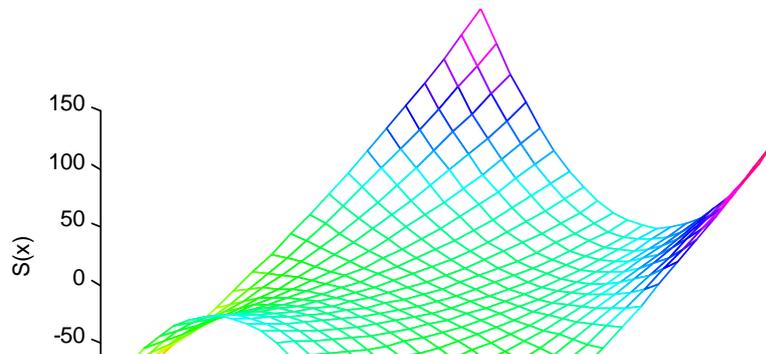
Exemplo 2.2: $S(x_1, x_2) = x_1^2 + x_1 x_2^2$

$$\nabla S(x) = \begin{bmatrix} 2x_1 + x_2^2 \\ 2x_1 x_2 \end{bmatrix}$$

então, $x_1^* = x_2^* = 0$, e:

$$H(x) = \begin{bmatrix} 2 & 2x_2 \\ 2x_2 & 2x_1 \end{bmatrix} \rightarrow H(x^*) = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

isto é, uma matriz positiva semidefinida. O gráfico abaixo ilustra a função $S(x)$, onde observa-se que $x^* = 0$ não é um ponto de mínimo. Fazendo a mesma análise com a mudança de variável $y = x_2^2$, verifica-se que a origem é um ponto sela.



2.2.2 Otimização com restrições

Seja o problema de otimização sujeito a restrições de igualdade, $h(x)$, e desigualdade, $g(x)$:

$$\begin{aligned} & \min S(x) \\ & \text{sujeito a: } h_j(x) = 0, \quad j = 1, 2, \dots, m \\ & \quad \quad \quad g_j(x) \leq 0, \quad j = 1, 2, \dots, p \\ & \quad \quad \quad x \in X \subseteq \mathfrak{R}^n \end{aligned}$$

onde $S(x)$, $g(x)$, e $h(x) \in C^2$. O conjunto de todos os pontos viáveis é definido por:

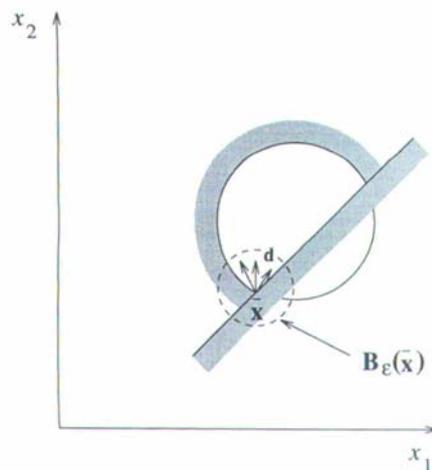
$$K = \{x \in X \subseteq \mathfrak{R}^n / h(x) = 0, g(x) \leq 0\}$$

Uma restrição de desigualdade $g_j(x)$ é chamada de *ativa* em um ponto viável \bar{x} se $g_j(\bar{x}) = 0$, caso contrário ela é uma *restrição inativa*. As restrições ativas restringem a região de viabilidade, enquanto que as inativas não impõem restrição alguma na vizinhança do ponto \bar{x} , definida pela hiperesfera de raio ε em torno deste ponto, denotada por $B_\varepsilon(\bar{x})$.

Um vetor d é chamado de *vetor de direção viável* a partir do ponto \bar{x} se existe uma hiperesfera de raio ε tal que:

$$(\bar{x} + \lambda d) \in \{B_\varepsilon(\bar{x}) \cap K\} \text{ para todo } 0 \leq \lambda \leq \frac{\varepsilon}{\|d\|}.$$

O conjunto de vetores de direções viáveis a partir de \bar{x} é chamado de *cone de direções viáveis* de K no ponto \bar{x} . A figura abaixo ilustra estas definições.



Se $d \neq 0$, então \bar{x} deve satisfazer as seguintes condições:

$$d^T \nabla h(\bar{x}) = 0$$

$$d^T \nabla g(\bar{x}) \leq 0 \text{ para as } g(\bar{x}) \text{ ativas, pois } g(x) \approx \overbrace{g(\bar{x})}^{=0} + \lambda \nabla^T g(\bar{x}) d \leq 0$$

e se $d^T \nabla S(\bar{x}) < 0$, então d é uma direção viável e promissora, isto é,

$$S(\bar{x} + \lambda d) < S(\bar{x}) \text{ para todo } 0 \leq \lambda \leq \frac{\varepsilon}{\|d\|}, \text{ pois } S(x) - S(\bar{x}) \approx \lambda \nabla^T S(\bar{x}) d < 0.$$

Se $\bar{x} = x^*$ é um ponto de mínimo local do problema, então para um λ suficientemente pequeno, tem-se:

$$S(x^*) \leq S(x^* + \lambda d).$$

A idéia chave para desenvolver as condições necessárias e suficientes para um problema de otimização com restrições é transformá-lo em um problema de otimização sem restrições e aplicar as condições para este caso. Uma forma de fazer esta transformação é através da introdução de uma função auxiliar, chamada de função de Lagrange, $L(x, \lambda, \mu)$, definida como:

$$L(x, \lambda, \mu) = S(x) + \lambda^T h(x) + \mu^T g(x), \quad \mu \geq 0$$

onde λ e μ são os multiplicadores de Lagrange associados com as restrições de igualdade e desigualdade, respectivamente (μ são também conhecidos como multiplicadores de Kuhn-Tucker). Deste modo, o problema transformado torna-se:

$$\max_{\lambda, \mu \geq 0} \min_x L(x, \lambda, \mu)$$

onde os multiplicadores λ associados com as restrições de igualdade, $h(x) = 0$, assumem sinais positivos quando $h(x) \geq 0$ e negativos quando $h(x) \leq 0$. No ponto ótimo tem-se:

$$L(x^*, \lambda^*, \mu^*) = S(x^*)$$

Cada multiplicador de Lagrange indica o quão sensível é a função objetivo em relação à restrição associada. Por exemplo, se as restrições de igualdade são perturbadas por um vetor b , isto é, $h_b(x) = b$, então:

$$\nabla_b S(x^*) = -\lambda^*.$$

Note que neste caso a função de Lagrange tem a forma:

$$L(x, \lambda) = S(x) + \lambda^T [h_b(x) - b]$$

e sua sensibilidade em relação ao parâmetro b é dada por:

$$\nabla_b L = \nabla_b^T x [\nabla_x S(x) + \nabla_x^T h_b(x) \lambda] + \nabla_b^T \lambda [h_b(x) - b] - \lambda$$

Como os termos entre colchetes da expressão acima devem ser nulos no ponto ótimo (condição necessária de primeira ordem), então:

$$\nabla_b L(x^*, \lambda^*) = -\lambda^* \quad \text{e} \quad \nabla_b S(x^*) = -\lambda^*$$

pois $\nabla_b L(x^*, \lambda^*) = \nabla_b S(x^*) + \nabla_b^T \lambda [h_b(x^*) - b] + \nabla_b^T h_b(x^*) \lambda^* - \lambda^*$,

$$h_b(x) = b \quad \text{e} \quad \nabla_b h_b(x) = I$$

Portanto, o valor de $S(x)$ aumenta ou diminui a partir de $S(x^*)$ com um aumento ou diminuição em b , dependendo do sinal de λ^* . Por isso, os multiplicadores de Lagrange são também conhecidos como “*shadow prices*” ou “custos marginais” das restrições, porque a mudança no valor ótimo da função objetivo por unidade de acréscimo no lado direito da restrição de igualdade é dado por λ^* .

Exemplo 2.3: Seja o seguinte problema de otimização com restrição

$$\min S(x) = (x_1 - 5)^2 + (x_2 - 5)^2$$

$$\text{sujeito a: } h(x) = x_1 - x_2 = 0$$

Introduzindo uma perturbação na restrição de igualdade do tipo: $x_1 - x_2 = b$, então a função de Lagrange toma a forma:

$$L(x, \lambda) = (x_1 - 5)^2 + (x_2 - 5)^2 + \lambda (x_1 - x_2 - b)$$

cujos gradientes nulos com relação a x_1 , x_2 e λ leva ao seguinte sistema de equações:

$$2(x_1 - 5) + \lambda = 0$$

$$2(x_2 - 5) - \lambda = 0$$

$$x_1 - x_2 - b = 0$$

resultando na solução ótima: $x_1^* = 5 + b/2$, $x_2^* = 5 - b/2$ e $\lambda^* = -b$.

Deste modo $S(x^*) = b^2/2$ e $\nabla_b S(x^*) = b = -\lambda^*$.

Para entender a origem da função de Lagrange, o ótimo do exemplo acima deve satisfazer as seguintes condições:

$$dS = \frac{\partial S}{\partial x_1} \delta x_1 + \frac{\partial S}{\partial x_2} \delta x_2 = 0$$

$$dh = \frac{\partial h}{\partial x_1} \delta x_1 + \frac{\partial h}{\partial x_2} \delta x_2 = 0$$

Se $S(x)$ fosse uma função sem restrição, então as suas duas derivadas parciais seriam nulas no ponto ótimo e $dS(x^*)$ seria nulo para quaisquer valores das variações δx_1 e δx_2 . Entretanto, como as variáveis x_1 e x_2 estão restritas (δx_1 e δx_2 não são independentes), as duas derivadas parciais de $S(x)$ não podem ser arbitrariamente

igualadas a zero. Contudo, $S(x)$ deve ser um extremo no ponto ótimo e portanto $dS(x^*) = 0$. A segunda condição, $dh(x^*) = 0$, existe porque $h(x) = 0$. Para se obter uma solução (δx_1 e δx_2) não trivial do sistema de equações acima, a matriz dos coeficientes do sistema:

$$\begin{bmatrix} \frac{\partial S}{\partial x_1} & \frac{\partial S}{\partial x_2} \\ \frac{\partial h}{\partial x_1} & \frac{\partial h}{\partial x_2} \end{bmatrix}$$

deve ter determinante nulo, ou seja, as linhas da matriz são linearmente dependentes:

$$\frac{\partial S}{\partial x_1} + \lambda \frac{\partial h}{\partial x_1} = 0 \quad \text{e} \quad \frac{\partial S}{\partial x_2} + \lambda \frac{\partial h}{\partial x_2} = 0$$

Então, definindo uma função auxiliar: $L(x, \lambda) = S(x) + \lambda^T h(x)$

as condições acima são satisfeitas se: $\nabla_x L(x, \lambda) = 0$. Para que a restrição de igualdade, $h(x) = 0$, seja também satisfeita é necessário que $\nabla_\lambda L(x, \lambda) = 0$. Portanto, no ponto ótimo é necessário que $\nabla L(x^*, \lambda^*) = 0$.

A existência dos multiplicadores de Lagrange depende da forma das restrições, e estará garantida se e somente se os gradientes das restrições de desigualdade ativas, $\nabla g_j(x^*)$, e das restrições de igualdade, $\nabla h(x^*)$, forem linearmente independentes. Por exemplo, no caso de um problema somente com restrições de igualdade, a condição necessária de primeira ordem para $L(x, \lambda)$ fica:

$$\nabla_x S(x) + [\nabla_x h(x)]^T \lambda = 0$$

cuja solução para λ existirá somente se a matriz $\nabla_x h(x)$ possuir posto completo, m , isto é, estar composta por m vetores linearmente independentes.

• Condição necessária de primeira ordem de Karush-Kuhn-Tucker (KKT):

Para que x^* seja um ótimo local do problema com restrições, com $S(x)$, $g(x)$, e $h(x)$ diferenciáveis em x^* , é necessário que:

os gradientes das restrições de desigualdade ativas, $\nabla g_j(x^*)$, e das restrições de igualdade, $\nabla h(x^*)$, sejam linearmente independentes (*qualificação de segunda ordem das restrições*), e que as seguintes condições sejam satisfeitas:

$$\nabla_x L(x^*, \lambda^*, \mu^*) = \nabla S(x^*) + (\lambda^*)^T \nabla h(x^*) + (\mu^*)^T \nabla g(x^*) = 0$$

$$h(x^*) = 0$$

$$g(x^*) \leq 0$$

$$\mu_j^* g_j(x^*) = 0, j = 1, 2, \dots, p \text{ (condições de complementaridade)}$$

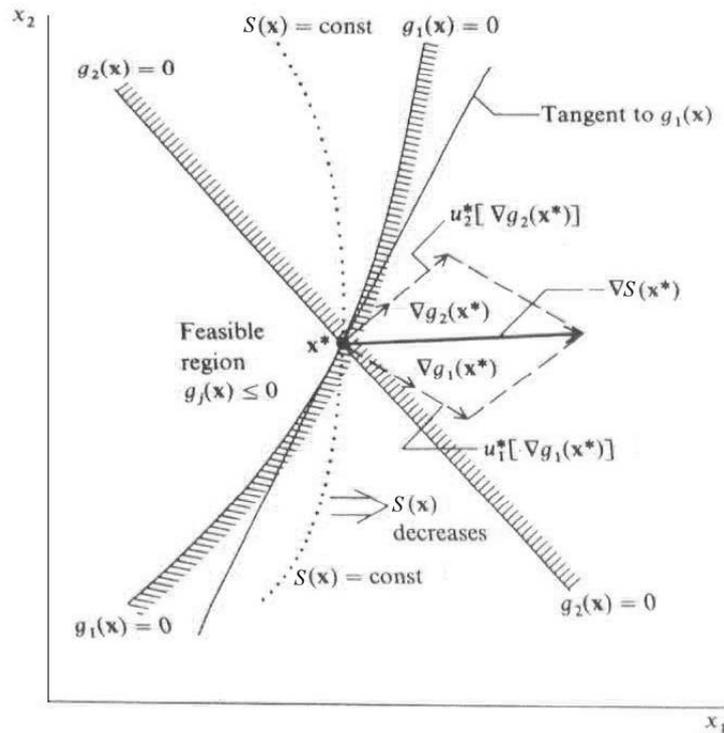
$$\mu^* \geq 0$$

A condição de independência linear pode ser relaxada por outras qualificações de primeira e segunda ordens das restrições (Floudas, 1995, pg. 59 e 64).

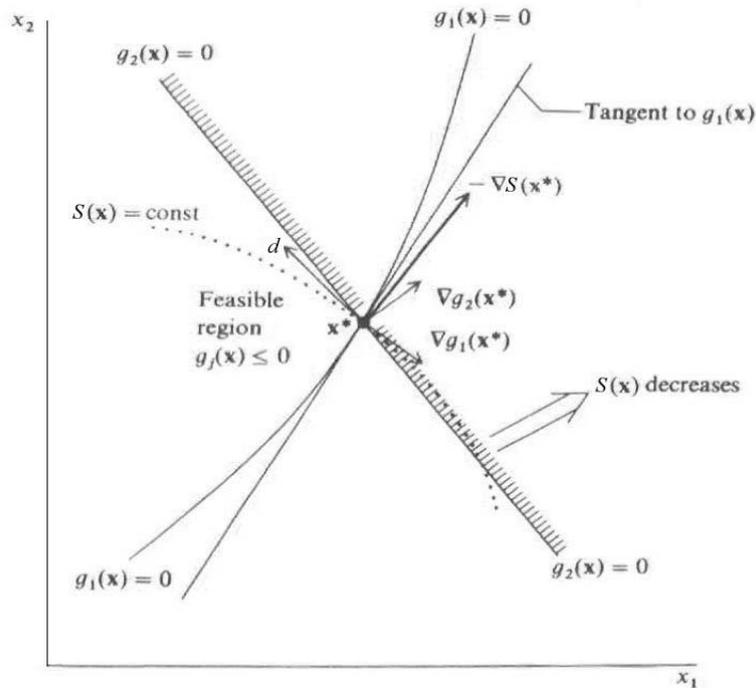
A condição do gradiente nulo, $\nabla_x L(x^*, \lambda^*, \mu^*) = 0$, implica em:

$$(\lambda^*)^T \nabla h(x^*) + (\mu^*)^T \nabla g(x^*) = -\nabla S(x^*)$$

que interpretada graficamente, figura abaixo, mostra que o vetor $-\nabla S(x^*)$ pertence ao cone das direções viáveis, formado pelos gradientes das restrições de igualdade e desigualdade ativas (uma vez que $\mu_j^* = 0$ para as restrições inativas).



Supondo que $-\nabla S(x^*)$ caia fora do cone das direções viáveis, então haveria uma direção d tal que $d^T \nabla S(x^*) < 0$, $d^T \nabla g(x^*) \leq 0$ e $d^T \nabla h(x^*) = 0$, isto é, existiria um ponto melhor que x^* , como ilustra a figura abaixo.



• Condição necessária de segunda ordem de KKT:

Para que x^* seja um mínimo local do problema com restrições, com $S(x)$, $g(x)$, e $h(x)$ duas vezes diferenciáveis em x^* , é necessário que:

a condição de primeira ordem de KKT seja satisfeita e, que a matriz Hessiana da função de Lagrange, $\nabla_x^2 L(x^*, \lambda^*, \mu^*)$, seja positiva semidefinida para todo vetor não nulo d tal que:

$$d^T \nabla h_i(x^*) = 0, \quad i = 1, 2, \dots, m$$

$$d^T \nabla g_j(x^*) = 0 \text{ para as } g_j(x^*) \text{ ativas}$$

isto é, $d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d \geq 0$.

• Condição suficiente de KKT:

Para que x^* seja um mínimo local do problema com restrições, com $S(x)$, $g(x)$, e $h(x)$ duas vezes diferenciáveis em x^* , é suficiente que:

a condição de primeira ordem de KKT seja satisfeita e, que a matriz Hessiana da função de Lagrange, $\nabla_x^2 L(x^*, \lambda^*, \mu^*)$, seja positiva definida para todo vetor não nulo d tal que:

$$d^T \nabla h_i(x^*) = 0, \quad i = 1, 2, \dots, m$$

$$d^T \nabla g_j(x^*) = 0 \text{ para as } g_j(x^*) \text{ ativas } \{g_j(x^*) = 0 \text{ e } \mu_j^* > 0\}$$

$$d^T \nabla g_j(x^*) \leq 0 \text{ para as } g_j(x^*) \text{ inativas } \{g_j(x^*) < 0 \text{ e } \mu_j^* = 0\}$$

isto é, $d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d > 0$.

A positividade da matriz Hessiana com restrição, isto é:

$$d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d > 0 \quad \forall d \in \{d / d^T \nabla h_i(x^*) = 0, d^T \nabla g_j(x^*) = 0, d \neq 0\}$$

é garantida se todas as raízes do polinômio característico

$$p(\lambda) = \begin{vmatrix} \lambda I - \nabla_x^2 L & M \\ M^T & 0 \end{vmatrix} = 0$$

forem positivas, onde M é a matriz formada pelos gradientes de $h(x^*)$ e $g(x^*)$ ativas, isto é, a matriz tal que $d^T M = 0$, com $m+p^a < n$ e com posto completo (p^a é o número de restrições g ativas). O mesmo critério se aplica para semipositividade, negatividade e seminegatividade, com os respectivos sinais das raízes.

Uma maneira de resolver o problema de valor característico acima é usando a decomposição QR (ou decomposição ortogonal-triangular, ou decomposição de Householder) da matriz $M (= Q R)$ para obter a matriz de projeção, Z^T , do vetor d no sub-espaço nulo de M , isto é, $d^T M = 0$, ou então a decomposição em valores singulares da matriz $M^T (= U S V^H)$. A matriz Z é formada pelas últimas w colunas de Q , ou ainda pelas últimas w colunas de V , onde w é a dimensão do espaço nulo (número de valores singulares nulos, ou número de linhas nulas da matriz R):

$$Z_{i,j} = Q_{i,j} = V_{i,j} \quad i = 1, 2, \dots, n, \quad j = m+p^a-w+1, \dots, m+p^a$$

onde $Q^H M = \begin{bmatrix} R \\ 0 \end{bmatrix}$ (Q é uma matriz unitária e $Q^{-1} = Q^H$ é a transposta conjugada)

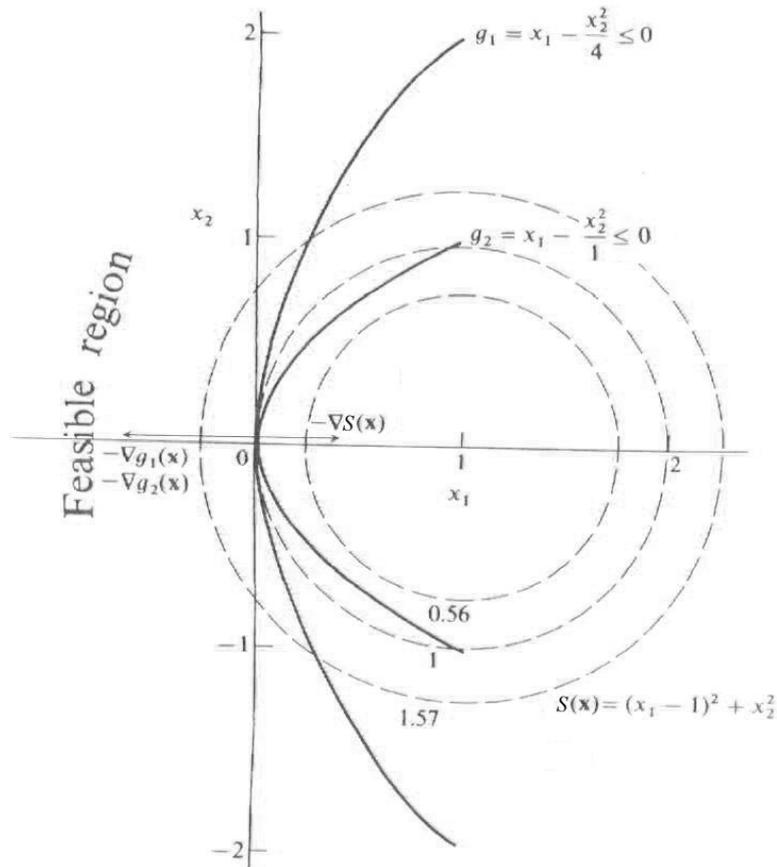
Uma vez encontrada a matriz Z , obtém-se os valores característicos da matriz Hessiana projetada neste sub-espaço: $Z^T \nabla_x^2 L Z$.

Exemplo 2.4: Verificar as condições necessárias e suficientes para o seguinte problema (Edgar & Himmelblau, 1988, pg. 314):

$$\begin{aligned} \min S(x) &= (x_1 - 1)^2 + x_2^2 \\ \text{sujeito a: } g_1(x) &= x_1 - x_2^2 / 4 \leq 0 \end{aligned}$$

Exemplo 2.5: Verificar as condições necessárias e suficientes para o problema com a mesma função objetivo do exemplo 2.4, mas usando a seguinte restrição:

$$g_2(x) = x_1 - x_2^2 \leq 0$$



Exemplo 2.6: Verificar se o ponto $x^* = [1 \ 4,9]^T$ é um mínimo local para o seguinte problema (Edgar & Himmelblau, 1988 , pg. 316):

$$\min S(x) = 4 x_1 - x_2^2 - 12$$

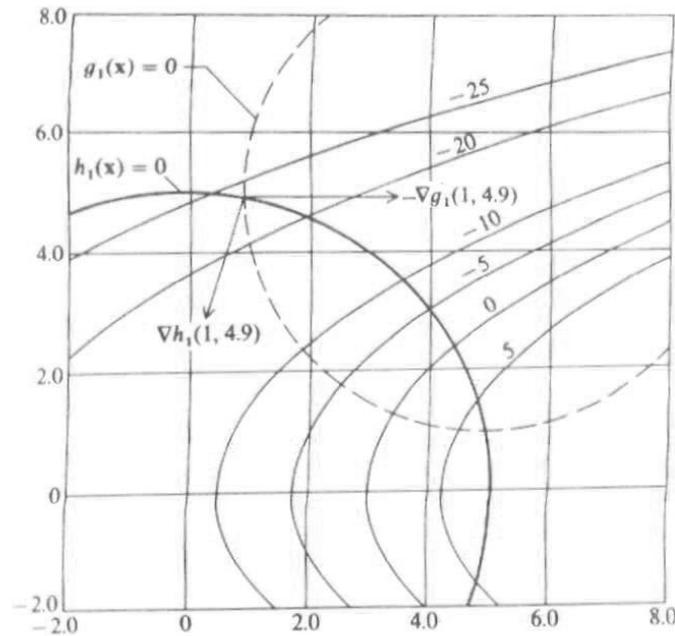
$$\text{sujeito a: } h_1(x) = 25 - x_1^2 - x_2^2 = 0$$

$$g_1(x) = x_1^2 - 10 x_1 + x_2^2 - 10 x_2 + 34 \leq 0$$

$$g_2(x) = -(x_1 - 3)^2 - (x_2 - 1)^2 \leq 0$$

$$g_3(x) = -x_1 \leq 0$$

$$g_4(x) = -x_2 \leq 0$$



2.3 Convexidade

Um subconjunto K de um espaço vetorial X é dito convexo se $\forall x_1, x_2 \in K$ e $0 \leq \alpha \leq 1$:

$$\alpha x_1 + (1 - \alpha) x_2 \in K$$

e apresenta as seguintes propriedades:

- a) $\beta K = \{x \in K / x = \beta y, y \in K\}$ é convexo para $\forall \beta \in \mathfrak{R}$;
- b) $K + L$ e $K \cap L$ são convexos para \forall subconjunto convexo L de X .

Seja K um convexo não vazio do \mathfrak{R}^n . A função $S: K \rightarrow \mathfrak{R}$ é dita convexa se $\forall x_1, x_2 \in K$ e $0 \leq \alpha \leq 1$:

$$S[\alpha x_1 + (1 - \alpha) x_2] \leq \alpha S(x_1) + (1 - \alpha) S(x_2)$$

A função $S(x)$ é estritamente convexa se a desigualdade for estrita. Uma função $T(x)$ é côncava se a função $S(x) = -T(x)$ for convexa.

Sejam $S(x), S_i(x), i=1,2,\dots,n$, funções convexas em um convexo não vazio K do \mathfrak{R}^n , então as seguintes propriedades são apresentadas:

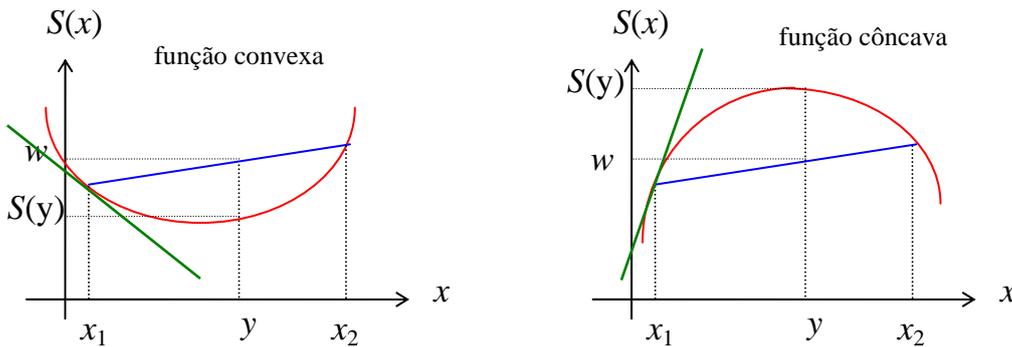
- a) $S(x)$ é contínua em qualquer ponto do interior de K ;
- b) as seções $K_\varepsilon = \{x \in K / S(x) \leq \varepsilon\}$ são conjuntos convexas;
- c) $S(x) = \sum_{i=1}^n S_i(x)$ é uma função convexa. Se no mínimo uma $S_i(x)$ é estritamente convexa, então $S(x)$ é estritamente convexa;

d) $\beta S(x)$ é convexa para $\beta > 0 \in \mathfrak{R}$.

e) se todas $S_i(x) < \infty \forall x \in K$, então $S(x) = \max\{S_1(x), S_2(x), \dots, S_n(x)\}$ é convexa.

Quando $S(x)$ é convexa, as condições de otimalidade simplificam-se, porque as condições de segunda ordem são equivalentes à convexidade local da função. Além disso, um mínimo local será também global e se a função for estritamente convexa o mínimo global é único.

Para ilustrar, define-se $y = \alpha x_1 + (1 - \alpha) x_2$ e $w = \alpha S(x_1) + (1 - \alpha) S(x_2)$



Seja X um conjunto aberto não vazio do \mathfrak{R}^n e $S(x)$ uma função diferenciável em $x^o \in X$. Se $S(x)$ é convexa em x^o , então

$$S(x) - S(x^o) \geq \nabla^T S(x^o)(x - x^o)$$

Para uma função $S(x)$ diferenciável em X ,

$$S(x) \text{ é convexa} \Leftrightarrow S(x^2) - S(x^1) \geq \nabla^T S(x^1)(x^2 - x^1) \quad \forall x_1, x_2 \in X.$$

Seja X um conjunto aberto não vazio do \mathfrak{R}^n e $S(x)$ uma função duas vezes diferenciável em $x^o \in X$. Se $S(x)$ é convexa em x^o , então $\nabla^2 S(x^o)$ é positiva semidefinida. Para uma função $S(x)$ duas vezes diferenciável em X ,

$$S(x) \text{ é convexa} \Leftrightarrow \nabla^2 S(x) \text{ é positiva semidefinida} \quad \forall x \in X.$$

Seja K um conjunto convexo não vazio do \mathfrak{R}^n , $x^o \in K$ e d um vetor não nulo tal que $(x^o + \alpha d) \in K$ para um $\alpha > 0$ suficientemente pequeno. Então, a *derivada direcional* de $S(x)$ no ponto x^o , ao longo da direção α , denotada por $S'(x^o, d)$, é definida pelo seguinte limite (incluindo $\pm\infty$):

$$S'(x^o, d) = \lim_{\alpha \rightarrow 0^+} \frac{S(x^o + \alpha d) - S(x^o)}{\alpha} \approx \lim_{\alpha \rightarrow 0^+} (\nabla^T S(x^o) d + \alpha d^T \nabla^2 S(x^o) d)$$

Portanto, a derivada direcional no ponto x^o é dada por $S'(x^o, d) = \nabla^T S(x^o) d$.

Seja K um conjunto convexo não vazio do \mathfrak{R}^n e $S(x)$ uma função convexa. Então, o sub-gradiente de $S(x)$ no ponto $x^o \in K$, denotado por d , é definido como:

$$S(x) \geq S(x^0) + d^T (x - x^0) \quad , \quad \forall x \in K$$

Isto é, a aproximação linear com o uso do vetor d sempre resulta em uma sub-estimativa de $S(x)$.

2.4.1 Generalização de funções convexas

Para generalizar a definição de funções convexas é necessário definir a continuidade de funções escalares multivariáveis.

Sejam $X \subseteq \mathfrak{R}^n$, $x^0 \in X$, e $S(x): X \rightarrow \mathfrak{R}$.

$S(x)$ é contínua em x^0 se:

para cada $\varepsilon > 0$ existir $\delta(\varepsilon)$ tal que $\|x - x^0\| < \delta \Rightarrow |S(x) - S(x^0)| < \varepsilon$, ou

para cada seqüência x^1, x^2, \dots, x^m em X convergindo para x^0 :

$$\lim_{m \rightarrow \infty} S(x^m) = S\left(\lim_{m \rightarrow \infty} x^m\right) = S(x^0).$$

$S(x)$ é contínua em X se ela for contínua para todo $x^0 \in X$.

$S(x)$ é semicontínua inferior em x^0 se:

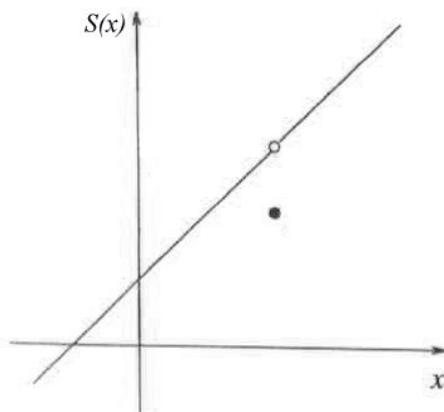
para cada $\varepsilon > 0$ existir $\delta(\varepsilon)$ tal que $\|x - x^0\| < \delta \Rightarrow -\varepsilon < S(x) - S(x^0)$, ou

para cada seqüência x^1, x^2, \dots, x^m em X convergindo para x^0 :

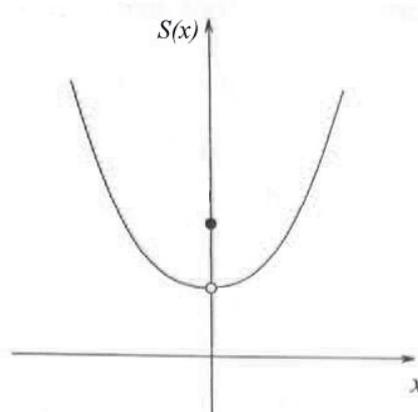
$$\liminf_{m \rightarrow \infty} S(x^m) = S\left(\lim_{m \rightarrow \infty} x^m\right) = S(x^0),$$

onde $\liminf_{m \rightarrow \infty} S(x^m) = \lim_{m \rightarrow \infty} \min\{S(x^1), S(x^2), \dots, S(x^m)\}$.

$S(x)$ é semicontínua inferior em X se ela for semicontínua inferior para todo $x^0 \in X$.



função semicontínua inferior



função semicontínua superior

$S(x)$ é semicontínua superior em x^0 se:

para cada $\varepsilon > 0$ existir $\delta(\varepsilon)$ tal que $\|x - x^0\| < \delta \Rightarrow S(x) - S(x^0) < \varepsilon$, ou

para cada seqüência x^1, x^2, \dots, x^m em X convergindo para x^0 :

$$\limsup_{m \rightarrow \infty} S(x^m) = S\left(\lim_{m \rightarrow \infty} x^m\right) = S(x^0),$$

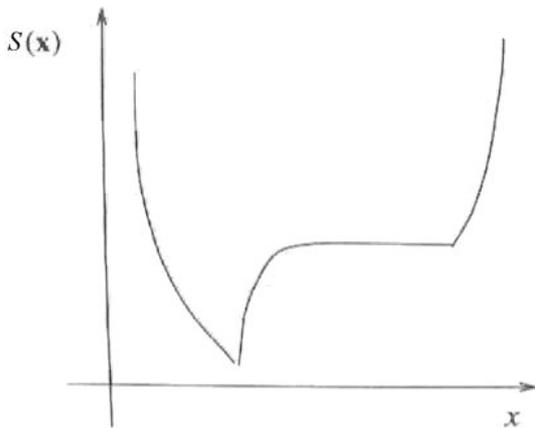
onde $\limsup_{m \rightarrow \infty} S(x^m) = \lim_{m \rightarrow \infty} \max\{S(x^1), S(x^2), \dots, S(x^m)\}$.

$S(x)$ é semicontínua superior em X se ela for semicontínua superior para todo $x^0 \in X$.

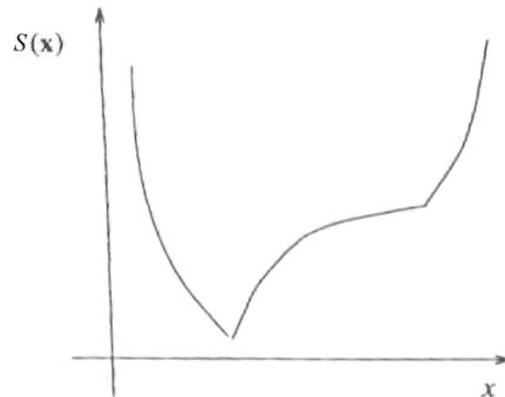
Seja K um convexo não vazio do \mathfrak{R}^n . A função $S: K \rightarrow \mathfrak{R}$ é dita *quasi-convexa* se $\forall x_1, x_2 \in K$ e $0 \leq \alpha \leq 1$:

$$S[\alpha x_1 + (1 - \alpha) x_2] \leq \max\{S(x_1), S(x_2)\}$$

A função $S(x)$ é estritamente quasi-convexa se a desigualdade for estrita quando $S(x_1) \neq S(x_2)$. Uma função $T(x)$ é (estritamente) *quasi-côncava* se a função $S(x) = -T(x)$ for (estritamente) quasi-convexa. Um mínimo local de uma função estritamente quasi-convexa será também global.



Função quasi-convexa



Função estritamente quasi-convexa

Definindo as seções $K_\varepsilon = \{x \in K / S(x) \leq \varepsilon\}$, então:

$$S(x) \text{ é quasi-convexa} \Leftrightarrow K_\varepsilon \text{ é convexo } \forall \varepsilon \in \mathfrak{R}.$$

Seja $S(x)$ uma função semicontínua inferior no convexo $K \subseteq \mathfrak{R}^n$. Se $S(x)$ é estritamente quasi-convexa em K , então $S(x)$ é quasi-convexa, mas o converso não é verdadeiro.

A soma de funções quasi-convexas não é necessariamente uma função quasi-convexa, propriedade válida para funções convexas. Por outro lado, o recíproco de uma função quasi-convexa, $1/S(x)$, é uma função quasi-côncava, mas o recíproco de uma função convexa não é necessariamente uma função côncava (já o recíproco de uma função côncava positiva é uma função convexa). Por exemplo, $S(x) = e^x$ é convexa e $1/S(x)$ é convexa também.

Seja $S(x)$ uma função diferenciável em um conjunto convexo aberto não vazio $K \subseteq \mathfrak{R}^n$, então:

$$S(x) \text{ é quasi-convexa} \Leftrightarrow S(x^2) \leq S(x^1) \rightarrow \nabla^T S(x^1)(x^2 - x^1) \leq 0 \quad \forall x_1, x_2 \in K.$$

Seja $S(x)$ uma função quasi-côncava duas vezes diferenciável em um conjunto convexo aberto não vazio $K \subseteq \mathfrak{R}^n$, então uma direção, d , ortogonal ao $\nabla S(x)$ exibe as seguintes propriedades:

- a) se $x^0 \in K$, $d \in \mathfrak{R}^n$ e $d^T \nabla S(x^0) = 0$, então $d^T \nabla^2 S(x^0) d \leq 0$;
- b) a matriz Hessiana de $S(x)$ tem no máximo um valor característico positivo $\forall x \in K$. Isto é, a generalização da concavidade de funções (para quasi-côncavidade) é equivalente à existência de no máximo um valor característico positivo da Hessiana.

Seja $S(x)$ uma função diferenciável em um conjunto aberto não vazio $X \subseteq \mathfrak{R}^n$.

$$S(x) \text{ é pseudo-convexa se } S(x^2) < S(x^1) \rightarrow \nabla^T S(x^1)(x^2 - x^1) < 0 \quad \forall x_1, x_2 \in X.$$

Uma função $T(x)$ é pseudo-côncava se a função $S(x) = -T(x)$ for pseudo-convexa. O recíproco de uma função pseudo-côncava, $1/S(x)$, é uma função pseudo-côncava. As seguintes relações são válidas:

- a) uma função convexa diferenciável é pseudo-convexa;
- b) uma função convexa é estritamente quasi-convexa;
- c) uma função convexa é quasi-convexa;
- d) uma função pseudo-convexa é estritamente quasi-convexa;
- e) uma função estritamente quasi-convexa e semicontínua inferior é quasi-convexa.

2.4 Formas funcionais

O método empregado para solucionar um problema de otimização depende, fundamentalmente, da forma da função objetivo e de suas restrições. Com relação a função objetivo tem-se:

função linear:
$$S(x) = a + c^T x = a + \sum_{i=1}^n c_i x_i$$

como neste caso $\nabla S(x) = c \neq 0 \forall x$, a função $S(x)$ não apresenta pontos de máximo ou mínimo. Para este tipo de função objetivo, só faz sentido a existência de um problema de otimização com restrição, cuja solução, se existir, estará localizada em algum ponto da fronteira da região de busca.

função quadrática:
$$S(x) = a + c^T x + \frac{1}{2} x^T A x = a + \sum_{i=1}^n c_i x_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$$

como $x^T A x$ é um escalar $\Rightarrow x^T A x = [x^T A x]^T = x^T A^T x$,

tem-se que:
$$x^T A x = \frac{1}{2} (x^T A x + x^T A^T x) = x^T \left[\frac{1}{2} (A + A^T) \right] x,$$

então definindo $Q = \frac{1}{2} (A + A^T)$, que é uma matriz simétrica: $Q = Q^T$,

resulta em: $x^T A x = x^T Q x$, isto é:

$$S(x) = a + c^T x + \frac{1}{2} x^T Q x = a + \sum_{i=1}^n c_i x_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j$$

e portanto:

$$\nabla S(x) = \left[\frac{\partial S}{\partial x_k} \right] = \left[c_k + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n q_{ij} \left(\frac{\partial x_i}{\partial x_k} x_j + x_i \frac{\partial x_j}{\partial x_k} \right) \right],$$

mas como $\frac{\partial x_i}{\partial x_k} = \delta_{ik} = \begin{cases} 0 & , i \neq k \\ 1 & , i = k \end{cases}$, tem-se:
$$\nabla S(x) = \left[c_k + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (q_{kj} x_j + q_{ik} x_i) \right]$$

e sabendo que a matriz Q é simétrica:

$$\nabla S(x) = \left[c_k + \sum_{i=1}^n q_{ik} x_i \right] = c + Qx$$

e
$$H(x) = \nabla^2 S(x) = Q$$

com isto, a condição necessária de primeira ordem, $\nabla S(x^*) = 0$, para este tipo de função resulta em:

$$Q x^* = -c \quad \text{ou de forma similar} \quad (x^*)^T Q = -c^T$$

e

$$S(x^*) = a + c^T x^* + \frac{1}{2} (x^*)^T Q x^* = a + \frac{1}{2} c^T x^*$$

Subtraindo $S(x^*)$ da função $S(x)$ tem-se:

$$S(x) = S(x^*) + c^T (x - x^*) + \frac{1}{2} [x^T Q x - (x^*)^T Q x^*]$$

como $c^T = -(x^*)^T Q$,

$$S(x) = S(x^*) + \frac{1}{2} [x^T Q x - (x^*)^T Q x^* - (x^*)^T Q (x - x^*) - (x^*)^T Q (x - x^*)]$$

$$S(x) = S(x^*) + \frac{1}{2} [x^T Q x - (x^*)^T Q x - (x^*)^T Q (x - x^*)]$$

$$S(x) = S(x^*) + \frac{1}{2} [(x - x^*)^T Q x - (x^*)^T Q (x - x^*)]$$

$$S(x) = S(x^*) + \frac{1}{2} (x - x^*)^T Q (x - x^*)$$

função não linear: $S(x): \mathfrak{R}^n \rightarrow \mathfrak{R}$

a expansão em série de Taylor de $S(x)$ em torno de um ponto x_0 é dada por:

$$S(x) = S(x_0) + \nabla^T S(x_0) (x - x_0) + \frac{1}{2} (x - x_0)^T \nabla^2 S(x_0) (x - x_0) + \dots$$

No caso particular de $x_0 = x^*$, tem-se $\nabla S(x^*) = 0$ e:

$$S(x) \approx S(x^*) + \frac{1}{2} (x - x^*)^T H(x^*) (x - x^*)$$

Se $S(x)$ for duas vezes diferenciável em x^* , então $H(x^*) = \left[\frac{\partial^2 S}{\partial x_i \partial x_j} \right]_{x^*}$ é simétrica.

Portanto, tanto no caso de função quadrática, onde $H(x^*) = Q$, quanto no caso geral de função não-linear, a matriz Hessiana fornece as características do(s) ponto(s) ótimo(s) de $S(x)$.

Algumas observações adicionais:

- 1) no caso de $S(x)$ ser uma função quadrática, o ponto ótimo é global;
- 2) se $H(x^*)$ é positiva definida, então $S(x)$ é estritamente convexa na vizinhança de x^* (ou $\forall x$ no caso da função quadrática);

- 3) se $H(x^*)$ é positiva semidefinida, então $S(x)$ é convexa na vizinhança de x^* (ou $\forall x$ no caso da função quadrática)..
- 4) se $H(x^*)$ é negativa definida, então $S(x)$ é estritamente côncava na vizinhança de x^* (ou $\forall x$ no caso da função quadrática);
- 5) se $H(x^*)$ é negativa semidefinida, então $S(x)$ é côncava na vizinhança de x^* (ou $\forall x$ no caso da função quadrática);

função mista linear e inteira: $S(x, y) = c^T x + d^T y$, $x \in X \subseteq \mathfrak{R}^n$ e $y \in Y$

onde y é um vetor de variáveis inteiras de dimensão q , $Y \subseteq \mathfrak{T}^q$, ou um vetor de variáveis binárias de dimensão q , $Y = \{0, 1\}^q$. Como a função é linear, só faz sentido a existência de um problema de otimização com restrição.

função mista não linear e inteira: $S(x, y): X \times Y \rightarrow \mathfrak{R}$

Problemas de otimização mista não linear e inteira apresentam grandes desafios e dificuldades associados com o caráter combinatorial do domínio discreto (inteiro) junto com as comumente encontradas não convexidades e não linearidades do domínio contínuo.

função unimodal: é a função que apresenta apenas um extremo.

função multimodal: é a função que apresenta mais de um extremo.

NOTA: toda função côncava ou convexa é unimodal, mas nem toda função unimodal é necessariamente côncava ou convexa. Veja exemplo $S(x) = (x^2 - 1)^3$.

2.5 Álgebra vetorial

Sejam x e $y \in X \subseteq \mathfrak{R}^n$, A e $B \in \mathfrak{R}^{n \times n}$, $S(x): X \rightarrow \mathfrak{R}$, $g(x)$ e $h(x): X \rightarrow X$. Então as seguintes regras de diferenciação se aplicam:

$$\nabla[g(x)^T h(x)] = \nabla^T g(x) h(x) + \nabla^T h(x) g(x)$$

$$\nabla[S(x) g(x)] = g(x) \nabla^T S(x) + S(x) \nabla g(x)$$

$$\nabla(A x) = A$$

$$\nabla(x^T A) = A$$

$$\nabla(x^T A x) = (A + A^T) x$$

$$\nabla S[g(x)] = \nabla^T g(x) \nabla_g S(g)$$

$$\nabla S\{g[h(x)]\} = \nabla^T h(x) \nabla_h^T g(h) \nabla_g S(g)$$

$$\nabla g[h(x)] = \nabla_h g(h) \nabla h(x)$$

$$\frac{d(x^T y)}{dt} = \dot{x}^T y + x^T \dot{y}$$

$$\frac{d(Ax)}{dt} = \dot{A}x + A\dot{x}$$

$$\frac{d(AB)}{dt} = \dot{A}B + A\dot{B}$$

$$\frac{d(x^T Ax)}{dt} = \dot{x}^T Ax + x^T \dot{A}x + x^T A\dot{x}$$

Exercícios de fixação

1. Teste as condições necessárias e suficientes do problema abaixo.

$$\min S(x) = x_1 x_2$$

$$\text{sujeito a: } g_1(x) = x_1^2 + x_2^2 - 25 \leq 0$$

2. Determine se as funções abaixo são côncavas, convexas, estritas ou não, ou nenhum destes casos ?

a) $S(x) = 2x_1 + 3x_2 + 6$

b) $S(x) = 2x_1^2 - 3x_1 x_2 + 2x_2^2$

c) $S(x) = x_1^3 + x_2^2 - 3x_1 + 8x_2 + 2$

3. Verifique se as restrições abaixo, definindo uma região fechada, formam uma região convexa.

$$g_1(x) = -x_1^2 + x_2 \geq 1 \quad \text{e} \quad g_2(x) = x_2 - x_1 \leq 2$$

NOTA: se todas as restrições colocadas na forma $g_i(x) \leq 0$ são funções convexas (ou na forma $g_i(x) \geq 0$ são funções côncavas), então elas formam uma região convexa. Funções lineares são tanto convexas quanto côncavas.

4. Mostre que a função $S(x) = -x_1 x_2$ com $x_1 \geq 0$ e $x_2 \geq 0$ é quasi-convexa. Ela é também convexa ? Por quê ?

5. Mostre que a função $S(x) = x_1^2 + 5x_1 x_2 + 3x_2^2$ com $x_1 > 0$ e $x_2 > 0$ é pseudo-convexa. Ela é também convexa ? Por quê ?

3. Otimização Sem Restrição

Na otimização sem restrição o problema que está sendo resolvido é:

$$\min_{x \in \mathfrak{R}^n} S(x) \quad \text{ou} \quad \max_{x \in \mathfrak{R}^n} S(x)$$

como $\max_{x \in \mathfrak{R}^n} S(x)$ é equivalente a $\min_{x \in \mathfrak{R}^n} -S(x)$, os métodos descritos a seguir são para problemas de minimização. Os métodos existentes para a solução deste problema podem ser agrupados em duas categorias:

- 1) métodos que não usam derivadas (métodos de busca, métodos diretos);
- 2) métodos que usam derivadas (métodos analíticos, métodos da métrica variável, métodos indiretos).

Como regra geral, na solução de problemas sem restrição, os métodos que usam derivadas convergem mais rapidamente que os métodos de busca. Por outro lado, os métodos de busca não requerem regularidade e continuidade da função objetivo e, principalmente o cálculo de derivadas primeira ou segunda de $S(x)$.

3.1 Métodos de busca monovariável e multivariável

Uma forma de avaliar os diferentes algoritmos de busca é utilizando os seguintes critérios:

- 1) Número de problemas resolvidos
- 2) Número de cálculos da função objetivo em cada exemplo
- 3) Tempo computacional para a solução de cada exemplo
- 4) Qualidade da solução

Pode-se então definir uma tabela de classificação, com índices de 0 (pior) a 100 (melhor) usando os seguintes índices:

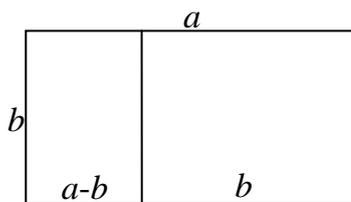
$$\theta_i = \frac{100}{N} \sum_{j=1}^N \frac{T_j^*}{T_{i,j}} \quad \chi_i = \frac{100}{N} \sum_{j=1}^N \frac{S_j^*}{S_{i,j}} \quad \eta_i = 100 \frac{N_i}{N} \quad \xi_i = \frac{100}{N} \sum_{j=1}^N \frac{(d_j^* + \varepsilon)}{(d_{i,j} + \varepsilon)}$$

onde, para o i -ésimo algoritmo, θ_i é a sua eficácia, χ_i é a sua eficiência, η_i é a sua robustez para resolver problemas, ξ_i é a qualidade da solução obtida, $T_{i,j}$ é o seu tempo computacional para resolver o problema j , T_j^* é o menor tempo entre os algoritmos para resolver o problema j , $S_{i,j}$ é o seu número de avaliações da função objetivo para

resolver o problema j , S_j^* é o menor número de avaliações entre os algoritmos para resolver o problema j , d_j^* é a melhor qualidade da solução para o problema j , $d_{i,j}$ é a qualidade da solução para o problema j definida como $d_{i,j} = \frac{\|x_{i,j} - x_j^*\|}{\varepsilon_x} + \frac{|S(x_{i,j}) - S(x_j^*)|}{\varepsilon_S}$, ε é a precisão da máquina, ε_x é a tolerância na variável independente, ε_S é a tolerância na função objetivo, x_j^* é a solução exata do problema j , N_i é o seu número de problemas resolvidos e N é o número total de problemas. Quando o i -ésimo algoritmo não consegue resolver o j -ésimo problema, então $T_{i,j} = \infty$ e $S_{i,j} = \infty$.

3.1.1 Método da seção áurea

É um método de busca monovariável, onde a cada iteração o intervalo de busca é reduzido por um fator α , chamado de razão áurea, obtido pela relação geométrica abaixo (retângulo áureo) :



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

$$\left(\frac{b}{a}\right)^2 + \frac{b}{a} - 1 = 0$$

$$\alpha = \frac{b}{a} = \frac{-1 + \sqrt{5}}{2} = 0,618$$

Outras escolhas do fator α levariam a métodos similares, como por exemplo o método da bisseção para $\alpha = 0,5$. Contudo, a vantagem da razão áurea está na redução do número de cálculos da função objetivo, em função da propriedade deste método de conservar o retângulo áureo a cada iteração. Outro método com características similares a seção áurea é a busca de Fibonacci.

algoritmo

1) Determinar o intervalo de busca $[L^0, U^0]$ que contém o ponto de mínimo

2) Fazer $k = 0$ e calcular $\Delta^0 = U^0 - L^0$,

$$x_L^0 = L^0 + \alpha \Delta^0, S_L^0 = S(x_L^0),$$

$$x_U^0 = U^0 - \alpha \Delta^0, S_U^0 = S(x_U^0),$$

3) **Se** $S(x_U^k) > S(x_L^k)$, **então** $L^{k+1} = x_U^k, x_U^{k+1} = x_L^k, S_U^{k+1} = S_L^k$

senão $U^{k+1} = x_L^k, x_L^{k+1} = x_U^k, S_L^{k+1} = S_U^k$

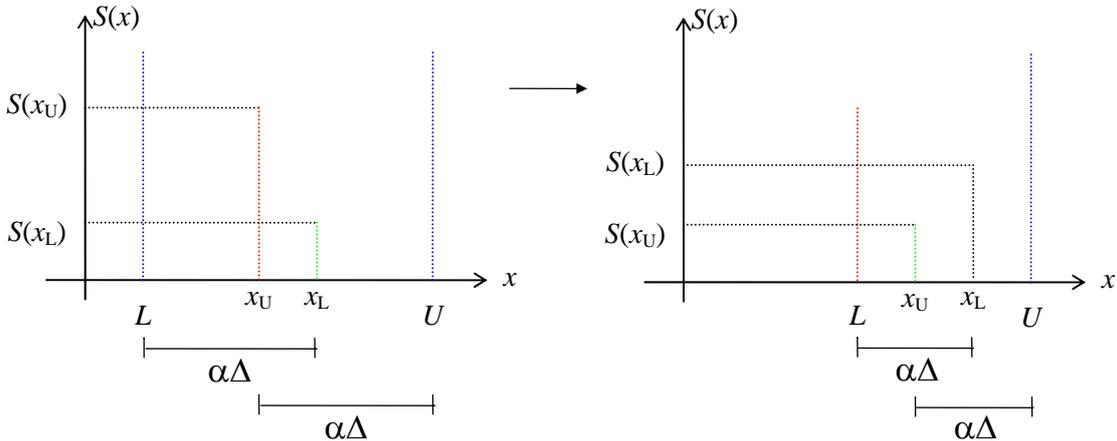
4) Fazer $(k \leftarrow k + 1)$ e calcular $\Delta^k = U^k - L^k$,

$$x_L^k = L^k + \alpha \Delta^k, \quad S_L^k = S(x_L^k) \quad \text{se } S(x_U^{k-1}) > S(x_L^{k-1})$$

ou
$$x_U^k = U^k - \alpha \Delta^k, \quad S_U^k = S(x_U^k) \quad \text{se } S(x_U^{k-1}) < S(x_L^{k-1})$$

5) Se $\Delta^k > \varepsilon$ (tolerância), **então** (ir para 3)

senão FIM.



3.1.2 Método da aproximação polinomial

Na verdade é uma classe de métodos de busca monovariável, que aproxima a função $S(x)$ por uma interpolação polinomial, $P_n(x)$:

$$P_n(x) = \sum_{j=1}^n \ell_j(x) S(x_j)$$

onde $\ell_j(x) = \prod_{\substack{k=1 \\ k \neq j}}^n \frac{(x - x_k)}{(x_j - x_k)}$ são os interpoladores de Lagrange. Quando a derivada de

$S(x)$ está disponível, então ela também é usada na aproximação polinomial:

$$P_{2n-1}(x) = \sum_{j=1}^n [h_1(x) S(x_j) + h_2(x) \nabla S(x_j)]$$

onde $h_1(x) = \ell_j^2(x) [1 - 2(x - x_j) \nabla \ell_j(x_j)]$ e $h_2(x) = \ell_j^2(x) (x - x_j)$ são os interpoladores de Hermite.

Interpolação quadrática ou método de Coggins (ou DSC-Powell)

O método de G.F. Coggins (1964) ou de Davies-Swann-Campey-Powell (1964) aproxima a função $S(x)$ por uma interpolação quadrática, $P_2(x)$:

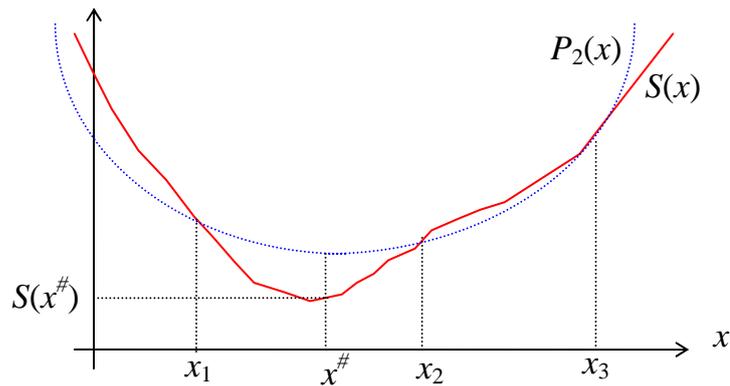
$$P_2(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} S(x_1) + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} S(x_2) + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} S(x_3)$$

calculando o mínimo desta função quadrática, isto é, $\frac{dP_2}{dx} = 0$, tem-se:

$$x^\# = \frac{1}{2} \frac{(x_2^2 - x_3^2)S(x_1) + (x_3^2 - x_1^2)S(x_2) + (x_1^2 - x_2^2)S(x_3)}{(x_2 - x_3)S(x_1) + (x_3 - x_1)S(x_2) + (x_1 - x_2)S(x_3)}$$

algoritmo

- 1) Determinar o intervalo de busca $[x_1, x_3]$
- 2) Calcular $S_1 = S(x_1)$ e $S_3 = S(x_3)$
- 3) Calcular $x_2 = 0,5 (x_1 + x_3)$ e $S_2 = S(x_2)$
- 4) Calcular $x^\# = \frac{1}{2} \frac{(x_2^2 - x_3^2)S_1 + (x_3^2 - x_1^2)S_2 + (x_1^2 - x_2^2)S_3}{(x_2 - x_3)S_1 + (x_3 - x_1)S_2 + (x_1 - x_2)S_3}$ e $S^\# = S(x^\#)$
- 5) **Se** $|x^\# - x_i| < \epsilon$ para algum $i = 1, 2, 3$, **então FIM.**
- 6) **Se** $(x_3 - x^\#)(x^\# - x_2) > 0$, **então** $k = 1$
 senão $k = 3$
- 7) **Se** $S^\# < S_2$, **então** $x_k \leftarrow x_2$, $S_k \leftarrow S_2$ e $k = 2$
- 8) $x_{4-k} \leftarrow x^\#$, $S_{4-k} \leftarrow S^\#$ e (ir para 4).



Interpolação cúbica

Para a aproximação cúbica são necessárias quatro informações sobre a função objetivo, ou seja, quatro valores da função ou dois valores da função e de sua derivada:

$$P_3(x) = \sum_{j=1}^4 \ell_j(x) S(x_j) \quad \text{ou} \quad P_3(x) = \sum_{j=1}^2 [h_1(x) S(x_j) + h_2(x) \nabla S(x_j)]$$

Escrevendo o polinômio na forma: $P_3(x) = a_3 x^3 + a_2 x^2 + a_1 x^1 + a_0$, a condição necessária de primeira ordem para o mínimo:

$$\frac{dP_3(x)}{dx} = 3 a_3 x^2 + 2 a_2 x + a_1 = 0$$

resulta no ponto $x^\# = \frac{-a_2 \pm \sqrt{a_2^2 - 3a_1 a_3}}{3a_3}$ cujo sinal usado é aquele que fornece um

valor positivo para a derivada segunda de $P_3(x)$, isto é, $6 a_3 x^\# + 2 a_2 > 0$. No caso de usar dois pontos tem-se as expressões:

$$x^\# = x_2 - \left[\frac{\nabla S(x_2) + w - z}{\nabla S(x_2) - \nabla S(x_1) + 2w} \right] (x_2 - x_1)$$

onde $z = \nabla S(x_1) + \nabla S(x_2) - 3 [S(x_1) - S(x_2)] / (x_1 - x_2)$ e

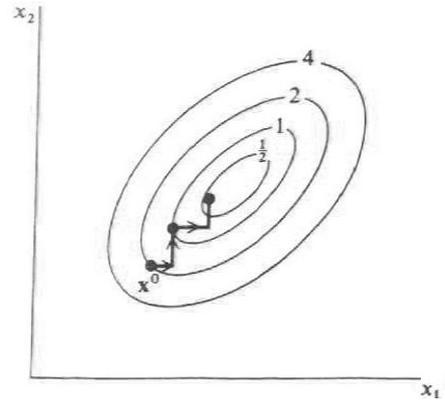
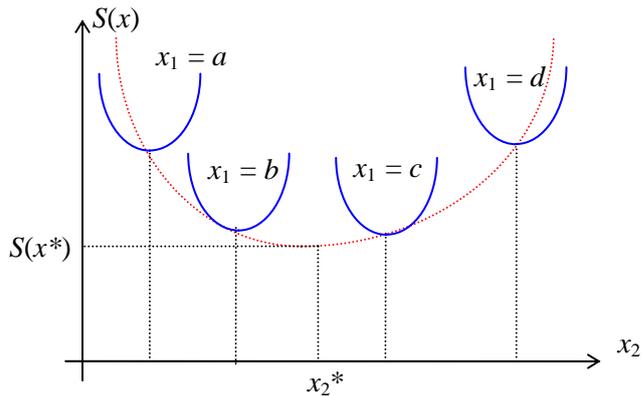
$$w = [z^2 - \nabla S(x_1) \nabla S(x_2)]^{1/2}$$

algoritmo

- 1) Determinar o intervalo de busca $[x_1, x_2]$
- 2) Calcular $S_1 = S(x_1)$, $\nabla S_1 = \nabla S(x_1)$ e $S_2 = S(x_2)$, $\nabla S_2 = \nabla S(x_2)$
- 3) Calcular $x^\#, S^\# = S(x^\#)$ e $\nabla S^\# = \nabla S(x^\#)$
- 4) **Se** $|x^\# - x_i| < \varepsilon$ para algum $i = 1, 2$, **então** FIM.
- 5) **Se** $\nabla S^\# \nabla S_1 > 0$, **então** $k = 1$
senão $k = 2$
- 6) $x_k \leftarrow x^\#, S_k \leftarrow S^\#, \nabla S_k \leftarrow \nabla S^\#$ e (ir para 3).

3.1.3 Método de busca seccionada

É um método de busca multivariável, mas em uma única direção por vez, que utiliza os métodos de busca univariável em cada direção. As figuras abaixo ilustram a idéia do método na busca de um ponto de mínimo.

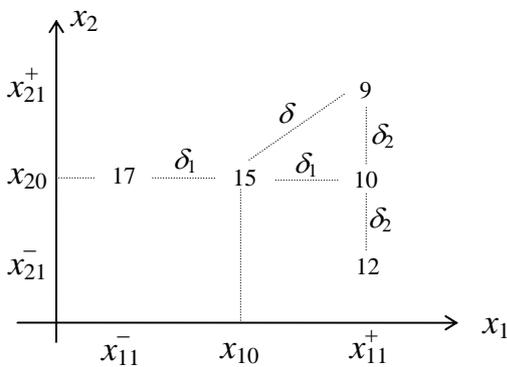


3.1.4 Método de Hooke & Jeeves

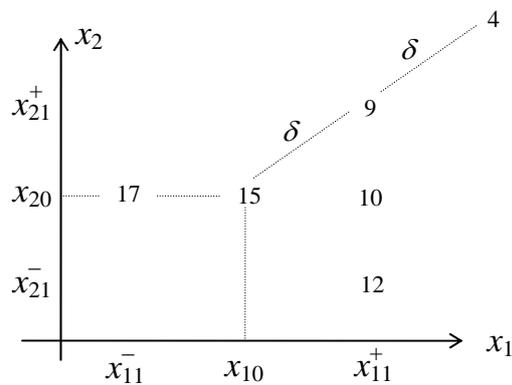
É um método de busca multivariável dividido em duas fases (R. Hooke e T.A. Jeeves, 1962):

fase de exploração: estimar a direção provável do extremo, a partir de um ponto inicial (ponto base).

fase de progressão: progredir na direção provável do extremo enquanto o valor da função objetivo for diminuindo.



exploração



progressão

algoritmo

Partida:

- 1) Determinar a região de busca $[L_i, U_i]$, $(i = 1, 2, \dots, n)$
- 2) Selecionar o ponto base inicial x_{i0} $(i = 1, 2, \dots, n)$
- 3) Calcular o valor $S_0 = S(x_0)$ da função objetivo em x_0
- 4) Selecionar os incrementos iniciais δ_i e as respectivas tolerâncias ε_i $(i = 1, 2, \dots, n)$
- 5) Tomar a primeira direção de busca $(k = 1)$

Fase de Exploração:

- 6) Calcular $x_{k0}^- = x_{k0} - \delta_k$ (sentido negativo)
- 7) **Se** x_{k0}^- estiver fora da região de busca, **então** insucesso em k^- (ir para 10)
- 8) Calcular o valor de $S_o^- = S(x_o^-)$
- 9) **Se** $S_o^- > S_o$ **então** insucesso em k^-
 senão sucesso em k^- e fazer $x_{k0} \leftarrow x_{k0}^-$ e $S_o \leftarrow S_o^-$ (ir para 14)
- 10) Calcular $x_{k0}^+ = x_{k0} + \delta_k$ (sentido positivo)
- 11) **Se** x_{k0}^+ estiver fora da região de busca, **então** insucesso em k^+ (ir para 14)
- 12) Calcular o valor de $S_o^+ = S(x_o^+)$
- 13) **Se** $S_o^+ > S_o$ **então** insucesso em k^+
 senão sucesso em k^+ e fazer $x_{k0} \leftarrow x_{k0}^+$ e $S_o \leftarrow S_o^+$
- 14) **Se** já foram exploradas todas as direções ($k = n$) **então** (ir para 15)
 senão tomar a direção seguinte ($k \leftarrow k + 1$) e (ir para 6)
- 15) **Se** houve sucesso em alguma direção **então** (ir para 17)
- 16) **Se** $\delta_i \leq \varepsilon_i \quad \forall i$ **então** FIM.
 senão $\delta_i \leftarrow \delta_i / 2 \quad \forall i$ tal que $\delta_i > \varepsilon_i$ (ir para 5)

Fase de Progressão:

- 17) Tomar $x_{i1} = x_{i0} \pm \delta_i$ (e, opcionalmente, $\delta_i \leftarrow 2\delta_i$) para todas as direções que houve sucesso
- 18) **Se** x_1 estiver fora da região de busca, **então** insucesso na progressão (ir para 16)
- 19) Calcular o valor de $S_1 = S(x_1)$
- 20) **Se** $S_1 > S_o$ **então** insucesso na progressão (ir para 5)
 senão sucesso na progressão e fazer $x_o \leftarrow x_1$ e $S_o \leftarrow S_1$ (ir para 17)

3.1.5 Método de Rosenbrock

É um método de busca multivariável parecido com a fase de exploração do método de Hooke & Jeeves (H.H. Rosembrock, 1960). Entretanto, ao invés de continuamente explorar nas direções dos eixos coordenados, são construídas novas direções ortogonais, usando o procedimento de Gram-Schmidt, baseadas nos tamanhos dos passos das direções bem sucedidas.

Sejam $d_i^k \in \mathcal{R}^n$, $i = 1, 2, \dots, n$, vetores direção unitários do k -ésimo estágio de busca e $\Delta_i^k = \sum \delta_i$ a soma de todos os passos bem sucedidos (distância percorrida) na i -ésima direção durante o k -ésimo estágio de busca. Ordenando os valores das

distâncias em ordem decrescente: $|\Delta_1^k| > |\Delta_2^k| > \dots > |\Delta_n^k|$, onde as últimas m direções (se existirem) possuem $\Delta_i^k = 0$, $i = (n - m) + 1, \dots, n$, então os novos vetores direção para o estágio de busca $k + 1$ são obtidos pelo seguinte processo de ortogonalização:

$$A_i^k = \sum_{j=i}^n \Delta_j^k d_j^k, \quad 1 \leq i \leq n$$

$$d_1^{k+1} = \frac{A_1^k}{\|A_1^k\|}$$

$$d_i^{k+1} = \frac{B_i^k}{\|B_i^k\|}, \quad B_i^k = A_i^k - \sum_{j=1}^{i-1} [(A_i^k)^T d_j^{k+1}] d_j^{k+1} \quad 1 < i \leq (n - m)$$

$$d_i^{k+1} = d_i^k \quad (n - m) < i \leq n$$

algoritmo

- 1) Selecionar o ponto base inicial x^0 , fazer $k = 0$, e $d_i^0 = e_i$ (direção dos eixos)
- 2) Selecionar os incrementos iniciais δ_i e as respectivas tolerâncias ε_i ($i = 1, 2, \dots, n$)
- 3) Calcular $S_0 = S(x^0)$, escolher a direção $m = 1$ e fazer $\Delta_i^0 = 0$ ($i = 1, 2, \dots, n$)
- 4) Fazer $x^1 = x^0 + \delta_m d_m^k$ e calcular $S_1 = S(x^1)$
- 5) **Se** $S_1 > S_0$ **então** insucesso e fazer $\delta_m \leftarrow -\delta_m / 2$
senão sucesso e fazer $x^0 \leftarrow x^1$, $S_0 \leftarrow S_1$, $\Delta_m^k \leftarrow \Delta_m^k + \delta_m$, $\delta_m \leftarrow 3 \delta_m$
- 7) Se alguma direção ainda não teve insucesso, fazer $m \leftarrow (m < n ? m+1 : 1)$
 (ir para 4)
- 7) **Se** $\delta_i \leq \varepsilon_i \quad \forall i$ **então** FIM.
- 8) Calcular a nova direção ortogonal e fazer $k \leftarrow k + 1$ (ir para 4).

3.1.6 Método de Powell

É um método de busca multivariável ao longo de direções conjugadas (C.S. Smith, 1962 e M.J.D. Powell, 1964). Duas direções d_1 e $d_2 \in \mathcal{R}^n$ são ditas conjugadas entre si, com respeito a uma matriz H , se:

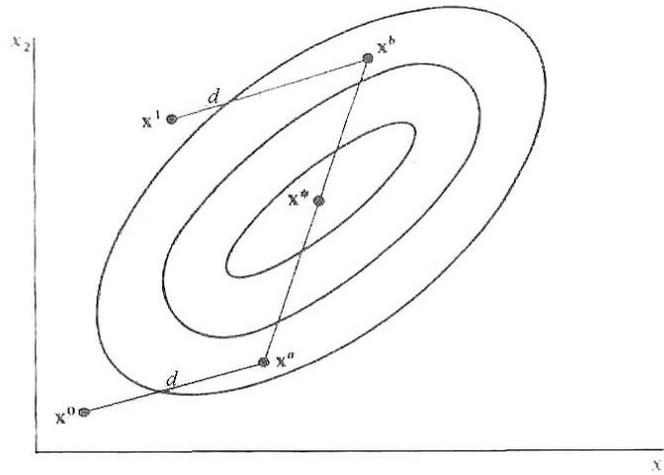
$$d_1^T H d_2 = 0$$

A ortogonalidade é um caso particular de conjugância, onde $H = I$. Em geral, um conjunto de n direções linearmente independentes $d_i \in \mathcal{R}^n$, $i = 1, 2, \dots, n$, são conjugados com respeito a uma matriz positiva definida $H \in \mathcal{R}^{n \times n}$ se:

$$d_i^T H d_j = 0 \quad 1 \leq i \neq j \leq n$$

Se H é a matriz Hessiana de uma função quadrática, $S(x)$, então o mínimo é alcançado após n estágios de buscas unidirecionais exatas. Se os eixos coordenados são transladados e rotacionados por transformações adequadas de modo que os eixos estejam nas direções dos vetores característicos de H e centralizados no ponto estacionário de $S(x)$, então a conjugância pode ser interpretada como ortogonalidade no espaço transformado.

Para construir uma direção conjugada sem o uso das derivadas de $S(x)$, pode-se partir de um ponto x^o e obter o ponto x^a que minimiza $S(x)$ ao longo de uma direção d , como ilustra a figura abaixo. Então, partindo de outro ponto $x^1 \neq x^o$ e obtendo o ponto x^b que minimiza $S(x)$ ao longo da mesma direção d , o vetor $(x^b - x^a)$ é conjugado ao vetor d , com respeito a matriz Hessiana, H , se $S(x)$ é uma função quadrática (para funções não lineares em geral, este conceito é uma aproximação).



Para verificar esta conjugância, basta representar $S(x)$ em série de Taylor em torno do ponto x^o :

$$S(x) = S(x^o) + \nabla^T S(x^o)(x - x^o) + \frac{1}{2}(x - x^o)^T H(x - x^o)$$

e minimizá-la ao longo da direção d : $x = x^o + \alpha d$, isto é, obter α tal que:

$$\frac{dS(x^o + \alpha d)}{d\alpha} = 0 = \nabla^T S(x^o) d + d^T H \alpha d = d^T \nabla S(x^o) + d^T H \alpha d$$

resultando em: $\alpha = -\frac{\nabla^T S(x^o) d}{d^T H d}$

Rescrevendo a função $S(x)$ na forma: $S(x) = a + b^T x + \frac{1}{2} x^T H x$, tem-se:

$$\nabla S(x) = b + H x \quad \text{e} \quad \nabla S(x^o) = b + H x^o$$

que substituído na equação $dS/d\alpha = 0$, sabendo que $\alpha d = x^a - x^o$, resulta em:

$$d^T (b + H x^o) + d^T H (x^a - x^o) = 0 \rightarrow d^T (b + H x^a) = 0$$

e portanto: $d^T \nabla S(x^a) = 0$, isto é, o gradiente da função objetivo no mínimo da direção de busca é ortogonal a esta direção. Fazendo a mesma análise para o ponto x^1 , tem-se: $d^T \nabla S(x^b) = d^T (b + H x^b) = 0$. Subtraindo estes dois resultados chega-se a:

$$d^T H (x^b - x^a) = 0$$

que mostra que d é conjugado a direção $(x^b - x^a)$ com respeito a matriz H .

O método de Powell é uma implementação da busca em direções conjugadas, que pode ser sumarizado nos seguintes passos:

algoritmo

- 1) Selecionar um ponto inicial x_0^0 , fazer estágio $k = 0$ e $d_i^0 = e_i$ (direção dos eixos)
- 2) A partir do ponto x_0^k determine α_1 via busca unidimensional na direção de d_1^k e $x_1^k = x_0^k + \alpha_1 d_1^k$. A partir do ponto x_1^k , determinar α_2 via busca unidimensional na direção de d_2^k e fazer $x_2^k = x_1^k + \alpha_2 d_2^k$, e assim sucessivamente até determinar todos os $\alpha_i, i = 1, 2, \dots, n$. A transição do ponto x_0^k para o ponto x_m^k é equivalente a:

$$x_m^k = x_0^k + \sum_{i=1}^m \alpha_i d_i^k \quad , \quad m = 1, 2, \dots, n$$

- 3) Calcular $x_{n+1}^k = 2x_n^k - x_0^k$, isto é, fazer um passo adicional de tamanho $(x_n^k - x_0^k)$ na direção resultante das n buscas unidimensionais. Este passo procura garantir que duas direções de busca não se tornem colineares quando algum $\alpha_i = 0$.
- 4) Calcular $\Delta^k = \max_{i=1, \dots, n} [S(x_{i-1}^k) - S(x_i^k)]$. A direção de busca relativa a esta variação máxima será designada por d_m^k . Definindo $S_1 = S(x_0^k)$, $S_2 = S(x_n^k)$ e $S_3 = S(x_{n+1}^k)$, **Se** $S_3 \geq S_1$ e/ou $(S_1 - 2S_2 + S_3) (S_1 - S_2 - \Delta^k)^2 \geq 1/2 \Delta^k (S_1 - S_3)^2$, **então** usar as mesmas direções do estágio k , isto é, $d_i^{k+1} = d_i^k, i = 1, 2, \dots, n$ e iniciar o próximo estágio a partir do ponto $x_0^{k+1} = x_n^k$ (ou x_{n+1}^k se $S_2 > S_3$).
- 5) **Senão** usar as direções $[d_1^{k+1} \ d_2^{k+1} \ \dots \ d_n^{k+1}] = [d_1^k \ d_2^k \ \dots \ d_{m-1}^k \ d_{m+1}^k \ \dots \ d_n^k \ d^k]$, onde a direção d_m^k (de máxima variação) é eliminada e a nova direção resultante, d^k , (de x_0^k para x_n^k) é incluída em último lugar, com o correspondente valor de α obtido via minimização nesta direção e $x_0^{k+1} = x_n^k + \alpha d^k$.

6) Se $\|x_n^k - x_0^k\| \leq \varepsilon$ então FIM

7) Fazer $k \leftarrow k + 1$ (ir para 2).

3.1.7 Método de busca de limites

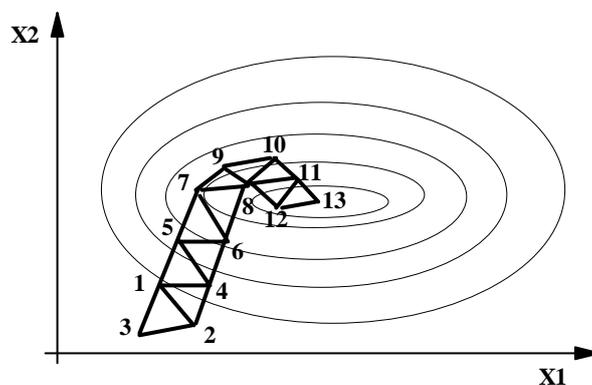
A maioria dos métodos de busca necessitam da definição do intervalo de busca, que contém o ponto extremo, para garantirem a determinação do ótimo. Segue abaixo um algoritmo de busca para a determinação de um intervalo $[x_0, x_1]$ que contém um ponto de mínimo.

algoritmo

- 1) Escolher um ponto inicial, x_0 , e um passo inicial, α
- 2) Calcular $S_0 = S(x_0)$, $k = 0$
- 3) Calcular $x_1 = x_0 + \alpha$ e $S_1 = S(x_1)$
- 4) Se $S_0 \leq S_1$, então $\alpha \leftarrow -\alpha$, $k \leftarrow k + 1$ e (ir para 7)
- 5) $S_0 \leftarrow S_1$, $\alpha \leftarrow 2\alpha$, $x_1 = x_0 + \alpha$ e $S_1 = S(x_1)$
- 6) Se $S_0 > S_1$, então (ir para 5)
- senão $k = 1$ (ir para 8)
- 7) Se $k < 2$, então (ir para 3)
- 8) $x_0 \leftarrow x_0 + \alpha(k - 1)$, $I = [x_0, x_1]$, FIM.

3.1.8 Método dos poliedros flexíveis

É um método de busca multivariável (J.A. Nelder e R. Mead, 1964), onde o pior vértice de um poliedro com $n + 1$ vértices é substituído por um novo vértice colinear com o vértice antigo e o centróide, como ilustra a figura abaixo.



As coordenadas do centróide são dadas por:

$$x_{0,j} = \frac{1}{n} \left[\sum_{i=1}^{n+1} x_{i,j} - x_{h,j} \right] \quad j = 1, 2, \dots, n$$

onde $x_{h,j}$ é o pior vértice.

O algoritmo envolve quatro operações de busca, que para o caso da minimização da função objetivo têm as seguintes formas:

$$1) \text{ Reflexão: } \begin{cases} x_R^k = x_0^k + \alpha (x_0^k - x_h^k) \quad , \quad \alpha > 0 \\ \text{onde } S(x_h^k) = \max \{ S(x_1^k), \dots, S(x_{n+1}^k) \} \end{cases}$$

$$2) \text{ Expansão: } \begin{cases} \text{Se } S(x_R^k) \leq S(x_\ell^k) = \min \{ S(x_1^k), \dots, S(x_{n+1}^k) \}, \\ \text{então } x_E^k = x_0^k + \gamma (x_R^k - x_0^k) \quad , \quad \gamma > 1 \\ \text{Se } S(x_E^k) < S(x_R^k), \text{ então } x_h^{k+1} = x_E^k \\ \text{senão } x_h^{k+1} = x_R^k \\ k = k + 1 \quad (\text{ir para 1}) \end{cases}$$

onde x_ℓ^k é o melhor vértice.

$$3) \text{ Contração: } \begin{cases} \text{Se } S(x_R^k) > S(x_i^k) \quad \forall \quad i \neq h, \text{ então } x_C^k = x_0^k + \beta (x_h^k - x_0^k) \\ x_h^{k+1} = x_C^k, \quad 0 < \beta < 1 \\ k = k + 1 \quad (\text{ir para 1}) \end{cases}$$

$$4) \text{ Redução: } \begin{cases} \text{Se } S(x_R^k) > S(x_h^k), \text{ então } x_i^{k+1} = x_\ell^k + \frac{1}{2} (x_i^k - x_\ell^k) \\ i = 1, 2, \dots, n + 1 \\ k = k + 1 \quad (\text{ir para 1}) \end{cases}$$

O critério usado por Nelder e Mead para terminar a busca é o seguinte:

$$\left\{ \frac{1}{n+1} \sum_{i=1}^{n+1} [S(x_i^k) - S(x_0^k)]^2 \right\}^{\frac{1}{2}} \leq \varepsilon.$$

3.1.9 Métodos de busca aleatória

São os métodos de busca menos elegantes e eficientes (menores valores para χ) de todas as técnicas de busca, mas são mais robustos para problemas multimodais e atrativos para computação paralela (multiprocessamento).

Método complex

É similar ao método dos poliedros flexíveis, sem a restrição de usar somente $n+1$ vértices (M. J. Box, 1965). A partir de um ponto inicial, x_1 , outros $p-1$ pontos são obtidos aleatoriamente:

$$x_i = L + \psi_i [U - L] \quad , \quad i = 2, 3, \dots, p$$

onde L e U são vetores que limitam a região de busca e ψ_i é uma matriz diagonal de números aleatórios, distribuídos uniformemente no intervalo $[0,1]$. A busca procede de forma análoga ao método dos poliedros flexíveis. Recomenda-se, em geral, usar $p = 2n$.

Busca aleatória repetitiva

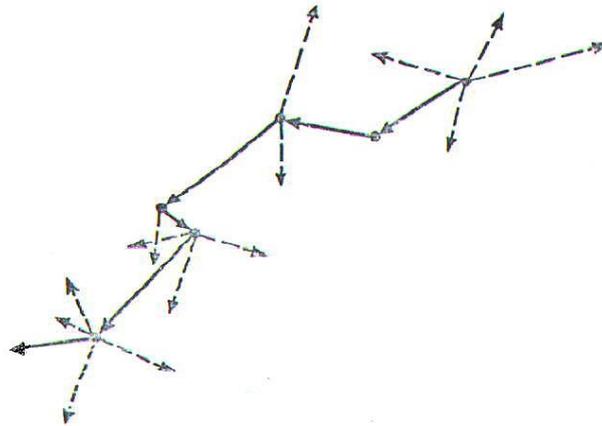
É uma busca completamente aleatória (R.J. Kelly e R.F. Wheeling, 1962), onde um caminho aleatório é construído na seqüência dos passos em busca do ótimo, através da seguinte relação:

$$x^{k+1} = x^k + \alpha_k \left[(1 - \beta) \frac{d^k}{\|d^k\|} + \beta r^k \right] \quad , \quad k = 1, 2, \dots$$

onde $\alpha_k \in \mathfrak{R}$ é o passo, que é ampliado após uma direção bem sucedida e reduzido caso contrário, β é um coeficiente ajustável durante a busca (grau de aleatoriedade das direções), $r^k \in \mathfrak{R}^n$ é um vetor unitário gerado aleatoriamente, $d^k \in \mathfrak{R}^n$ é um vetor da história da busca (uma direção média dos passos anteriores), obtido por:

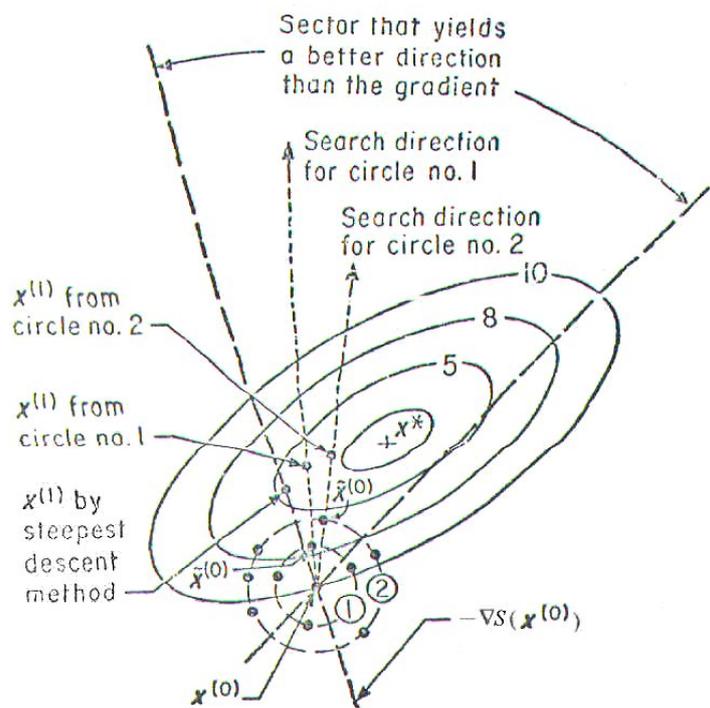
$$d^{k+1} = \gamma d^k + (1 - \gamma) (x^{k+1} - x^k) \Lambda^k$$

onde $\gamma \in [0, 1]$ é o fator de esquecimento das direções e $\Lambda^k \in \mathfrak{R}^{n+n}$ é uma matriz diagonal de fatores de escala para as variáveis x . A figura abaixo ilustra um caminho típico de uma busca aleatória.



Direções aleatórias

É uma técnica onde as direções de busca são aleatórias, geradas a partir de um raio de busca uniforme, isto é, os pontos aleatórios em cada estágio estão localizados sobre a superfície de uma hiperesfera com centro no ponto x^k . Uma busca em linha é realizada na direção que passa pelo ponto x^k e pelo melhor ponto da hiperesfera. A figura abaixo mostra que quanto maior o raio da hiperesfera (limitado pela posição do ponto ótimo), maior é a probabilidade do melhor ponto cair dentro do cone de direções melhores que a direção do gradiente no ponto x^k . Para atender os critérios de convergência, o raio de busca deve ser periodicamente reduzido. A busca pode ser acelerada se os pontos aleatórios forem restritos a um ângulo mínimo entre eles e a direção do estágio anterior.



Busca aleatória adaptativa

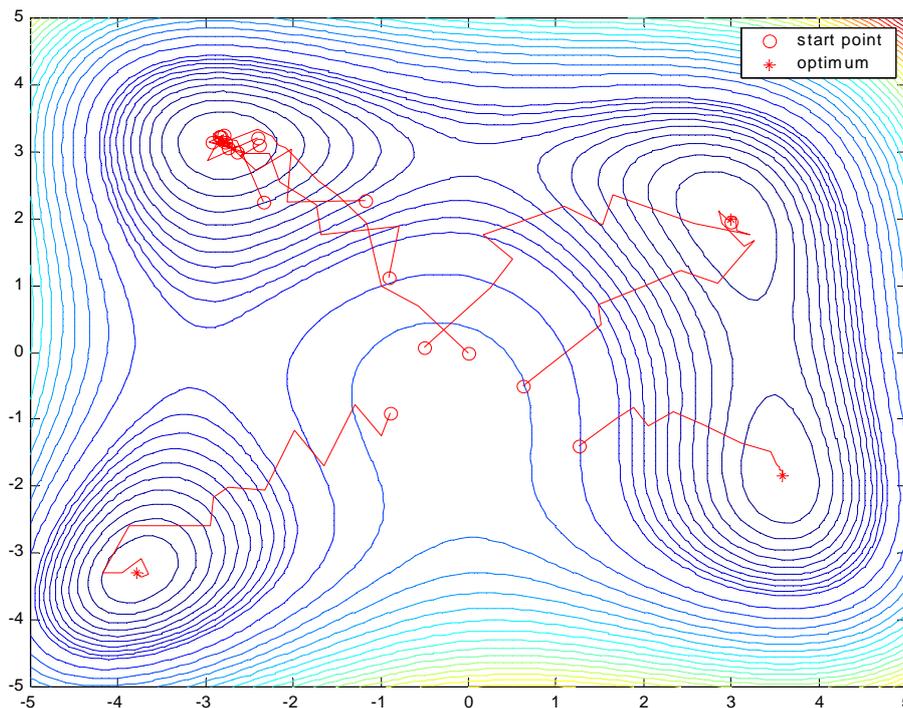
É método de otimização “global”, baseado em uma busca aleatória memorizada, com distribuição assimétrica e redução sistemática da hiperelipse de amostragem (A.R. Secchi e C.A. Perlingeiro, 1989), onde os pontos são selecionados da seguinte forma:

$$x_i^{k+1} = x_i^k + \frac{R_i}{\delta} [B_i (A_i \psi_i - 1)]^\delta, \quad i = 1, 2, \dots, n, \quad k = 1, 2, \dots$$

onde $x^k \in \mathfrak{R}^n$ é o melhor ponto da última amostragem, ψ é um vetor de números aleatórios $\in (0, 1)$, R é o vetor dos eixos da hiperelipse, δ é o parâmetro de forma da distribuição das amostras ($\delta = 1, 3, 5, \dots$, sendo que para $\delta = 1$ tem-se uma distribuição uniforme), A é o vetor de assimetria da distribuição ($A_i = 2 \rightarrow$ distribuição simétrica), B é o vetor de direção da assimetria ($B_i = -1 \rightarrow$ assimetria positiva, no sentido de aumento de x_i , e $B_i = 1 \rightarrow$ assimetria negativa). Após a amostragem, se estiver na etapa de exploração, um número determinado de pontos são armazenados para uso posterior, quando a busca na direção do melhor ponto for encerrada. As etapas principais do algoritmo de busca são as seguintes:

- 1) Amostragem de pontos aleatórios em torno de x^k , usando a equação acima;
- 2) Ordenação das amostras baseado no valor da função objetivo;
- 3) Memorização das amostras, isto é, armazenar os melhores pontos, exceto o melhor de todos, durante a fase de exploração ($\delta = 1$);
- 4) Análise do melhor ponto: verificar os critérios de convergência;
- 5) Checar validade de um novo ponto ótimo encontrado: análise de redundância de pontos, baseado em uma métrica;
- 6) Resgate de um ponto da memória;
- 7) Eliminação pelos ótimos: verificar a tendência da busca;
- 8) Ajustar direção e critérios de amostragem: alteração dos parâmetros de busca (A, B, δ);
- 9) Análise dos critérios de parada: número máximo de avaliações e ótimos;
- 10) Ordenação dos ótimos.

A figura abaixo ilustra a característica de memorização de pontos amostrados, que são utilizados como novos pontos de partida quando a busca em uma determinada direção é terminada. O encerramento da busca em uma direção pode ocorrer devido a localização de um ponto ótimo local ou pela análise de redundância dos pontos.



Algoritmo genético

A idéia dos algoritmos genéticos (GA), J. Holland (1975), está baseada no processo evolutivo dos organismos biológicos da natureza (princípios da seleção natural e da sobrevivência do mais adaptado). Os indivíduos que têm mais sucesso para se adaptar ao meio ambiente terão mais chance de sobreviver e reproduzir, enquanto que os menos ajustados serão eliminados. Isto significa que os genes dos indivíduos mais adaptados serão disseminados em um número crescente de indivíduos nas próximas gerações. A combinação de características de ancestrais bem ajustados pode produzir indivíduos que são bem mais adaptados que os pais. Os GAs procuram simular este processo natural pela aplicação de *operadores genéticos* sobre uma população inicial e suas próximas gerações. Cada indivíduo da população é codificado em um *cromossomo (string)*, que representa uma solução possível para o problema. A adaptação de um indivíduo é avaliada pelo valor da função objetivo. Indivíduos altamente ajustado são permitidos a se reproduzirem pela troca de parte de sua informação genética (em um procedimento de cruzamento) com outro indivíduo altamente ajustado, produzindo filhos (*offsprings*) com características de ambos os pais. Mutações são freqüentemente permitidas pela alteração de alguns genes dos cromossomos. Os filhos podem substituir toda a geração anterior (*técnica das gerações*), ou substituir apenas os indivíduos menos ajustados (*técnica do estado estacionário*). Os passos básicos de um algoritmo genético são os seguintes:

- 1) Gerar aleatoriamente uma população inicial e avaliar a adaptação de cada indivíduo;
- 2) Selecionar os pais de uma população, cruzá-los para produzirem os filhos e avaliar a adaptação dos filhos;
- 3) Substituir alguns ou todos os indivíduos da população pelos filhos;
- 4) Se o critério de convergência não foi satisfeito ir para o passo (2);
- 5) FIM.

O procedimento de cruzamento (operação genética) consiste em gerar aleatoriamente um ou mais pontos de cruzamento e, então trocar os segmentos dos cromossomos dos dois pais para produzir dois filhos.

A mutação é aplicada a um filho após um certo número de cruzamentos, pela substituição de um gene do cromossomo por um valor aleatório (ou inversão de bit no caso de cromossomos binários). A mutação ajuda resguardar a possível perda de informação genética valiosa devido a uma convergência prematura, expandindo o espaço de busca.

Simulated annealing

É uma técnica de busca direta com relaxação estocástica (S. Kirkpatrick, C. Gelatt Jr. e M. Vecchi, 1983), baseada no processo de recozimento de sólidos (onde um sólido é aquecido a altas temperaturas e gradualmente resfriado para permitir a sua cristalização). Como o processo de aquecimento permite que os átomos se movimentem aleatoriamente, se o resfriamento não é realizado muito rapidamente, então os átomos terão tempo suficiente para se alinharem de modo a atingir um estado de mínima energia. Usando esta analogia, o estado do sólido corresponde a uma solução viável e a energia de cada estado corresponde ao valor da função objetivo. Neste procedimento, a busca é permitida a prosseguir mesmo se o movimento causar um aumento na função objetivo, através do uso de um parâmetro chamado “*temperatura*” (T), onde o movimento é aceito com uma probabilidade de $\exp(-\Delta/T)$, com $\Delta = S(x^{k+1}) - S(x^k)$. O valor de $T \in (0, \infty)$ é controlado pelo processo de “*resfriamento*”, sendo que valores baixos resultam em um algoritmo de busca local, ao passo que valores altos reduzem a velocidade de convergência. Os passos básicos de um algoritmo *simulated annealing* (SA) são os seguintes:

- 1) Escolher um ponto inicial, $x_0 \in \mathcal{R}^n$, e os parâmetros α e β
- 2) Calcular $S_0 = S(x^0)$, $T = T_0$, $T_f = \alpha T_0$, $N = \beta n \lambda$, $r = 0$, $c = 0$, $k = 0$

3) Fazer N vezes:

Mudar de direção λ componentes de x^k e gerar aleatoriamente um vizinho, x

Calcular $\Delta = S(x) - S(x^k)$

Se $\Delta < 0$ ou $\exp(-\Delta/T_k) > Y$, **então** $x^k \leftarrow x$ e $c = 0$, **senão** $r \leftarrow r + 1$

4) **Se** $r = N$, **então** $c \leftarrow c + 1$

5) Atualizar T_k , $r = 0$, $k \leftarrow k + 1$

6) **Se** o processo não estiver “congelado” ($T > T_f$ ou $c \leq C_{max}$), **então** ir para (3)

7) FIM.

onde α é um parâmetro que define o nível de *temperatura* (T_f) que o algoritmo deve terminar, β é um parâmetro utilizado para definir o número de iterações (N) em cada nível de *temperatura*, T_o é a *temperatura* inicial, r é o número de rejeições, λ é número de mudanças de direção, $Y \in [0,1]$ é uma variável aleatória uniforme e C_{max} é o número de máximo de níveis consecutivos de temperatura sem progressão (isto é, rejeição total). O cálculo de T_o e a atualização de T_k podem ser feitos de diversas maneiras, tal como (Aarts et al., 1988):

$$T_o = \hat{\Delta} \left(\ln \frac{m_2}{m_2 \eta - (1 - \eta) m_1} \right)^{-1} \quad \text{e} \quad T_{k+1} = \frac{T_k}{1 + \frac{T_k \ln(1 + \delta)}{3\sigma_k}}$$

onde após um número m de tentativas iniciais, m_1 e m_2 são os números de movimentos com $\Delta \leq 0$ e $\Delta > 0$, respectivamente, $\hat{\Delta}$ é o valor médio dos $\Delta > 0$, $\eta \in (0, 1)$ é a taxa de aceitação inicial ($\approx 0,95$), σ_k é o desvio padrão da função objetivo durante a busca em T_k e δ é a máxima mudança relativa permitida para a probabilidade de aceitação dos estados ($\approx 0,1$).

PSO

O *Particle Swarm Optimization* (PSO) é um algoritmo que tem como fundamento o comportamento de organismos sociais tais como uma revoada de pássaros ou um cardumes de peixes, onde cada indivíduo da população (partícula) modifica sua posição com o tempo (geração). A posição é modificada de acordo com a experiência do indivíduo e a dos demais componentes da população, valendo-se de sua melhor posição e a melhor posição do conjunto.

Sistemas Fórmicos

Um Sistema Fórmico (ou *Ant System*) é um algoritmo baseado em agentes que emulam o comportamento natural de formigas e desenvolvem mecanismos de cooperação e aprendizado. O *Ant System* foi proposto como uma nova heurística para otimização combinatorial. Ele é composto de um conjunto de formigas artificiais que cooperam entre si para resolver um problema através de troca de informações via feromônio que é depositado nas arestas de um grafo. As formigas passeiam pelo grafo construindo uma solução para o problema. Após cada solução ser gerada, as formigas atualizam a intensidade de feromônio nas arestas que fazem parte de sua solução como uma função da qualidade da solução encontrada por aquelas formigas. Além disso, parte do feromônio evapora a cada passo, diminuindo a intensidade de trilhas que não são seguidas. O feromônio depositado nas arestas desempenha o papel de memória distribuída através da qual as formigas reforçam as soluções progressivamente melhores. Ao mesmo tempo, a escolha probabilística de cada passo da solução evita a conversão prematura para um ótimo local.

Busca tabu

É um procedimento para levar métodos de busca local para fora da região de atração de um ótimo local (meta-estratégia ou meta-heurísticas), pela incorporação de funções de memória flexível para proibir movimentos que levam a características (ou *atributos*, ou estado das variáveis) de soluções passadas (F. Glover, 1986). Os atributos que não são permitidos a serem retomados são chamados de *tabu* e são mantidos em uma memória de curta duração (*lista de tabus*).

Para maiores detalhes deste procedimento e dos dois métodos anteriores, sugere-se a leitura de Hasan et al. (2000) e suas referências.

3.2 Métodos analíticos (métrica variável)

Apesar da literatura referenciar somente os métodos quasi-Newton, que utilizam aproximações para o cálculo da matriz Hessiana (também conhecidos como métodos da secante), como métodos da métrica variável, nestas notas o termo “métrica variável” engloba todos os métodos que utilizam a primeira e/ou a segunda derivada da função objetivo. Estes métodos têm como equação básica para o processo iterativo:

$$x^{k+1} = x^k - \alpha_k W(x^k) \nabla S(x^k) \tag{3.1}$$

onde α_k é o tamanho do passo, $d^k = -W(x^k) \nabla S(x^k)$ é o vetor direção e

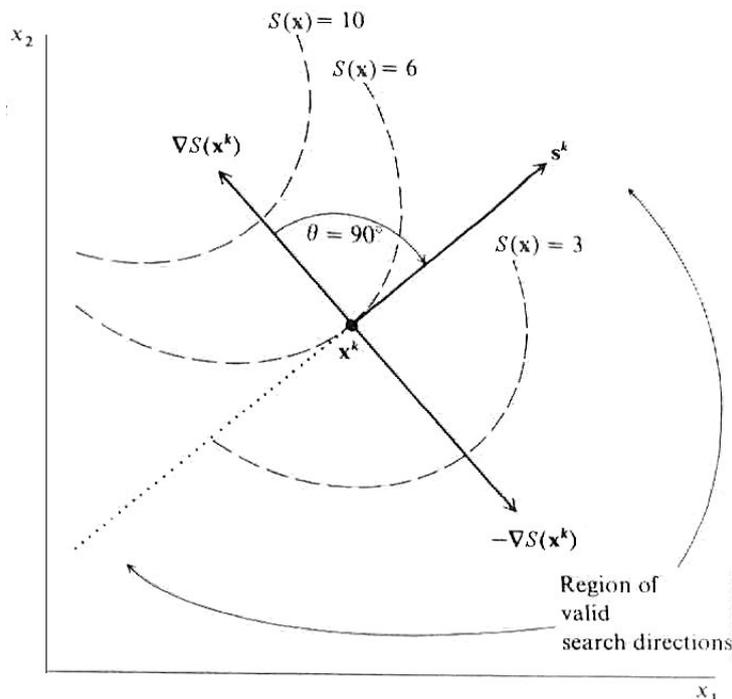
$W(x^k)$ é a matriz direção (inversa da matriz Hessiana ou aproximação desta)

Em qualquer método de otimização, uma boa direção de busca deve reduzir (para o caso da minimização) o valor da função objetivo, isto é, $S(x^{k+1}) < S(x^k)$. Tal direção, d^k , satisfaz o seguinte critério em cada ponto:

$$\nabla^T S(x^k) d^k < 0$$

ou em outras palavras, o ângulo (θ) formado entre os vetores $\nabla S(x^k)$ e d^k deve ser sempre maior que 90° , ou seja:

$$\nabla^T S(x^k) d^k = |\nabla^T S(x^k)| |d^k| \cos \theta < 0 \Leftrightarrow \theta > 90^\circ$$



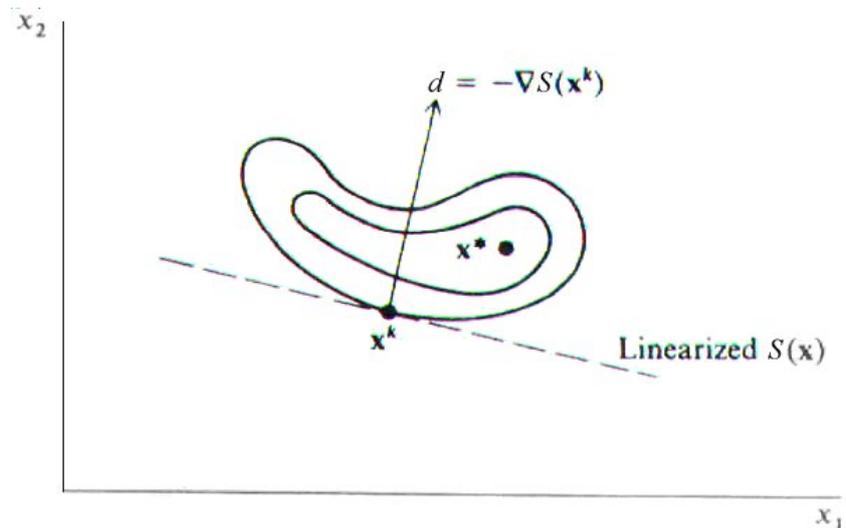
Como a otimização sem restrições é equivalente a encontrar a solução do sistema de equações não-lineares $F(x) = \nabla S(x) = 0$, pode-se utilizar todos os métodos disponíveis para a solução de $F(x) = 0$. Por exemplo, na utilização do método de Newton-Raphson, a matriz Jacobiana é a própria matriz Hessiana.

A seguir serão descritos os métodos mais conhecidos desta classe, seguidos de uma generalização dos métodos da métrica variável.

3.2.1 Métodos gradientes

Utilizam somente a primeira derivada da função objetivo, caso em que $W(x^k) = I$:

$$x^{k+1} = x^k - \alpha_k \nabla S(x^k)$$



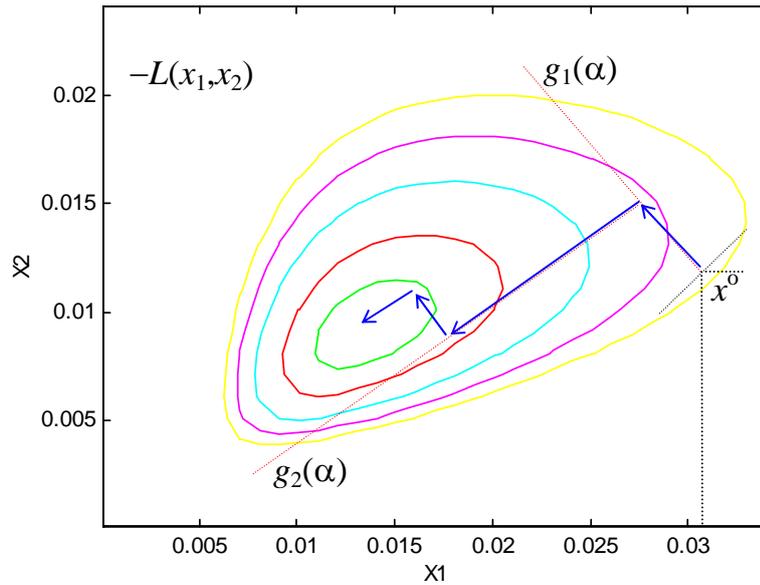
Quando α_k é escolhido de modo a minimizar:

$$g_k(\alpha) = S(x^k - \alpha \nabla S(x^k)) \quad , \quad \alpha > 0 \quad (3.2)$$

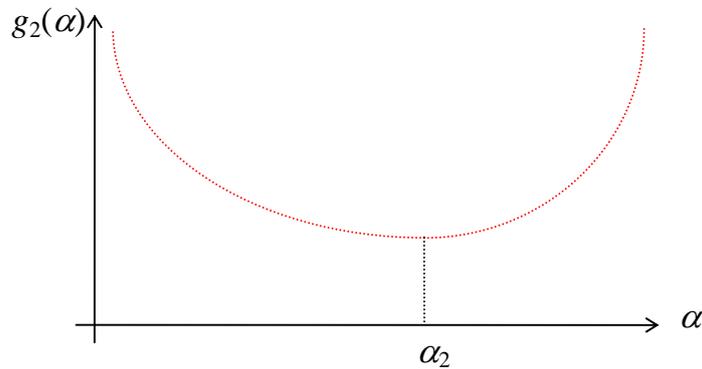
tem-se o método da maior descida (“steepest descent”), cujo algoritmo básico pode ser escrito da seguinte forma.

algoritmo

- 1) Escolher um ponto inicial x^0 , $k = 0$
- 2) Calcular $d^k = -\nabla S(x^k)$
- 3) Encontrar α_k tal que $S(x^k + \alpha_k d^k) = \min_{\alpha > 0} g_k(\alpha) = S(x^k + \alpha d^k)$
- 4) Calcular $x^{k+1} = x^k + \alpha_k d^k$
- 5) Se o critério de convergência não foi satisfeito, **então** $k \leftarrow k + 1$ (ir para 2)
- 6) FIM.



A minimização de $g_k(\alpha)$, conhecida como *função de mérito*, também chamada de busca em linha (“linesearch”), pode ser realizada com o uso de qualquer método de minimização univariável. Para ilustrar esta função, a figura abaixo mostra a função $g_2(\alpha)$ do problema acima:



Aproximando $S(x)$ por uma função quadrática:

$$S(x^{k+1}) \approx S(x^k) + \nabla^T S(x^k)(x^{k+1} - x^k) + \frac{1}{2}(x^{k+1} - x^k)^T H(x^k)(x^{k+1} - x^k)$$

ou de forma similar:

$$g_k(\alpha) = S(x^k + \alpha d^k) \approx S(x^k) + \alpha \nabla^T S(x^k) d^k + \frac{1}{2} \alpha^2 (d^k)^T H(x^k) d^k$$

que minimizando em relação a α , $\frac{dg_k}{d\alpha} = 0$, resulta:

$$\alpha^* = \alpha_k = -\frac{\nabla^T S(x^k) d^k}{(d^k)^T H(x^k) d^k} = \frac{(d^k)^T d^k}{(d^k)^T H(x^k) d^k} \quad (3.3)$$

Contudo, a equação (3.3) não é utilizada para o cálculo de α nos métodos gradientes, pois exigiria o cálculo da segunda derivada da função objetivo. Neste caso, utiliza-se, em geral, métodos de busca para a sua seleção.

3.2.2 Método de Newton

Faz uso da segunda derivada da função objetivo, caso em que $W(x^k) = [H(x^k)]^{-1}$:

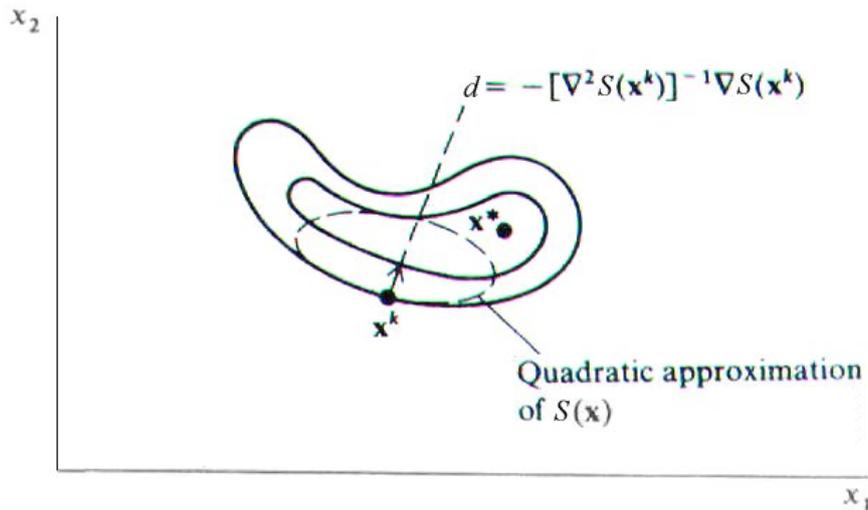
$$x^{k+1} = x^k - \alpha_k [H(x^k)]^{-1} \nabla S(x^k)$$

que é resultado da minimização da aproximação de $S(x)$ por uma função quadrática:

$$S(x) \approx S(x^k) + \nabla^T S(x^k) \Delta x^k + \frac{1}{2} (\Delta x^k)^T H(x^k) \Delta x^k \quad (3.4)$$

onde $\Delta x^k = x - x^k$, na direção Δx^k , isto é: $\frac{\partial S}{\partial \Delta x_i^k} = 0$

$$\Delta x^k = -[H(x^k)]^{-1} \nabla S(x^k)$$



Neste caso α_k ou é um parâmetro de relaxação do processo iterativo $0 < \alpha_k \leq 1$, ou é um fator de correção da inversa da matriz Hessiana, caso esta não seja atualizada em todas as iterações. A positividade da matriz Hessiana deve estar sempre garantida para evitar a migração para um ponto sela. E, para assegurar a convergência do método de Newton, a correção Δx^k deve ser tal que $S(x^{k+1}) < S(x^k)$.

Uma maneira de assegurar a positividade da matriz Hessiana é através da modificação de Levenberg-Marquardt, que adiciona um fator ajustável na diagonal da matriz Hessiana, ou em sua inversa:

$$\tilde{H}(x^k) = H(x^k) + \beta_k \mathbf{I} \quad , \quad \beta_k > -\min\{\lambda_i\}$$

$$W(x^k) = [H(x^k)]^{-1} + \gamma_k I \quad , \quad \gamma_k > -\min\{1/\lambda_i\}$$

onde λ_i são os valores característicos de $H(x^k)$.

Em particular, quando o método de Newton é utilizado para a solução de problemas de mínimos quadrados, ele é comumente referenciado na literatura como método de Gauss-Newton. Sendo que uma das aplicações é a solução de sistemas de equações não-lineares, $F(x) = 0$, transformados em problemas de mínimos quadrados ao procurar minimizar o quadrado dos resíduos, isto é,

$$S(x) = F^T(x)F(x) = \sum_{i=1}^m f_i^2(x)$$

neste caso $\nabla S(x^k) = 2 J^T(x^k) F(x^k)$ e $H(x^k) = 2 J^T(x^k) J(x^k) + 2 Q(x^k)$, onde $J(x) = \left[\frac{\partial f_i}{\partial x_j} \right]_{i,j}$ é a matriz Jacobiana do sistema, $Q(x) = \sum_{i=1}^m f_i(x) H_i(x)$ e $H_i(x)$ é a matriz Hessiana da função $f_i(x)$.

Quando $f_i(x^k) \rightarrow 0$ para $x^k \rightarrow x^*$, então $Q(x^k)$ tende a zero, e as direções de busca do método de Gauss-Newton para o problema de mínimos quadrados:

$$\min_{d \in \mathbb{R}^n} \|J(x^k)d - F(x^k)\|^2$$

são equivalentes as direções do método de Newton, ou seja:

$$d^k = -[J^T(x^k) J(x^k)]^{-1} J^T(x^k) F(x^k) \approx -[H(x^k)]^{-1} \nabla S(x^k)$$

3.2.3 Método do gradiente conjugado

Utiliza somente a primeira derivada da função objetivo, gerando uma seqüência de direções que são combinações lineares do gradiente:

$$d^{k+1} = \varepsilon_{k+1} d^k - \nabla S(x^{k+1}) \tag{3.5}$$

onde a nova direção é conjugada com a direção anterior com respeito a Hessiana:

$$(d^{k+1})^T H(x^k) d^k = 0$$

e $x^{k+1} = x^k + \alpha_k d^k$, onde α_k é obtido de forma similar ao método da maior descida. Para calcular ε_{k+1} , faz-se a aproximação quadrática de $S(x)$, equação (3.4), de onde obtém-se:

$$\nabla S(x) \approx \nabla S(x^k) + H(x^k) (x - x^k) \tag{3.6}$$

e portanto: $\nabla S(x^{k+1}) - \nabla S(x^k) = H(x^k) (x^{k+1} - x^k) = H(x^k) \alpha_k d^k$, que multiplicado por d^{k+1} à esquerda, resulta:

$$(d^{k+1})^T [\nabla S(x^{k+1}) - \nabla S(x^k)] = \alpha_k (d^{k+1})^T H(x^k) d^k = 0$$

substituindo a equação (3.5) na expressão acima, tem-se:

$$[\varepsilon_{k+1} d^k - \nabla S(x^{k+1})]^T [\nabla S(x^{k+1}) - \nabla S(x^k)] = 0,$$

mas devido a ortogonalidade entre a direção de busca e o gradiente da função objetivo no mínimo desta direção $(d^k)^T \nabla S(x^{k+1}) = 0$ e para a aproximação quadrática (3.6) $\nabla^T S(x^{k+1}) \nabla S(x^k) = 0$, resultando em:

$$\varepsilon_{k+1} = -\frac{\nabla^T S(x^{k+1}) \nabla S(x^{k+1})}{(d^k)^T \nabla S(x^k)} = \frac{\nabla^T S(x^{k+1}) \nabla S(x^{k+1})}{\nabla^T S(x^k) \nabla S(x^k)}$$

a última igualdade resulta do fato de $d^k = \varepsilon_k d^{k-1} - \nabla S(x^k)$, que multiplicado por $\nabla S(x^k)$ à direita:

$$(d^k)^T \nabla S(x^k) = \varepsilon_k (d^{k-1})^T \nabla S(x^k) - \nabla^T S(x^k) \nabla S(x^k) = -\nabla^T S(x^k) \nabla S(x^k),$$

pois $(d^{k-1})^T \nabla S(x^k) = 0$, pela mesma razão acima.

algoritmo

- 1) Escolher um ponto inicial x^0
- 2) Calcular $d^0 = -\nabla S(x^0)$, $k = 0$
- 3) Encontrar α_k tal que $S(x^k + \alpha_k d^k) = \min_{\alpha > 0} g_k(\alpha) = S(x^k + \alpha d^k)$
- 4) Calcular $x^{k+1} = x^k + \alpha_k d^k$ e $\nabla S(x^{k+1})$
- 5) **Se** o critério de convergência foi satisfeito, **então** FIM.
- 6) Calcular $d^{k+1} = -\nabla S(x^{k+1}) + d^k \frac{\nabla^T S(x^{k+1}) \nabla S(x^{k+1})}{\nabla^T S(x^k) \nabla S(x^k)}$, $k \leftarrow k + 1$
- 7) **Se** $k = n$, isto é, realizou n direções L.I. **então** fazer $x^o = x^k$ e (ir para 2)
senão (ir para 3)

3.2.4 Métodos da métrica variável

A partir da equação (3.6), pode-se tirar a seguinte relação:

$$x^{k+1} - x^k = \Delta x^k = [H(x^k)]^{-1} [\nabla S(x^{k+1}) - \nabla S(x^k)]$$

Fazendo uso de uma aproximação da inversa da matriz Hessiana:

$$[H(x^k)]^{-1} \approx \omega W(x^{k+1}) = \omega [W(x^k) + \Delta W(x^k)]$$

onde ω é um fator de escala e $\Delta W(x^k) = W(x^{k+1}) - W(x^k)$, resulta em:

$$\Delta x^k = \omega [W(x^k) + \Delta W(x^k)] [\nabla S(x^{k+1}) - \nabla S(x^k)],$$

que rearranjando tem-se:

$$\Delta W(x^k) \Delta f(x^k) = \frac{\Delta x^k}{\omega} - W(x^k) \Delta f(x^k) \quad (3.7)$$

onde $\Delta f(x^k) = \nabla S(x^{k+1}) - \nabla S(x^k)$. Como o sistema (3.7) possui n equações e n^2 variáveis a determinar (os elementos de $\Delta W(x^k)$), então existe um número infinito de soluções, sendo que a solução geral é dada por:

$$\Delta W(x^k) = \frac{1}{\omega} \frac{\{\Delta x^k, u^k\}}{\langle u^k, \Delta f(x^k) \rangle} - \frac{\{W(x^k) \Delta f(x^k), v^k\}}{\langle v^k, \Delta f(x^k) \rangle} \quad (3.8)$$

ou

$$\Delta H(x^k) = \omega \frac{\{\Delta f(x^k), v^k\}}{\langle v^k, \Delta x^k \rangle} - \frac{\{H(x^k) \Delta x^k, u^k\}}{\langle u^k, \Delta x^k \rangle} \quad (3.9)$$

onde u^k e v^k são vetores arbitrários

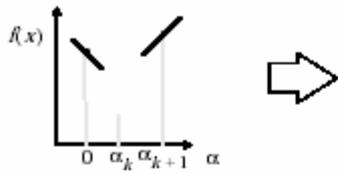
$\{x, y\} = x y^T$ é o produto externo (uma matriz)

$\langle x, y \rangle = x^T y$ é o produto interno (um escalar)

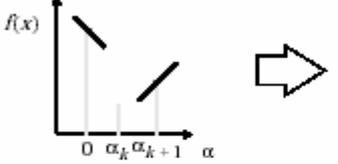
Conseqüentemente, pode existir uma infinidade de métodos que utilizam a equação (3.1), dependendo da escolha de ω , u^k e v^k . O parâmetro α_k pode ser minimizado (“*linesearch*”) ou escolhido adequadamente (“*trust region*”).

Do mesmo modo que no método de Newton o parâmetro α_k é usado essencialmente para assegurar a convergência ($S(x^{k+1}) < S(x^k)$), pois a inversa da matriz Hessiana já contém o tamanho ideal do passo (para funções quadráticas), os métodos quasi-Newton não necessitam minimizar α_k exatamente, pois $W(x^k) \rightarrow [H(x^k)]^{-1}$ a medida que as iterações avançam. Portanto, as técnicas de “*linesearch*” não necessitam encontrar uma solução precisa para α_k , mas ajustá-lo adequadamente. Por exemplo, as técnicas de ajuste usadas nas implementações de “*linesearch*” no MATLAB utilizam os seguintes critérios para a interpolação cúbica, quando são conhecidos dois valores da função objetivo e seus gradientes.

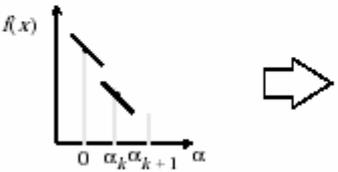
Case 1: $f(x_{k+1}) > f(x_k), \nabla f(x_{k+1})^T d > 0$



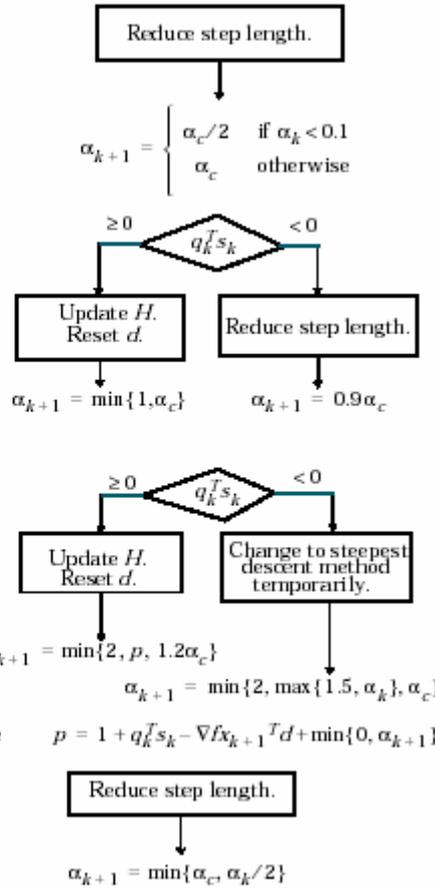
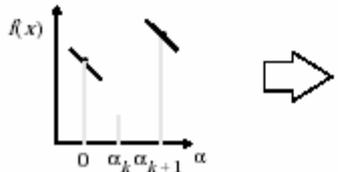
Case 2: $f(x_{k+1}) \leq f(x_k), \nabla f(x_{k+1})^T d \geq 0$



Case 3: $f(x_{k+1}) < f(x_k), \nabla f(x_{k+1})^T d < 0$

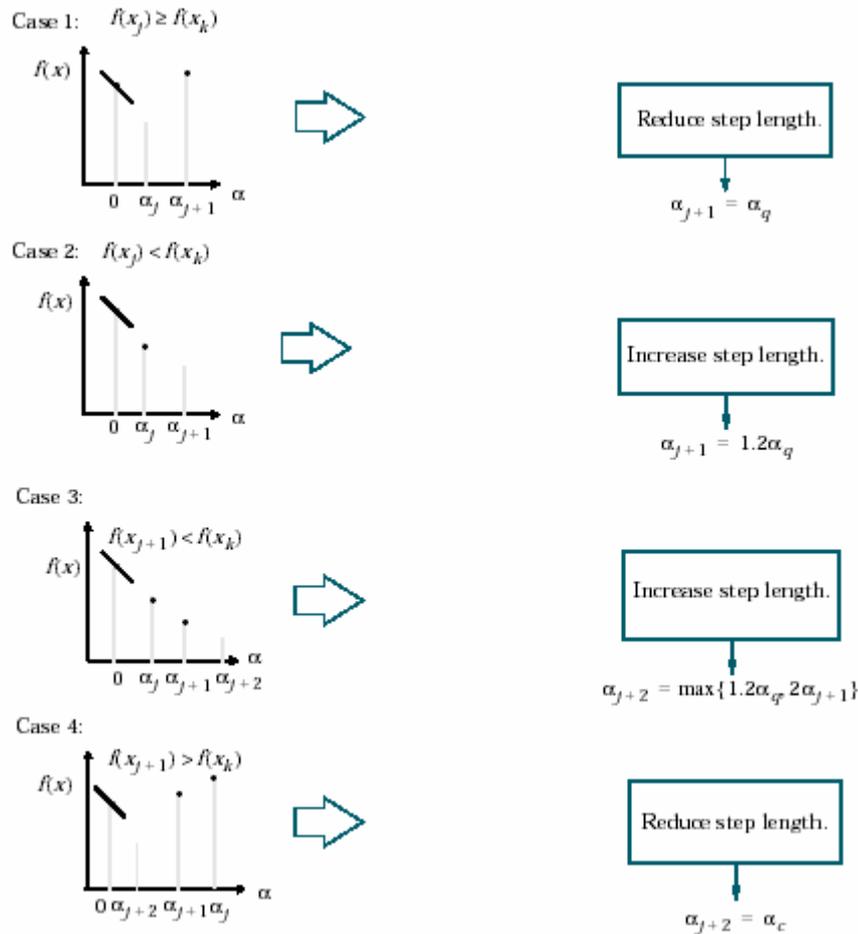


Case 4: $f(x_{k+1}) \geq f(x_k), \nabla f(x_{k+1})^T d \leq 0$



onde α_c é o mínimo obtido por interpolação ou extrapolação cúbica, $f(x) \equiv S(x)$ é a função objetivo, $q^k = \nabla S(x^{k+1}) - \nabla S(x^k)$ e $s^k = \Delta x^k = \alpha_k d^k$. A relação $(q^k)^T s^k > 0$ assegura a positividade da matriz Hessiana (eq. 3.6).

Outra técnica de “*linesearch*” empregada no MATLAB é quando apenas um valor do gradiente da função objetivo é utilizado juntamente com dois ou três valores de $S(x)$, onde uma mescla de interpolação cúbica e quadrática são realizadas. Neste caso os critérios utilizados para a seleção do passo são os seguintes.

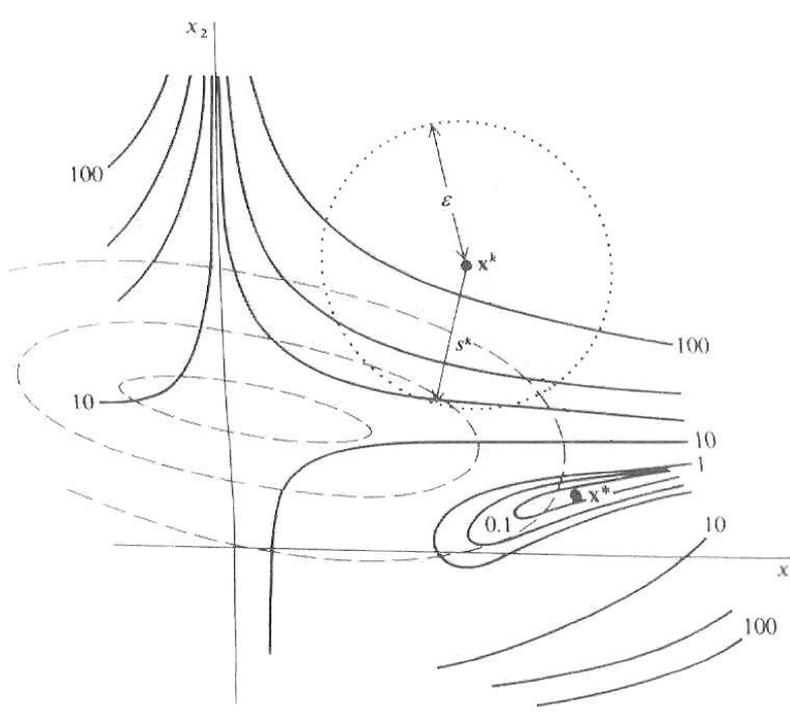


onde α_q é o mínimo obtido por interpolação ou extrapolação quadrática. Em qualquer uma das estratégias, quando o α_q ou α_c resultarem em valores negativos, então $\alpha_{k+1} = 2\alpha_k$.

As técnicas de “trust region” (região de confiança) para o ajuste da direção de busca, consiste em aproximar $S(x)$ por uma função mais simples, $Q(x)$, (geralmente uma forma quadrática) que represente razoavelmente bem a função $S(x)$ em uma vizinhança $B_\varepsilon(x)$ em torno de x . Então um novo passo, $s^k = \Delta x^k = \alpha_k d^k$, é obtido pela minimização de $Q(x)$ em $B_\varepsilon(x)$, ou seja:

$$s^k = \arg \min_s Q(x^k + s) \quad , \quad (x^k + s) \in B_\varepsilon(x^k)$$

se $S(x^k + s^k) < S(x^k)$, então o novo ponto é aceito, caso contrário a região de confiança é reduzida e o procedimento é repetido. A figura abaixo ilustra esta técnica, onde as curvas de níveis tracejadas representam a aproximação quadrática convexa de $S(x)$ no ponto x^k .



As peculiaridades dos métodos da região de confiança estão relacionadas com a escolha da função $Q(x)$, a forma de redução de $B_\varepsilon(x)$ e o quão preciso é resolvido o problema de minimização de $Q(x^k + s)$.

algoritmo da métrica variável (com *linsearch*)

- 1) Escolher x^0 , ω , u^k e v^k
- 2) Calcular $W(x^0)$, $k = 0$
- 2) Calcular $d^k = -W(x^k) \nabla S(x^k)$
- 3) Encontrar α_k tal que $S(x^k + \alpha_k d^k) \approx \min_{\alpha > 0} g_k(\alpha) = S(x^k + \alpha d^k)$
- 4) Calcular $x^{k+1} = x^k + \alpha_k d^k$, $S(x^{k+1})$ e $\Delta f(x^k) = \nabla S(x^{k+1}) - \nabla S(x^k)$
- 5) **Se** o critério de convergência foi satisfeito, **então** FIM.
- 6) calcular $W(x^{k+1})$, $k \leftarrow k + 1$, verificar a necessidade de *reset* e (ir para 3)

onde *reset* é uma reinicialização (ou modificação) de $W(x^k)$ devido a problemas de convergência. No caso do método dos gradientes conjugados, o *reset* acontece a cada n iterações (n direções linearmente independentes). Para os demais casos, ele é necessário quando a matriz $W(x^k)$ deixa de ser positiva definida, ou quando a correção $\Delta W(x^k)$ torna-se divergente devido a erros de arredondamento, ou ainda quando d^k estiver na mesma direção de d^{k-1} , tornando $W(x^{k+1})$ singular.

No algoritmo acima, quando a interpolação cúbica é usada, então os passos (3) e (4) são concomitantes.

Definindo:

$$\begin{array}{ll}
 W = W(x^k) & W^{-1} = [W(x^k)]^{-1} \\
 \Delta x = \Delta x^k & \Delta f = \Delta f(x^k) \\
 d = \Delta x - W \Delta f & \hat{d} = \Delta f - W^{-1} \Delta x \\
 \delta = \langle \Delta x, \Delta f \rangle & \hat{\delta} = \langle \Delta f, \Delta f \rangle \\
 \gamma = \langle \Delta f, W \Delta f \rangle & \hat{\gamma} = \langle \Delta x, W^{-1} \Delta x \rangle \\
 A = \{ \Delta x, \Delta x \} / \delta & \hat{A} = \{ \Delta f, \Delta f \} / \delta \\
 B = \{ W \Delta f, W \Delta f \} / \gamma & \hat{B} = \{ W^{-1} \Delta x, W^{-1} \Delta x \} / \hat{\gamma} \\
 E = \{ \Delta x, \Delta f \} & F = \{ \nabla S(x^k), \nabla S(x^{k+1}) \} / \langle \nabla S(x^k), \nabla S(x^k) \rangle
 \end{array}$$

é apresentado na tabela a seguir alguns métodos de otimização que utilizam a primeira e/ou a segunda derivada da função objetivo. Por exemplo, para o método BFGS a atualização apresenta a seguinte formulação:

$$H(x^{k+1}) = H(x^k) + \frac{\Delta f^k (\Delta f^k)^T}{(\Delta x^k)^T \Delta f^k} - \frac{H(x^k) \Delta x^k (\Delta x^k)^T [H(x^k)]^T}{(\Delta x^k)^T H(x^k) \Delta x^k}$$

que ao multiplicar à esquerda por $(\Delta x^k)^T$ e à direita por Δx^k , resulta em:

$$(\Delta x^k)^T H(x^{k+1}) \Delta x^k = (\Delta f^k)^T \Delta x^k = (q^k)^T s^k$$

mostrando que se $(q^k)^T s^k > 0$, então $H(x^{k+1})$ é positiva definida.

método	$W(x^{k+1})$
<i>Steepest descent</i>	I
Gradiente Conjugado	$I + W F$
Newton	$[H(x^{k+1})]^{-1}$
Greenstadt, $\lambda_i(C) \rightarrow \max\{ \lambda_i , 10^{-4}\}$	$D^{-1} C D^{-1}$; $D = \text{diag}(W^{-1} _{i,j}^{1/2})$ e $C = D^{-1} W D^{-1}$
Levenberg-Marquardt	$D^{-1} (C + \beta I) D^{-1}$; $\beta > -\min(\text{valor caract. de } C)$
Broyden	$W + \frac{\{d, d\}}{\langle d, \Delta f \rangle}$
Davidon-Fletcher-Powell (DFP)	$W + A - B$
Pearson I	$W + \frac{\{d, \Delta x\}}{\delta} = W + A - \gamma \hat{B}^{-1} E^T$
Pearson II	$W + \frac{\{d, W \Delta f\}}{\gamma} = \left(1 + \frac{\delta}{\gamma}\right) W - B$
Newton Projetado (Zoutendijk)	$W - B$
Greenstadt-Goldfarb I (GGI)	$W + \frac{1}{\hat{\delta}} \left(\{d, \Delta f\} + \{\Delta f, d\} - \frac{\delta}{\hat{\delta}} \langle \Delta f, d \rangle \hat{A} \right)$
Greenstadt-Goldfarb II (GGII)	$W + \frac{E W + W E^T}{\gamma} - \left(1 + \frac{\delta}{\gamma}\right) B$
Gill-Murray ou (GGIII)	$W + \left(1 + \frac{\gamma}{\delta}\right) A - \frac{E W + W E^T}{\delta}$
Fletcher ou (GGIII) ⁻¹	$\left[W^{-1} + \left(1 + \frac{\hat{\gamma}}{\delta}\right) \hat{A} - \frac{E^T W^{-1} + W^{-1} E}{\delta} \right]^{-1}$
Broyden-Fletcher-Goldfarb-Shanno (BFGS) ou (DFP) ⁻¹	$\left[W^{-1} + \hat{A} - \hat{B} \right]^{-1}$
(Broyden) ⁻¹	$\left[W^{-1} + \frac{\{\hat{d}, \hat{d}\}}{\{\hat{d}, \Delta x\}} \right]^{-1}$
Goldstein-Price	aproximação por diferenças-finitas

4. Teoria da Dualidade

Problemas de otimização com restrição apresentam duas formas de representação: o *problema primal (P)* e o *problema dual (D)*.

4.1 Problema primal

Genericamente, o problema primal pode ser escrito da seguinte forma:

$$P = \begin{cases} \min S(x) \\ \text{sujeito a: } h_j(x) = 0, j = 1, 2, \dots, m \\ g_j(x) \leq 0, j = 1, 2, \dots, p \\ x \in X \subseteq \mathfrak{R}^n \end{cases}$$

onde X é um conjunto convexo não vazio.

Definindo a função perturbação, $v(y)$, abaixo associada ao problema primal (P):

$$v(y) = \begin{cases} \inf S(x) \\ \text{sujeito a: } h_j(x) = 0, j = 1, 2, \dots, m \\ g_j(x) \leq y, j = 1, 2, \dots, p \\ x \in X \subseteq \mathfrak{R}^n \end{cases}$$

onde $v(0)$ corresponde ao valor ótimo de P . Se o conjunto Y , definido como:

$$Y = \{y \in \mathfrak{R}^p \mid h(x) = 0, g(x) \leq y \text{ para algum } x \in X\}$$

for convexo, então $v(y)$ é uma função convexa em Y , e $v(y) = \infty \Leftrightarrow y \notin Y$. A propriedade de convexidade de $v(y)$ é o elemento fundamental para a relação entre os problemas primal e dual. Particularmente, se $h(x)$ é linear e $g(x)$ é convexo em X , então Y é um conjunto convexo.

O problema P é *estável* se $v(0)$ é finito e $v(y)$ é uma função Lipschitz contínua, isto é, existe um escalar $L > 0$ tal que:

$$v(0) - v(y) \leq L \|y\|, \quad \forall y \neq 0$$

Esta definição de estabilidade não depende da norma $\|y\|$ usada, sendo necessária e suficiente a consideração de um vetor y variando em uma dimensão.

Seja x^* a solução ótima de P , então os multiplicadores λ^* e μ^* existem se e somente se P for estável. Portanto, a estabilidade de P é uma qualificação necessária e suficiente das restrições. Vale lembrar que a existência dos multiplicadores de

Lagrange depende da forma das restrições, e estará garantida se e somente se os gradientes das restrições de desigualdade ativas, $\nabla g_j(x^*)$, e das restrições de igualdade, $\nabla h(x^*)$, forem linearmente independentes.

Se a função objetivo, $S(x)$, for suficientemente bem comportada, então o problema P será estável, mesmo que as restrições não sejam bem comportadas. Por exemplo, se $S(x) = c$, então P é estável desde que as restrições estejam satisfeitas. Por outro lado, se a restrição for bem comportada (por exemplo, linear) é ainda possível que P seja instável.

Exemplo 4.1: Seja o problema:

$$\begin{aligned} \min S(x) &= -x^{1/3} \\ \text{sujeito a: } g(x) &= x \leq 0 \\ x \in X &= \{x \mid x \geq 0\} \end{aligned}$$

Perturbando $g(x)$ na direção positiva tem-se:

$$v(0) - v(y) = 0 - (-y^{1/3}) = y^{1/3}, \quad \|y\| = |y| \quad \text{e} \quad 1 \leq L y^{2/3}$$

ou seja, para $y \rightarrow 0$ tem-se $L \rightarrow \infty$, isto é, $v(y)$ não é Lipschitz contínua.

4.2 Problema dual

O problema dual de P tem a seguinte forma:

$$D = \begin{cases} \max_{\mu \geq 0, \lambda} \inf_{x \in X} L(x, \lambda, \mu) \\ \text{sujeito a: } L(x, \lambda, \mu) = S(x) + \lambda^T h(x) + \mu^T g(x) \end{cases}$$

Como o problema interno $\phi(\lambda, \mu) = \inf_{x \in X} L(x, \lambda, \mu)$ do dual D é uma função paramétrica de

λ e μ , $\phi(\lambda, \mu)$, também chamado de *função dual*, ele pode assumir valores $(-\infty)$ para algum valor de λ ou μ . Se o problema interno é finito para todos $\mu \geq 0$ e λ , então ele pode ser escrito como:

$$\phi(\lambda, \mu) = \min_{x \in X} L(x, \lambda, \mu)$$

Observe que para um valor de x fixo no ínfimo de $L(x, \lambda, \mu)$, o problema externo, $\max_{\mu \geq 0, \lambda} \phi(\lambda, \mu)$, torna-se linear em λ e μ .

Sejam $S(x)$, $h(x)$ e $g(x)$ funções contínuas e $X \subseteq \mathfrak{R}^n$ um conjunto compacto não vazio, então $\phi(\lambda, \mu)$ é uma função côncava. Observe que $\phi(\lambda, \mu)$ é côncava sem a necessidade de considerar algum tipo de convexidade da função objetivo ou de suas

restrições, pois ela é uma coleção de pontos ínfimos de funções lineares em λ e μ . Como $\phi(\lambda, \mu)$ é côncava e o problema externo é uma maximização, então o ótimo de $\phi(\lambda, \mu)$ é global. Contudo, a dificuldade surge da forma paramétrica de $\phi(\lambda, \mu)$ em relação a λ e μ , isto é, não se tem uma forma funcional explícita de λ e μ .

4.3 Interpretação geométrica do dual

Considere o problema primal (P) monovariável:

$$P = \begin{cases} \min S(x) \\ \text{sujeito a: } g_1(x) \leq 0, \\ \phantom{\text{sujeito a:}} g_2(x) \leq 0, \\ \phantom{\text{sujeito a:}} x \in X \subseteq \mathfrak{R} \end{cases}$$

e seu dual (D):

$$D = \begin{cases} \max_{\mu \geq 0} \min_{x \in X} L(x, \mu) \\ \text{sujeito a: } L(x, \mu) = S(x) + \mu_1 g_1(x) + \mu_2 g_2(x) \end{cases}$$

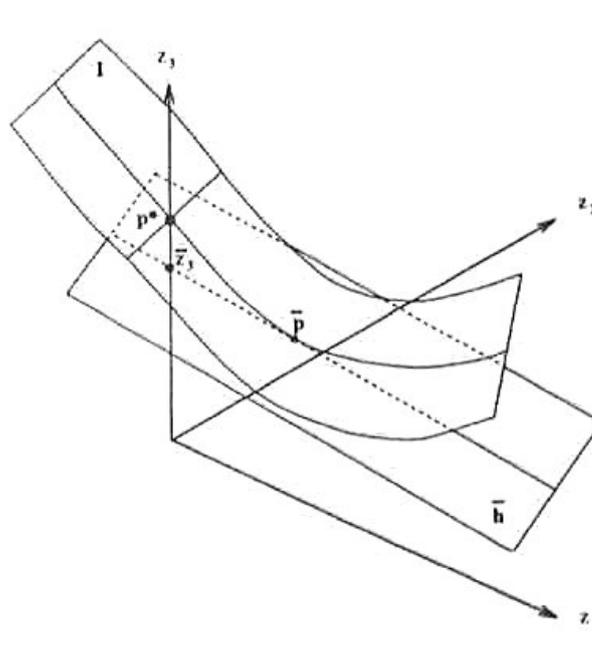
O conjunto $I \subseteq \mathfrak{R}^3$ que é a imagem de X sob as três funções $S(x)$, $g_1(x)$ e $g_2(x)$ é dado por:

$$I = \{(z_1, z_2, z_3) \in \mathfrak{R}^3 \mid z_1 = g_1(x), z_2 = g_2(x) \text{ e } z_3 = S(x) \text{ para algum } x \in X\}$$

Sobre este conjunto a formulação equivalente do dual é dada por:

$$D = \begin{cases} \max_{\mu \geq 0} \min_{z \in I} L(z, \mu) \\ \text{sujeito a: } L(z, \mu) = z_3 + \mu_1 z_1 + \mu_2 z_2 \end{cases}$$

Para valores fixos dos multiplicadores $\bar{\mu}_1 \geq 0$ e $\bar{\mu}_2 \geq 0$, $L(z, \bar{\mu}) = \text{constante}$ é um plano no \mathfrak{R}^3 com inclinação $(-\bar{\mu}_1, -\bar{\mu}_2)$. O conjunto I e um plano tangente, \bar{h} , no ponto $\bar{P} \in I$ estão ilustrados na figura abaixo.



Interpretação geométrica do problema primal: O ponto de interseção de I com o eixo z_3 , denotado por P^* na figura acima, é a imagem da solução ótima, x^* , do problema primal:

$$P^* = [g_1(x^*), g_2(x^*), S(x^*)]$$

Portanto, o problema primal consiste na determinação do ponto em I que minimiza z_3 sujeito a $z_1 \leq 0$ e $z_2 \leq 0$.

Interpretação geométrica da função dual: A função dual $\phi(\mu)$ no ponto $(\bar{\mu}_1, \bar{\mu}_2)$ corresponde a determinação do plano mais baixo com inclinação $(-\bar{\mu}_1, -\bar{\mu}_2)$ que intercepta o conjunto I , ou seja $\min_{z \in I} L(z, \bar{\mu})$. Isto corresponde ao hiperplano suporte \bar{h} que é tangente ao conjunto I no ponto \bar{P} mostrado na figura acima.

Note que o ponto \bar{P} é a imagem de x no $\min_{x \in X} S(x) + \bar{\mu}_1 g_1(x) + \bar{\mu}_2 g_2(x)$, cuja solução é o valor de z_3 onde o hiperplano \bar{h} intercepta a ordenada, denotada por \bar{z}_3 na figura acima, pois $L(z, \bar{\mu}) = \text{constante} = z_3$ quando $z_1 = z_2 = 0$.

Interpretação geométrica do problema dual: O ponto P^* na figura acima corresponde aos valores de $\bar{\mu}_1$ e $\bar{\mu}_2$ que maximizam \bar{z}_3 . Portanto o problema dual consiste na determinação dos valores de $\bar{\mu}_1$ e $\bar{\mu}_2$ que definem a inclinação do hiperplano suporte do conjunto I , tal que ele intercepta a ordenada na valor mais alto possível.

Nem sempre é possível obter o valor ótimo do problema dual igual ao valor ótimo do problema primal. Esta igualdade está garantida quando $S(x)$ e $g(x)$ forem funções convexas, $h(x)$ for uma função afim (mapeamento linear de x) e o problema primal for estável. A diferença entre os valores ótimos dos problemas primal e dual é

conhecida como brecha dual (*duality gap*). Esta diferença pode ser devido a uma descontinuidade da função perturbação $v(y)$ em $y = 0$, que impossibilita a existência de um hiperplano suporte neste ponto.

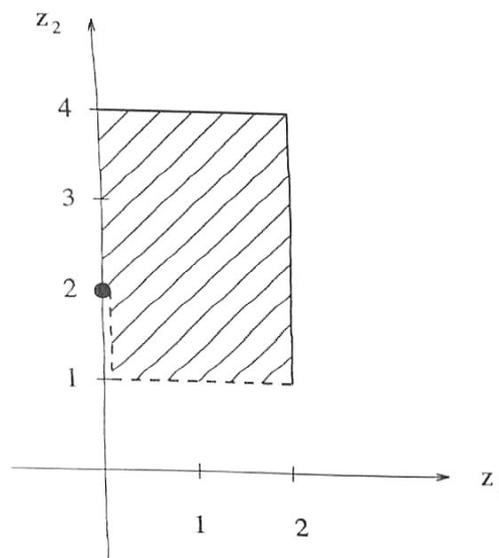
Se a função perturbação $v(y)$ é finita em $y = 0$, então não existe brecha dual se e somente se $v(y)$ é semicontínua inferior em $y = 0$. Para que isto ocorra as seguintes condições devem ser satisfeitas:

- a) X é um conjunto fechado;
- b) $S(x)$ e $g(x)$ são funções contínuas em X ;
- c) $S(x^*)$ é finito;
- d) $\{x \in X \mid h(x) = 0, g(x) \leq 0 \text{ e } S(x) \leq \alpha\}$ é um conjunto limitado, não vazio e convexo, para algum escalar $\alpha \geq v(0)$.

Exemplo 4.2: Seja o seguinte problema:

$$P = \begin{cases} \min S(x) = x_2 \\ \text{sujeito a: } x_1 \leq 0, \\ x \in X = \{(x_1, x_2) \mid 0 \leq x_1 \leq 2, 1 \leq x_2 \leq 4 \text{ e } x_2 \geq 2 \text{ se } x_1 = 0\} \subseteq \mathbb{R}^2 \end{cases}$$

onde $z_1 = x_1$ e $z_2 = x_2$. A figura abaixo ilustra a imagem de X deste problema.



A solução ótima do problema primal é obtida no ponto $x_1 = 0$ e $x_2 = 2$. Contudo, o valor ótimo do problema dual não pode ser igual ao do primal devido a perda de semicontinuidade inferior de $v(y)$ em $y = 0$.

4.4 Dualidade fraca e forte

A possível existência da brecha dual leva a necessidade da definição de dualidade fraca e dualidade forte.

Dualidade Fraca: seja \bar{x} uma solução viável do problema primal (P) e $(\bar{\lambda}, \bar{\mu})$ uma solução viável do problema dual (D). Então,

$$S(\bar{x}) \geq \phi(\bar{\lambda}, \bar{\mu})$$

Observe que qualquer solução viável do problema D é um limite inferior da solução ótima do problema P e qualquer solução viável do problema P é um limite superior para a solução ótima do problema D . Estas relações de limites são importantes para estabelecer critérios de parada em algoritmos de otimização. Isto é, se em alguma iteração de um algoritmo existirem soluções viáveis para os problemas primal e dual e estas estiverem suficientemente próximas, então elas podem ser consideradas soluções ótimas dentro da precisão desejada. Além disto, estas relações não estão baseadas em considerações de convexidade, portanto são de grande importância para tratar de problemas não convexos, desde que o problema dual seja resolvido eficientemente.

Nota: se o valor ótimo de P é $-\infty$ então D deve ser inviável, ao passo que se o valor ótimo de D é $+\infty$ então P deve ser inviável.

Dualidade Forte: sejam $S(x)$ e $g(x)$ funções convexas, $h(x)$ função afim e $X \subseteq \mathfrak{R}^n$ um conjunto convexo não vazio. Se o problema primal (P) é estável, então

- o problema dual (D) tem uma solução ótima;
- os valores ótimos dos problemas P e D são iguais;
- (λ^*, μ^*) é uma solução ótima de $D \Leftrightarrow (-\lambda^*, -\mu^*)$ é um sub-gradiente da função perturbação $v(y)$ em $y = 0$, isto é,

$$v(y) \geq v(0) + d^T y, \text{ onde } d \text{ é um vetor sub-gradiente}$$

- todas as soluções ótimas (λ^*, μ^*) do problema D caracterizam o conjunto de todas as soluções ótimas do problema P obtidas pela minimização da função de Lagrange: $L(x, \lambda^*, \mu^*) = S(x) + (\lambda^*)^T h(x) + (\mu^*)^T g(x)$ para $x \in X$, satisfazendo as restrições $h(x) = 0$, $g(x) \leq 0$ e a condição de complementaridade $(\mu^*)^T g(x) = 0$.

Exemplo 4.3: Seja o seguinte problema não convexo (devido a bilinearidade de g):

$$P = \begin{cases} \min S(x) = -x_1 - x_2 \\ \text{sujeito a: } g(x) = x_1 x_2 - 4 \leq 0, \\ x \in X = \{(x_1, x_2) \mid 0 \leq x_1 \leq 4, 0 \leq x_2 \leq 8\} \subseteq \mathfrak{R}^2 \end{cases}$$

cuja ótimo global é $x^* = [0,5 \ 8]^T$ e $S(x^*) = -8,5$. O seu dual (D) somente em relação a restrição $g(x)$ é dado por:

$$D = \begin{cases} \max_{\mu \geq 0} \min_{x \in X} L(x, \mu) \\ \text{sujeito a: } L(x, \mu) = -x_1 - x_2 + \mu (x_1 x_2 - 4) \end{cases}$$

Fazendo $\nabla_x L(x, \mu) = 0$, tem-se como resultado $x_1 = x_2 = 1/\mu$ e, portanto, a função dual (mínimo de L em relação a x) toma a forma:

$$\phi(\mu) = -1/\mu - 4\mu$$

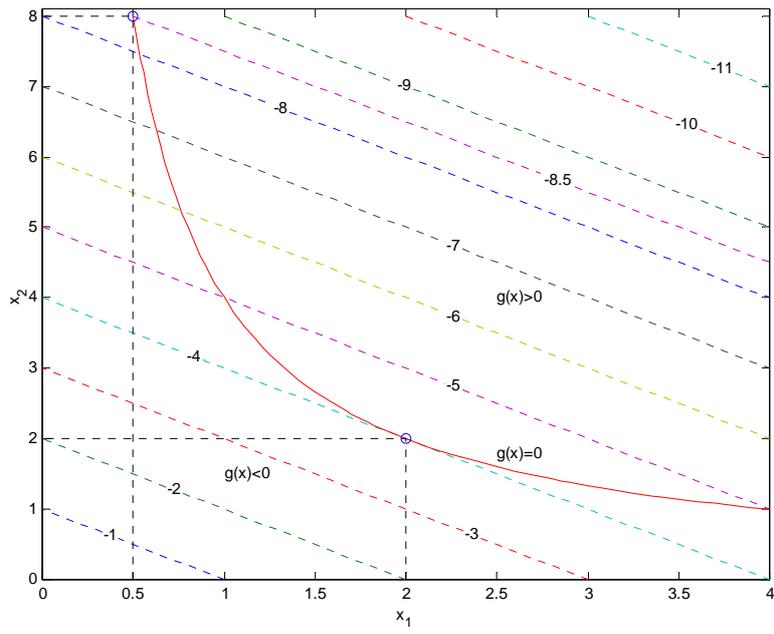
cujo máximo ($\nabla \phi = 0$) ocorre quando $\bar{\mu} = 0,5$ e $\phi(\bar{\mu}) = -4$. Neste ponto $\bar{x} = (2, 2)$ e $S(\bar{x}) = -4 \geq \phi(\bar{\mu})$. Contudo $S(x^*) < \phi(\bar{\mu})$, contrariando o limite inferior da dualidade fraca. O problema é que o ponto \bar{x} na verdade é um ponto de máximo de $S(x)$ sobre a restrição $g(x)$, pois para vetores direção satisfazendo $d^T \nabla g(\bar{x}) = 0$, tem-se $d^T \nabla_x^2 L(\bar{x}, \bar{\mu}) d < 0$. Por outro lado, se a restrição $x_2 \leq 8$ for incluída na formulação do dual:

$$D = \begin{cases} \max_{\mu \geq 0} \min_{x \in X} L(x, \mu) \\ \text{sujeito a: } L(x, \mu) = -x_1 - x_2 + \mu_1 (x_1 x_2 - 4) + \mu_2 (x_2 - 8) \end{cases}$$

O mínimo de L em relação a x é dado por: $x_1 = (1 - \mu_2) / \mu_1$ e $x_2 = 1/\mu_1$. Neste caso a função dual é dada por:

$$\phi(\mu) = -1/\mu_1 - 4\mu_1 + \mu_2/\mu_1 - 8\mu_2$$

cujo máximo ocorre quando $\bar{\mu} = [1/8 \ 15/16]^T$ e $\phi(\bar{\mu}) = -8,5$. Neste ponto $\bar{x} = x^*$ e $S(\bar{x}) = -8,5 \geq \phi(\bar{\mu})$. A figura abaixo ilustra as duas situações descritas acima.



5. Programação Linear (LP)

Na otimização com restrição o problema que está sendo resolvido é:

$$\min S(x) \quad \text{ou} \quad \max S(x)$$

sujeito a restrições de igualdade e/ou de desigualdade, que definem a região viável, sendo que qualquer ponto nesta região é uma solução viável. Dependendo do tipo de função objetivo e de suas restrições, os problemas de otimização com restrição são comumente chamados de programação linear, programação quadrática, programação não-linear, programação inteira e programação mista.

Programação linear: função objetivo e restrições lineares

$$\begin{aligned} \min S(x) &= c^T x \\ \text{sujeito a: } A x &\leq b \\ x &\geq 0 \end{aligned}$$

onde A é uma matriz $m \times n$, isto é, m restrições e n variáveis.

Programação quadrática: função objetivo quadrática e restrições lineares

$$\begin{aligned} \min S(x) &= c^T x + \frac{1}{2} x^T Q x \\ \text{sujeito a: } A x &\leq b \\ x &\geq 0 \end{aligned}$$

Programação não-linear: função objetivo e/ou restrições não-lineares

$$\begin{aligned} \min S(x) \\ \text{sujeito a: } h_j(x) &= 0, \quad j = 1, 2, \dots, m \\ g_j(x) &\leq 0, \quad j = 1, 2, \dots, p \end{aligned}$$

onde $h_j(x)$ são m restrições de igualdade e $g_j(x)$ são p restrições de desigualdade.

Programação inteira: variáveis de decisão pertencem ao campo dos números inteiros.

Programação mista: é uma combinação da programação inteira com as demais. Por exemplo, no problema de projeto de trocadores de calor as variáveis de decisão poderiam ser as temperaturas das correntes (campo real) e os tipos de trocadores existentes (campo inteiro).

Neste capítulo é tratado o problema da programação linear. As demais formulações são assuntos dos capítulos subsequentes.

Lista-se abaixo alguns exemplos típicos de problemas de gerenciamento de plantas industriais que recaem em programação linear:

- a) associar trabalhadores a horários diários de trabalho de modo a adequar as necessidades da planta, maximizando a satisfação do trabalhador e a produtividade;
- b) selecionar produtos a serem fabricados para o próximo período, tirando o melhor proveito dos recursos disponíveis e dos preços atuais para maximizar os lucros;
- c) determinar o padrão de distribuição de produtos das plantas para os armazéns de modo a minimizar os custos dentro dos limites de capacidade.

A programação linear é um tipo de programação convexa, onde a função objetivo é convexa e as restrições formam um conjunto convexo, portanto o ótimo local é também global.

5.1 Fundamentos

Semi-espaco vetorial aberto: $H = \{x / c^T x < \alpha, x \in \mathfrak{R}^n, \alpha \in \mathfrak{R}, c \neq 0 \in \mathfrak{R}^n\}$

Semi-espaco vetorial fechado: $\bar{H} = \{x / c^T x \leq \alpha, x \in \mathfrak{R}^n, \alpha \in \mathfrak{R}, c \neq 0 \in \mathfrak{R}^n\}$

Hiperplano: $h = \{x / c^T x = \alpha, x \in \mathfrak{R}^n, \alpha \in \mathfrak{R}, c \neq 0 \in \mathfrak{R}^n\}$

Politopo: $\bigcap_{i=1}^m \bar{H}_i$

Poliedro convexo: Politopo limitado.

Casca convexa: é o conjunto convexo, $C(X)$, formado pela interseção de todos conjuntos convexos no \mathfrak{R}^n que contêm o conjunto $X \subseteq \mathfrak{R}^n$ (convexo ou não convexo).

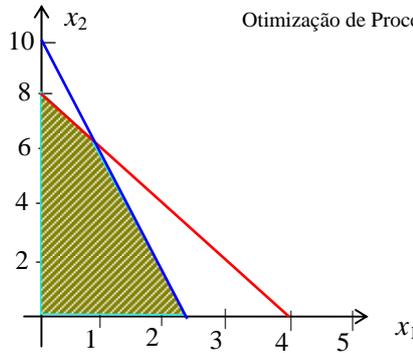
Se $x \in C(X)$, então

$$x = \sum_{i=1}^{n+1} \alpha_i x^i \quad \text{com} \quad \sum_{i=1}^{n+1} \alpha_i = 1, \quad \alpha_i \geq 0 \quad \text{e} \quad x^i \in X \quad (\text{teorema de Caratheodory}).$$

A região de busca da programação linear é formada por interseções de planos, dados pelas restrições lineares, $Ax \leq b$ e $x \geq 0$, gerando um poliedro convexo, isto é, um conjunto $P = \{x / Ax \leq b \text{ e } x \geq 0\}$ de vetores no \mathfrak{R}^n . Por exemplo, para as seguintes restrições:

$$\begin{aligned} 2x_1 + x_2 &\leq 8 \\ 4x_1 + x_2 &\leq 10 \\ x_1 &\geq 0 \text{ e } x_2 \geq 0 \end{aligned}$$

isto é, $A = \begin{bmatrix} 2 & 1 \\ 4 & 1 \end{bmatrix}$ e $b = [8 \ 10]^T$, o poliedro formado é mostrado abaixo.



A solução da programação linear pode ser enunciada através do seguinte teorema:

“O ótimo de uma função linear em um poliedro convexo $P \subseteq \mathbb{R}^n$ é obtido em no mínimo um vértice. Se ele é obtido em mais de um vértice, então ele é obtido em todos os pontos pertencentes a combinação convexa destes vértices”.

prova: sejam x^1, x^2, \dots, x^p , os vértices do poliedro P , então, fazendo:

$$S^* = \min_{k=1,2,\dots,p} \{S(x^k)\}$$

e sabendo que $\forall x \in P$ pode ser obtido pela combinação convexa $x = \sum_{k=1}^p \alpha_k x^k$, onde

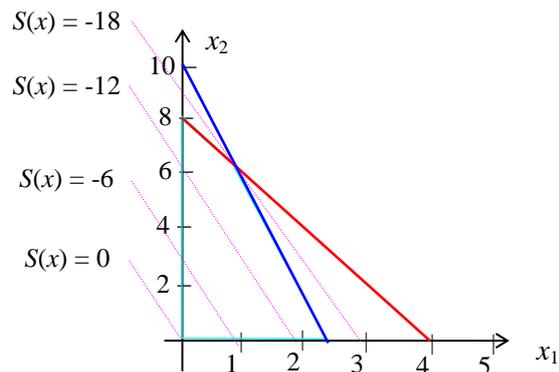
$\alpha_k \geq 0$ e $\sum_{k=1}^p \alpha_k = 1$, então $S(x) = \sum_{k=1}^p \alpha_k S(x^k)$, pois $S(x)$ é linear e

$$S(x) \geq S^* \sum_{k=1}^p \alpha_k = S^*.$$

Exemplo 5.1: $\min S(x_1, x_2) = -6x_1 - 2x_2$

sujeito a $2x_1 + x_2 \leq 8$
 $4x_1 + x_2 \leq 10$
 $x_1 \geq 0$ e $x_2 \geq 0$

$$x_2 = -\frac{S(x)}{2} - 3x_1$$



Se o problema de programação linear não está bem posto, ele pode apresentar uma solução não única (quando a função objetivo é paralela a restrição com melhor vértice), ou uma solução infinita (quando a região viável não é limitada), ou nenhuma solução (quando as restrições não formam uma região viável), tais problemas são ditos *degenerados*.

5.2 Método simplex

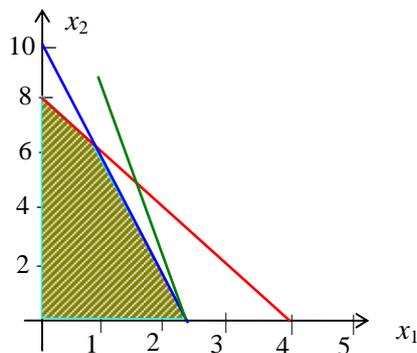
A generalização para o \mathcal{R}^n , chamado de método simplex (Dantzig, 1947), que é um método da classe dos conjuntos ativos (“*active set*”), é feita através da transformação das desigualdades em igualdades, pela inclusão das chamadas variáveis de folga, $f_i \geq 0$ ($i = 1, 2, \dots, m$, onde m é o número de restrições):

$$A x \leq b \Rightarrow A x + f = b, \quad b \geq 0$$

$$A x \geq b \Rightarrow A x - f = b, \quad b \geq 0$$

gerando um sistema sub-determinado de m equações com $p = n+m$ variáveis. Portanto, é necessário especificar n variáveis (chamadas de *variáveis não básicas*) para obter as demais (*variáveis básicas*) em função destas. Se algum $b_i < 0$, então basta multiplicar a inequação correspondente por (-1) para deixá-lo na forma padrão, antes de introduzir a variável de folga.

Quando atribui-se valores nulos para as variáveis não básicas e obtém-se valores não negativos para as variáveis básicas, tem-se uma *solução básica viável* (ou simplesmente *solução básica*), caso contrário tem-se um ponto inviável. Se alguma variável básica for nula, então tem-se uma *solução básica degenerada*, sendo necessário substituir as variáveis básicas nulas por variáveis não básicas. Esta situação ocorre na figura abaixo no ponto $x = (5/2, 0)$, onde tem-se $x_2 = f_2 = f_3 = 0$, isto é, como $n = 2$ e $m = 3$, o número de variáveis não básicas é 2 e o número de variáveis nulas é $3 > n$.



O método simplex busca o ótimo ao longo das restrições, pulando de um vértice viável para outro adjacente até não haver mais progressão. Para descrever o algoritmo simplex, observa-se que o problema de otimização modificado pode ser representado pela matriz aumentada

$$B = \begin{bmatrix} A & \pm I_m \\ c^T & 0 \end{bmatrix}$$

de dimensão $(m+1) \times p$, e pelo vetor aumentado $y = [x \ f]^T$, de dimensão p , onde I_m é a matriz identidade de tamanho $m \times m$. O sistema resultante é dado por:

$$B y = [b \ S(x)]^T$$

Considerando como ponto de partida o vértice $x = 0$, então as variáveis de folga, f , formam um conjunto de variáveis básicas do sistema. As demais variáveis, x , formam o conjunto de variáveis não básicas. Partindo desta formação, o algoritmo simplex (minimização) pode ser descrito da seguinte forma:

algoritmo

- 1) Se $c_i \geq 0 \ \forall \ i = 1, 2, \dots, n$, então a solução ótima foi encontrada, bastando igualar a zero as variáveis não básicas e resolver o sistema $[A \ I_m] y = b$. FIM.
- 2) Pivotamento: identificar o k -ésimo elemento da linha $m+1$ da matriz B com o menor valor (corresponde ao índice da variável que causa a maior redução da função objetivo). A k -ésima variável será incluída na base. Identificar o s -ésimo elemento do vetor $b_i / a_{i,k}$ com menor valor positivo (corresponde a restrição mais próxima, e valores negativos não violam a restrição com o aumento da k -ésima variável). A s -ésima variável de folga será removida da base.
- 3) Eliminação de Gauss-Jordan: executar as operações elementares na matriz B (para levar de uma solução básica para outra), tomando como elemento pivô: $a_{s,k}$ e (ir para 1).

Exemplo 5.2: resolvendo o exemplo anterior pelo método simplex, tem-se:

x_1	x_2	f_1	f_2	b		x_1	x_2	f_1	f_2	b		x_1	x_2	f_1	f_2	b
2	1	1	0	8	\Rightarrow	0	1/2	1	-1/2	3	\Rightarrow	0	1	2	-1	6
4	1	0	1	10		1	1/4	0	1/4	5/2		1	0	-1/2	1/2	1
-6	-2	0	0	S		0	-1/2	0	3/2	S+15		0	0	1	1	S+18
base: f_1 e f_2						base: x_1 e f_1						base: x_1 e x_2				

A solução do sistema $[A \ I_m] y = b$, após eliminação de Gauss-Jordan, com $f_1 = f_2 = 0$ (variáveis não básicas), é dada por:

$$\begin{bmatrix} 0 & 1 & -2 & -1 \\ 1 & 0 & -1/2 & 1/2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \end{bmatrix} \Rightarrow x_1 = 1 \text{ e } x_2 = 6$$

Observe que ao igualar as variáveis não básicas a zero, as variáveis básicas são obtidas simplesmente fazendo $x_i = b_i$. Deste modo, se algum $b_i = 0$ significa que há dependência entre restrições, gerando uma solução básica degenerada.

Outra observação importante é que ao escrever a função de Lagrange para o problema de programação linear tem-se:

$$L(x,\mu) = c^T x + \mu^T (A x - b) = (c^T + \mu^T A) x - \mu^T b = x^T (c + A^T \mu) - \mu^T b$$

e portanto $\nabla_x L(x^*, \mu^*) = c + A^T \mu^* = 0$, resulta em:

$$L(x^*, \mu^*) = -(\mu^*)^T b = S(x^*) = c^T x^*$$

Usando a definição de variável de folga, a função de Lagrange é dada por:

$$L(x,\mu) = c^T x - \mu^T f = x^T \eta - \mu^T b \quad , \quad \eta = c + A^T \mu$$

Consequentemente, ao obter a solução ótima pelo algoritmo simplex, os coeficientes das variáveis de folga na última linha da matriz aumentada correspondem aos multiplicadores de Lagrange das respectivas restrições. O sinal negativo na equação acima é compensado pelo fato de que no final do algoritmo tem-se do lado direito o termo $S - S(x^*)$. No exemplo acima $\mu_1 = \mu_2 = 1$. Vale lembrar que:

$$\nabla_b S(x) = -\mu^*$$

isto é, a sensibilidade da função objetivo em relação as restrições é dada pelo negativo do multiplicador de Lagrange. No exemplo dado, aumentando b_1 de 8 para 8,2 a função objetivo passaria de -18 para $-18 - 0,2 * \mu_1 = -18,2$. Esta constatação também pode ser obtida pelo fato de $S(x^*) = -(\mu^*)^T b$.

A formulação dual do problema de programação linear pode ser escrita como:

$$\max \phi(\mu) = -\mu^T b = -b^T \mu$$

$$\text{sujeito a: } -A^T \mu \leq c$$

$$\mu \geq 0$$

Onde $\phi(\mu) = \text{Inf}_x [L(x,\mu)] = -\mu^T b$. Este problema pode ser resolvido da mesma maneira que o problema primal, introduzindo as correspondentes variáveis de folga (η , definido acima).

Observe que o mínimo de $L(x,\mu)$ em relação a x existe se $\eta \geq 0$, cuja solução é $-\mu^T b$, pois pelas condições de complementaridade $x^T \eta = \mu^T f = 0$ (= *gap* dual). Ou seja, quando $A x^* = b \rightarrow (\mu^*)^T A x^* = (\mu^*)^T b = -c^T x^* \rightarrow [c^T + (\mu^*)^T A] x^* = 0$. Se $\eta < 0$, então o $\text{Inf}_x [L(x,\mu)] = -\infty$, pois $x \geq 0$. No caso de restrição de igualdade, a variável dual, λ , é irrestrita.

O número de soluções básicas possíveis nos problemas de programação linear primal e dual são dados pelas seguintes expressões, respectivamente:

$$N_p = \binom{p}{m} = \frac{p!}{m!(p-m)!} \quad N_d = \binom{p}{n} = \frac{p!}{n!(p-n)!}$$

sendo que o algoritmo simplex chega na solução ótima, geralmente, entre m e $3 \cdot m$ iterações para o problema primal (e entre n e $3 \cdot n$ para o dual), não necessariamente seguindo o caminho mais curto para o ótimo. O problema dual não requer uma solução inicial básica viável para x , mas para μ . Por outro lado, o problema primal sempre mantém a viabilidade das soluções para x a cada iteração.

5.2.1 Primeira solução viável

Se o ponto inicial não é viável, então é necessário realizar uma busca para a primeira solução viável. Isto pode ser feito com o uso de *variáveis artificiais*.

Exemplo 5.3: considere o problema

$$\begin{aligned} \min S(x_1, x_2) &= x_1 + 2 x_2 \\ \text{sujeito a} \quad &3 x_1 + 4 x_2 \geq 5 \\ &x_1 + x_2 \leq 4 \\ &x_1 \geq 0 \text{ e } x_2 \geq 0 \end{aligned}$$

Introduzindo as variáveis de folga f_1 e f_2 tem-se:

$$\begin{aligned} 3 x_1 + 4 x_2 - f_1 &= 5 \\ x_1 + x_2 + f_2 &= 4 \end{aligned}$$

e partindo do ponto inicial $x_1 = x_2 = 0$, resulta em $f_1 = -5$ e $f_2 = 4$, violando a restrição de não negatividade das variáveis. Definindo uma variável de folga artificial, v_1 , a restrição violada é reescrita como:

$$3 x_1 + 4 x_2 - f_1 + v_1 = 5$$

sendo satisfeita com valores positivos de f_1 e v_1 para $x = 0$ (e.g., $f_1 = 1$ e $v_1 = 6$). Deste modo uma solução viável é obtida quando as restrições forem satisfeitas e os valores das variáveis artificiais forem todos nulos. As variáveis de folga podem ser levadas a zero pela inclusão de uma penalização (Murty, 1983) na função objetivo do tipo:

$$S'(x, v) = S(x) + p^T v$$

onde $p_i \gg 0$ é a penalização da variável v_i . Ou pelo uso do algoritmo simplex em duas fases, sendo a primeira fase usando a seguinte função objetivo:

$$S'(v) = \sum_{i=1}^m v_i$$

e a segunda fase como descrita anteriormente a partir da solução básica obtida na primeira fase. Se o mínimo da primeira fase é diferente de zero então não há solução viável para o problema.

5.2.2 Forma padrão

Se um problema de otimização linear possuir alguma variável irrestrita, isto é, pode assumir valores positivos e negativos, então é possível transformá-lo na *forma padrão* de programação linear:

$$\begin{aligned} \min S(x) &= c^T x \\ \text{sujeito a: } A x &= b && \text{(forma padrão)} \\ x &\geq 0, \quad b \geq 0 \end{aligned}$$

pela substituição desta variável por outras duas variáveis restritas. Por exemplo, se um dado problema tem a variável x_1 irrestrita, então fazendo $x_1 = y_1 - z_1$, com $y_1 \geq 0$ e $z_1 \geq 0$, o problema equivalente com y_1 e z_1 no lugar de x_1 é restrito nestas variáveis. Se x possuir limite superior finito, estas restrições podem ser incluídas na matriz A com o uso das variáveis de folga. Por outro lado, se o limite inferior for diferente de zero ($x \geq w$), então faz-se uma mudança de variável do tipo $y = x - w$, com $y \geq 0$. Em suma, qualquer problema de programação linear pode ser posto na forma padrão. Observe que os problemas com restrições de desigualdade, tratados anteriormente, são transformados na forma padrão acima pela inclusão das variáveis de folga.

Quando uma linha, i , da matriz de restrições de igualdade possuir somente um elemento não nulo, o componente, j , de x associado a este valor é chamado de *singleton*, pois não pode assumir outro valor diferente de $x_j = b_i / a_{ij}$.

A etapa de eliminação Gaussiana do algoritmo simplex pode ser descrita em uma forma mais compacta através da seguinte notação matricial:

$$\begin{aligned} x^T &= [x_B \mid x_N] && , \quad x_B \in \mathfrak{R}^m \text{ (vetor básico)} \text{ e } x_N \in \mathfrak{R}^n \text{ (vetor não básico)} \\ A &= [B \mid N] && , \quad B \in \mathfrak{R}^{m \times m} \text{ (matriz base)} \text{ e } N \in \mathfrak{R}^{m \times n} \text{ (matriz não básica)} \\ c^T &= [c_B \mid c_N] && , \quad c_B \in \mathfrak{R}^m \text{ e } c_N \in \mathfrak{R}^n \text{ são os coeficientes da função objetivo} \end{aligned}$$

onde $n + m = p$ é o número total de variáveis do sistema. Usando esta notação a forma padrão é escrita como:

$$\begin{aligned} \min S(x) &= c_B^T x_B + c_N^T x_N \\ \text{sujeito a: } B x_B + N x_N &= b \\ x_B &\geq 0, \quad x_N \geq 0, \quad b \geq 0 \end{aligned}$$

Como a matriz base é de posto cheio, o vetor básico (das variáveis dependentes) é dado por:

$$x_B = B^{-1} b - B^{-1} N x_N$$

Esta operação transforma o problema na forma padrão para a sua *forma canônica*, ou seja:

$$\begin{aligned} \min S(x) &= c_B^T x_B + c_N^T x_N \\ \text{sujeito a: } & x_B + \bar{N} x_N = \bar{b} \quad (\text{forma canônica}) \\ & x_B \geq 0, x_N \geq 0, \bar{b} \geq 0 \end{aligned}$$

que é caracterizada por $x_B = \bar{b}$ quando $x_N = 0$. Substituindo x_B na função objetivo resulta em:

$$S(x) = c_B^T B^{-1} b + (c_N^T - c_B^T B^{-1} N) x_N$$

O termo $r_N = c_N - (B^{-1} N)^T c_B$, conhecido como vetor dos “custos relativos” ou gradiente reduzido da função objetivo ($\nabla_{x_N} S(x) = r_N$), é usado para determinar qual a variável não básica se tornará básica (determinação da coluna pivô). A variável que deixará a base resulta do teste da razão entre as linhas do termo independente $B^{-1} b$ e a coluna pivô da matriz $B^{-1} N$.

Fazendo $x_N = 0$ (como parte da solução básica viável), tem-se:

$$x_B = B^{-1} b \quad \text{e} \quad S(x) = c_B^T B^{-1} b$$

5.3 Método do ponto interior (Karmarkar)

O método de Karmarkar (1984) (ou Dikin, 1967), desenvolvido para tratar de problemas de dimensões elevadas (com um número elevado de vértices), determina direções de busca no interior estrito da região viável.

A tarefa mais difícil na solução do problema primal é determinar quais os componentes de x podem ser estritamente positivos em uma solução viável. Isto é equivalente a determinar quais as restrições do problema dual podem ser satisfeitas somente como restrições de igualdade (ou seja, satisfazer as condições de complementaridade). Os algoritmos de conjuntos ativos (*active set*), tal como o método simplex, atualizam repetitivamente um conjunto de no máximo m elementos de x que podem ser positivos a cada vez. Por outro lado, os algoritmos de ponto interior, tal como o de Karmarkar, trabalham com soluções viáveis onde todos os componentes de x são positivos. Eles repetitivamente fazem uma mudança de variável, calculam a direção de busca neste novo conjunto (em geral na direção de maior

descida ou conjugada), caminham alguma distância nesta direção e levam o ponto resultante para as variáveis originais.

Considerando o problema de programação linear na forma padrão:

$$\begin{aligned} \min S(x) &= c^T x \\ \text{sujeito a: } A x &= b \\ x &\geq 0 \end{aligned}$$

O algoritmo de Karmarkar inicia no centróide do simplex formado pelas restrições de igualdade $A x = b$, que é um ponto interior viável, x^0 (pontos próximos as fronteiras podem tornar lenta a convergência). A seguir o vetor direção da maior descida, dado por $d = -\nabla S(x) = -c$ é projetado na interseção das restrições de igualdade (que define o espaço nulo, ou núcleo da matriz A , isto é, o conjunto $\mathfrak{N} = \{d_p \in \mathfrak{R}^n \mid A d_p = 0\}$). A matriz de projeção é obtida sabendo-se que $d = d_p + d_r$, onde d_r é um vetor do espaço ortogonal ao espaço nulo (conhecido como “range space”), isto é, a imagem de A^T , ou seja $R = \{d_r \in \mathfrak{R}^n \mid d_r = A^T \mu, \mu \in \mathfrak{R}^m\} = \mathfrak{N}^\perp$. Portanto,

$$(d_r)^T d_p = 0 \quad \text{e} \quad d_p = d - d_r = d - A^T \mu \quad \rightarrow \quad \mu^T A (d - A^T \mu) = 0$$

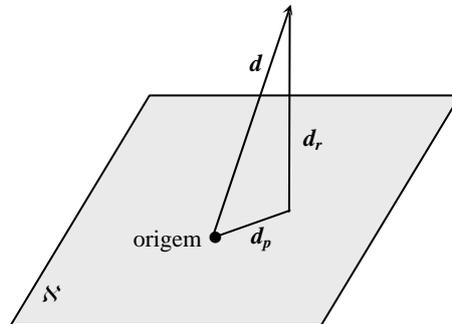
que resolvendo para μ resulta em:

$$\mu = (A A^T)^{-1} A^T d$$

e, deste modo, a direção projetada $d_p = P d$, é obtida pela seguinte matriz de projeção:

$$P = I - A^T (A A^T)^{-1} A$$

Esta projeção pode ser melhor interpretada ao analisar duas soluções viáveis durante a busca, x^k e x^{k+1} , onde $A x^k = b$ e $A x^{k+1} = b$, e portanto $A (x^{k+1} - x^k) = 0$, ou seja $(x^{k+1} - x^k) = \alpha_k d_p$, isto é, d_p é ortogonal as linhas da matriz A estando portanto no espaço nulo de A . A figura abaixo ilustra a projeção.



A busca segue então na direção projetada até as proximidades de uma restrição, obtendo o ponto x^k , que é normalizado por:

$$x^{k+1} = D^{-1} x^k$$

onde D é a matriz diagonal de x^k . Então o problema é reformulado em função desta matriz para que o ponto de partida do próximo estágio também esteja equidistante de todos os hiperplanos que formam o poliedro (ou seja, o centróide). Tem-se então:

$$A x^k = b = A D D^{-1} x^k = A D x^{k+1}$$

$$S(x) = c^T D D^{-1} x^k = c^T D x^{k+1}$$

resultando no novo problema:

$$\begin{aligned} \min S(x) &= c^T D x \\ \text{sujeito a: } &A D x = b \\ &x \geq 0 \end{aligned}$$

devendo-se proceder como no início do algoritmo, substituindo a matriz A por $A D$ e vetor c^T por $c^T D$. O critério de parada é verificado pela análise dos multiplicadores de Lagrange, calculados por:

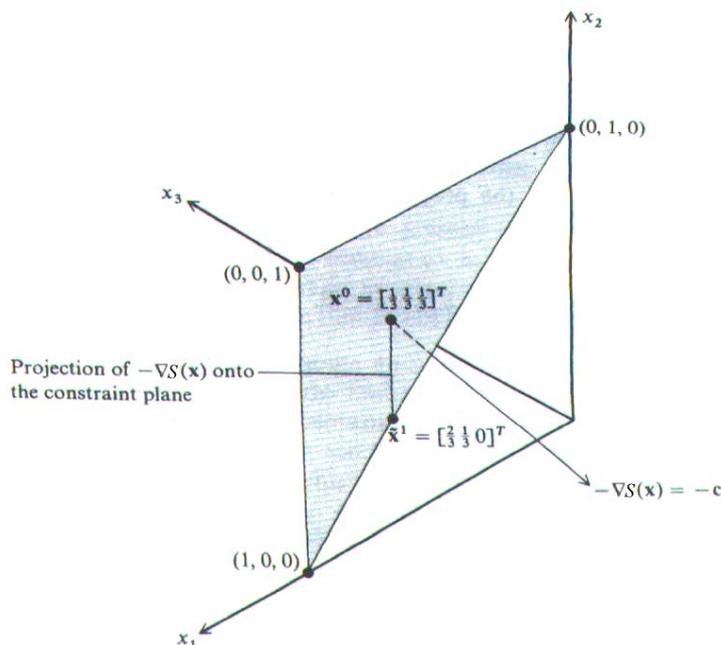
$$\mu = -(A A^T)^{-1} A c$$

e do gradiente projetado, isto é, na solução $P d = 0$.

Exemplo 5.4: considere o problema

$$\begin{aligned} \min S(x) &= x_1 + 2 x_2 + 3 x_3 \\ \text{sujeito a } &x_1 + x_2 + x_3 = 1 \\ &x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

ilustrado na figura abaixo, com centróide em $x^0 = [1/3 \ 1/3 \ 1/3]^T$ e $x^* = [1 \ 0 \ 0]^T$.



A primeira direção projetada é dada por: $d_p = P d = -P c = [1 \ 0 \ -1]^T$. O tamanho do passo nesta direção é obtido de modo a obter o maior avanço sem tornar a solução inviável, isto é:

$$\alpha^* = \gamma \times \max \{ \alpha > 0 \mid x^k + \alpha d_p \geq 0 \} = \gamma \times \min \{ -x_i^k / d_{pi} \mid d_{pi} < 0, i = 1, \dots, n \}$$

onde $0 < \gamma < 1$ é um *fator de segurança* para que os pontos permaneçam no interior da região viável (usualmente $\gamma \in [0,9 \ 0,9999]$). Para o exemplo $\alpha^* = \gamma \times 1/3$. Usando um valor $\gamma = 0,98$, tem-se o próximo ponto $x^1 = [1,98/3 \ 1/3 \ 0,02/3]^T$, ilustrado na figura acima pelo ponto da fronteira $[2/3 \ 1/3 \ 0]^T$. Portanto, a matriz de escalonamento é dada por $D = \text{diag}(1,98/3, \ 1/3, \ 0,02/3)$.

Existe uma grande variedades de implementações de algoritmos do ponto interior, tanto na forma primal quanto na dual, mantendo a idéia básica da projeção do gradiente no espaço nulo.

6. Programação Não Linear (NLP)

Os problemas de programação não linear podem ser representados pelas seguintes função objetivo e restrições não lineares:

$$\begin{aligned} \min S(x) \\ \text{sujeito a: } h_j(x) = 0, \quad j = 1, 2, \dots, m \\ g_j(x) \leq 0, \quad j = 1, 2, \dots, p \end{aligned}$$

onde $h_j(x)$ são m restrições de igualdade e $g_j(x)$ são p restrições de desigualdade.

A maioria dos algoritmos de programação não linear procuram linearizar localmente a fronteira da região viável, gerando restrições lineares. Alguns aproximam a função objetivo por funções quadráticas, outros por funções lineares. Portanto, os problemas de programação linear e quadrática são de grande importância para auxiliar na solução de problemas de programação não linear. O problema da programação linear já foi tratado no capítulo anterior. A seguir é discutido o problema de programação quadrática.

6.1 Programação Quadrática (QP)

O problema de programação quadrática é caracterizado por uma função objetivo quadrática e restrições lineares:

$$\begin{aligned} \min S(x) = c^T x + \frac{1}{2} x^T Q x \\ \text{sujeito a: } A x \leq b \\ x \geq 0 \end{aligned}$$

onde A é uma matriz $m \times n$, isto é, m restrições e n variáveis, c é um vetor de n coeficientes e Q é uma matriz simétrica $n \times n$. Vale lembrar que uma matriz quadrada, M , pode ser transformada em uma matriz simétrica usando a transformação $Q = (M + M^T) / 2$ e que $x^T M x = x^T Q x$.

O problema de programação quadrática geralmente surge como um sub-problema da programação não linear. Também ocorre em problemas de mínimos quadrados com restrição, controle ótimo de sistemas descritos por modelos lineares e função custo quadrática, controle preditivo linear, etc.

Sendo as restrições lineares e linearmente independentes, as condições de qualificação de segunda ordem das restrições são sempre satisfeitas. Portanto, as condições de primeira ordem de KKT são as condições necessárias para obter a solução ótima, ou seja:

$$\nabla_x L(x^*, \lambda^*, \mu^*) = \nabla S(x^*) + (\lambda^*)^T \nabla h(x^*) + (\mu^*)^T \nabla g(x^*) = 0$$

$$h(x^*) = 0$$

$$g(x^*) \leq 0$$

$$\mu_j^* g_j(x^*) = 0, j = 1, 2, \dots, p$$

$$\mu^* \geq 0$$

Além disto, se a matriz Q for positiva definida, as condições de segunda ordem de KKT são as condições suficientes para a solução ser um ponto extremo e o ótimo global do problema. Se Q não for positiva definida, então o problema pode não ter solução limitada ou apresentar mínimos locais.

A função de Lagrange para o problema de programação quadrática, na forma acima, é dada por:

$$L(x, \mu, \eta) = c^T x + \frac{1}{2} x^T Q x + \mu^T (A x - b) - \eta^T x$$

onde η são os multiplicadores associados ao limite inferior do vetor x (o sinal negativo surge porque a restrição é do tipo \geq). Portanto, o gradiente de L com relação a x fica:

$$\nabla_x L(x, \mu, \eta) = c + Q x + A^T \mu - \eta$$

Inserindo as variáveis de folga, f , e aplicando as condições KKT, resulta em:

$$A x + f - b = 0$$

$$c + Q x + A^T \mu - \eta = 0$$

$$\mu^T f = 0, \quad \eta^T x = 0 \quad (\text{ou } \mu^T f + \eta^T x = 0)$$

$$x \geq 0, f \geq 0, \mu \geq 0, \eta \geq 0$$

Multiplicando o segundo sistema de equações transposto por $x / 2$, tem-se:

$$\frac{1}{2} x^T Q x = \frac{1}{2} (\eta^T x - \mu^T A x - c^T x)$$

que substituindo na equação função objetivo leva a:

$$S(x, \mu, \eta, f) = \frac{1}{2} (c^T x - \mu^T A x + \eta^T x)$$

como $A x = b - f$, $\eta^T x = 0$ e $\mu^T f = 0$, tem-se então:

$$S(x, \mu, \eta, f) = \frac{1}{2} (c^T x - \mu^T b)$$

Fazendo o mesmo para a função de Lagrange tem-se:

$$L(x, \mu, \eta, f) = \frac{1}{2} (c^T x - \mu^T b) - \frac{1}{2} (\mu^T f + \eta^T x)$$

Portanto, o problema de programação quadrática pode ser resolvido pelo seguinte problema de programação linear, conhecido como *problema de complementaridade linear*:

$$\begin{aligned} \min S(\underline{x}) &= \underline{c}^T \underline{x} \\ \text{sujeito a: } \underline{A} \underline{x} &= \underline{b} \\ \underline{x} &\geq 0 \end{aligned}$$

onde $\underline{c}^T = \frac{1}{2} [c \mid 0_m \mid -b \mid 0_n]$

$$\underline{x}^T = [x \mid f \mid \mu \mid \eta] \in \mathfrak{R}^{2p}$$

$$\underline{b}^T = [b \mid -c]$$

$$\underline{A} = \left(\begin{array}{c|c|c|c} A & I_m & 0_m & 0_n \\ \hline Q & 0_m & A^T & -I_m \end{array} \right)$$

e observando a condição de complementaridade $\frac{1}{2} (\mu^T f + \eta^T x) = 0$ (= *gap* dual). Onde 0_m é um vetor nulo de tamanho m na definição de \underline{c} , e uma matriz nula de dimensão $m \times m$ na definição de \underline{A} . As condições de complementaridade implicam que no algoritmo simplex não é permitido que os pares (x_i, η_i) e (μ_i, f_i) sejam simultaneamente variáveis básicas em uma solução viável.

Observe que no problema acima, quando $Q = 0$, tem-se uma formulação primal-dual de um problema de programação linear.

Rescrevendo a função de Lagrange na seguinte forma, sem incluir a restrição $x \geq 0$ na função:

$$L(x, \mu) = (c^T + x^T Q + \mu^T A) x - \frac{1}{2} x^T Q x - b^T \mu$$

então o mínimo de $L(x, \mu)$ em relação a x existe se Q for pelo menos positiva semi-definida e se $\eta = c + Q x + A^T \mu \geq 0$, cuja solução é $-\frac{1}{2} x^T Q x - \mu^T b$, pois pelas condições de complementaridade $\eta^T x = 0$. Neste caso o problema dual pode ser formulado, sendo dado por:

$$\max \phi(\mu, z) = -\frac{1}{2} z^T Q z - b^T \mu$$

$$\text{sujeito a: } -A^T \mu - Q z \leq c$$

$$\mu \geq 0, \quad z \text{ irrestrito}$$

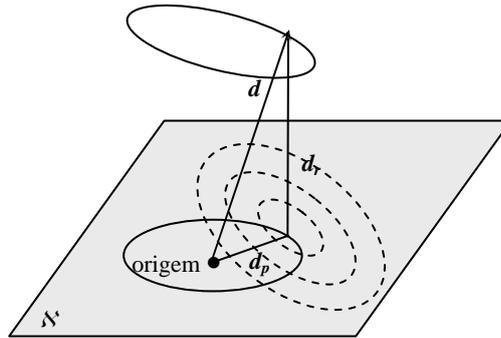
Do mesmo modo que na programação linear, também é possível resolver o problema de programação quadrática usando os métodos do ponto interior. Neste caso, a direção da maior descida, a matriz de projeção e os multiplicadores de Lagrange são dados por, respectivamente:

$$\begin{aligned} d &= -\nabla S(x^k) = -c - Q x^k \\ P &= Q^{-1} - Q^{-1} A^T (A Q^{-1} A^T)^{-1} A Q^{-1} \\ \mu &= Q^{-1} A^T (A Q^{-1} A^T)^{-1} A Q^{-1} d \end{aligned}$$

que resultam da solução das condições de otimalidade do problema:

$$\begin{aligned} \min S(x^k + d_p) &= S(x^k) + \nabla^T S(x^k) d_p + \frac{1}{2} (d_p)^T Q d_p \\ \text{sujeito a: } A d_p &= 0 \end{aligned}$$

isto é, da minimização da função objetivo sobre o espaço nulo, como ilustra abaixo, onde $d_p = P d$. A solução acima é possível se Q for positiva definida e A possuir posto completo.



Ou seja, ao aplicar a condição de gradiente nulo para a função de Lagrange do problema acima: $L(d_p, \mu) = S(x^k) + \nabla^T S(x^k) d_p + \frac{1}{2} (d_p)^T Q d_p + \mu^T A d_p$, tem-se:

$$\nabla_{d_p} L(d_p, \mu) = \nabla S(x^k) + Q d_p + A^T \mu = 0$$

$$\nabla_{\mu} L(d_p, \mu) = A d_p = 0$$

ou na forma matricial, usando $d = -\nabla S(x^k)$:

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d_p \\ \mu \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix}$$

cuja solução para μ e d_p é dada acima com o uso da matriz de projeção.

6.2 Relaxação Lagrangeana

O método da relaxação Lagrangeana consiste em utilizar os multiplicadores de Lagrange para a solução de problemas convexos, transformando o problema:

$$\begin{aligned} & \min_{x \in K} S(x) \\ \text{sujeito a: } & h_j(x) = 0, \quad j = 1, 2, \dots, m \\ & g_j(x) \leq 0, \quad j = 1, 2, \dots, p \end{aligned}$$

onde $S(x)$, $h_j(x)$ e $g_j(x)$ são funções convexas e K é um convexo, no problema:

$$\begin{aligned} & \max_{\lambda \in \mathcal{R}^m, \mu \in \mathcal{R}^p} \phi(\lambda, \mu) \\ \text{sujeito a: } & \mu \geq 0 \end{aligned}$$

onde $\phi(\lambda, \mu) = \min_{x \in K} \{S(x) + \lambda^T h(x) + \mu^T f(x)\}$, $f_j(x) = g_j(x) + (v_j)^2$ e v_j ($j = 1, 2, \dots, p$)

são as variáveis de folga. Isto é, transforma em um problema sem restrições.

Exemplo 6.1: $\min S(x) = x_1 x_2$

$$\text{sujeito a: } g_1(x) = x_1^2 + x_2^2 - 25 \leq 0$$

$$L(x, \mu) = x_1 x_2 + \mu_1 (x_1^2 + x_2^2 - 25 + v_1^2)$$

$$\frac{\partial L}{\partial x_1} = x_2 + 2\mu_1 x_1 = 0$$

$$\frac{\partial L}{\partial x_2} = x_1 + 2\mu_1 x_2 = 0$$

$$\frac{\partial L}{\partial \mu_1} = x_1^2 + x_2^2 - 25 + v_1^2 = 0$$

$$\frac{\partial L}{\partial v_1} = 2\mu_1 v_1 = 0$$

μ_1	x_1	x_2	v_1	$S(x)$	ponto
0	0	0	5	0	sela
0.5	3,54	-3,54	0	-12,5	mínimo
0,5	-3,54	3,54	0	-12,5	mínimo
-0,5	3,54	3,54	0	12,5	máximo
-0,5	-3,54	-3,54	0	12,5	máximo

Observe que como $\frac{\partial L}{\partial v_j} = 2\mu_j v_j = 0$ (isto é, as condições de complementaridade das restrições de desigualdade) implica em $\mu_j = 0$ ou $v_j = 0$, então para o caso de $v_j = 0$, tem-se a solução sobre a j -ésima restrição de desigualdade, e quando $\mu_j = 0$, a solução está distante da restrição pela variável de folga v_j . Observe também que o ponto ótimo de $S(x)$ corresponde a um ponto sela de sua correspondente função de Lagrange (isto é, um ponto de mínimo em relação a x e de máximo em relação aos multiplicadores, também conhecido como problema *minimax*).

O uso das variáveis de folga na forma $(v_j)^2$ nas restrições de desigualdade evita a necessidade de restringir $v \geq 0$, como foi feito para os casos da programação linear e quadrática.

Exemplo 6.2: $\min S(x) = 4x_1^2 + 5x_2^2$
 sujeito a: $h_1(x) = 2x_1 + 3x_2 - 6 = 0$

$$L(x, \lambda) = 4x_1^2 + 5x_2^2 + \lambda_1(2x_1 + 3x_2 - 6)$$

$$\frac{\partial L}{\partial x_1} = 8x_1 + 2\lambda_1 = 0$$

$$\frac{\partial L}{\partial x_2} = 10x_2 + 3\lambda_1 = 0$$

$$\frac{\partial L}{\partial \lambda_1} = 2x_1 + 3x_2 - 6 = 0$$

A solução deste sistema de equações leva a $x_1 = 1,071$, $x_2 = 1,286$, $\lambda_1 = -4,286$.

Em suma, a solução do método da relaxação Lagrangeana (ou método de Lagrange) consiste em aplicar as condições de otimalidade sobre $L(x, \lambda, \mu)$, ou seja:

$$\nabla_x L(x, \lambda, \mu) = \nabla S(x) + \lambda^T \nabla h(x) + \mu^T \nabla g(x) = 0$$

$$h(x) = 0$$

$$g_j(x) + (v_j)^2 = 0$$

$$\mu_j v_j = 0, \quad j = 1, 2, \dots, p$$

$$\mu \geq 0$$

Como o conjunto resultante de equações forma, geralmente, um sistema não linear de equações algébricas, então o método da relaxação Lagrangeana não é muito atrativo, pois a solução deste sistema não linear pode ser mais complexa que a solução direta do problema de otimização por outros métodos. Além disto, a solução somente está

garantida se as condições de convexidade forem satisfeitas. Caso contrário, os testes de otimalidade de segunda ordem devem ser verificados, isto é, a matriz Hessiana da função de Lagrange com relação a x , $\nabla_x^2 L(x^*, \lambda^*, \mu^*)$, deve ser pelo menos positiva semi-definida para todo vetor não nulo d tal que:

$$d^T \nabla h_i(x^*) = 0, \quad i = 1, 2, \dots, m$$

$$d^T \nabla g_j(x^*) = 0 \text{ para as } g_j(x^*) \text{ ativas } \{v_j^* = 0 \text{ e } \mu_j^* > 0\}$$

$$d^T \nabla g_j(x^*) \leq 0 \text{ para as } g_j(x^*) \text{ inativas } \{v_j^* \neq 0 \text{ e } \mu_j^* = 0\}$$

isto é, $d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d \geq 0$, ou se todas as raízes do polinômio característico:

$$p(\lambda) = \begin{vmatrix} \lambda I - \nabla_x^2 L & M \\ M^T & 0 \end{vmatrix} = 0$$

forem não negativas, onde M é a matriz formada pelos gradientes de $h(x^*)$ e $g(x^*)$ ativas, isto é, a matriz tal que $M^T d = 0$ (espaço nulo de M^T), com $m+p^a < n$ e com posto completo (p^a é o número de restrições g ativas). Observe que $M^T = A$ para os casos da programação linear e quadrática.

Para a segunda solução do exemplo 6.1 a matriz $M = \nabla g_1(x^*) = 2 [x_1 \ x_2]^T = [7,08 \ -7,08]^T$ e a matriz Hessiana é dada por:

$$\nabla_x^2 L = \begin{bmatrix} 2\mu & 1 \\ 1 & 2\mu \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

resultando em $d^T \nabla g_1(x^*) = 7,08 (d_1 - d_2) = 0 \rightarrow d_1 = d_2$ e $d^T \nabla_x^2 L d = 4d_1^2 > 0$, e portanto o ponto é um mínimo local, ou de forma equivalente $p(\lambda) = 0$ resulta no valor característico $\lambda = 2$. Para o exemplo 6.2, que é convexo, a solução obtida é o mínimo global do problema.

Apesar da dificuldade da aplicação do método de Lagrange para a solução de problemas não lineares, ele é muito útil para analisar a sensibilidade da função objetivo em relação as restrições.

A aplicação do método do gradiente para a solução da relaxação Lagrangeana no espaço dual, quando há somente restrições de desigualdade, sem o uso das variáveis de folga, resulta no seguinte algoritmo (Uzawa). Lembrando que se μ^* maximiza $\phi(\mu)$ sobre $\mu \geq 0$, então o correspondente x^* é solução do problema primal.

algoritmo

- 1) Escolher um ponto inicial μ^0 , $k = 0$
- 2) Resolver o problema $\phi(\mu^k) = \min_{x \in K} \{S(x) + (\mu^k)^T g(x)\}$, obtendo x^k .
- 3) $d_i^k = \begin{cases} g_i(x^k) & \text{se } \mu_i^k > 0 \\ \max\{0, g_i(x^k)\} & \text{se } \mu_i^k = 0 \end{cases} \quad i = 1, 2, \dots, p$
- 4) Encontrar α_k tal que $\phi(\mu^k + \alpha_k d^k) = \max_{\alpha > 0} \phi(\mu^k + \alpha d^k)$
- 5) Calcular $\mu^{k+1} = \mu^k + \alpha_k d^k$
- 6) **Se** o critério de convergência não foi satisfeito, **então** $k \leftarrow k + 1$ (ir para 2)
- 7) FIM.

Naturalmente, assim como foi utilizado o método do gradiente, qualquer outro método de otimização sem restrição poderia ser utilizado para resolver este problema.

6.3 Funções penalidade

Transforma o problema:

$$\min_{x \in K} S(x)$$

sujeito a: $h_j(x) = 0, j = 1, 2, \dots, m$

$g_j(x) \leq 0, j = 1, 2, \dots, p$

no problema:

$$\min_{x \in \mathbb{R}^n} [S(x) + P(x, r)] \quad , \quad \text{otimização sem restrição}$$

onde $P(x, r)$ é a função penalidade e $r \geq 0$. A função penalidade deve satisfazer as seguintes propriedades:

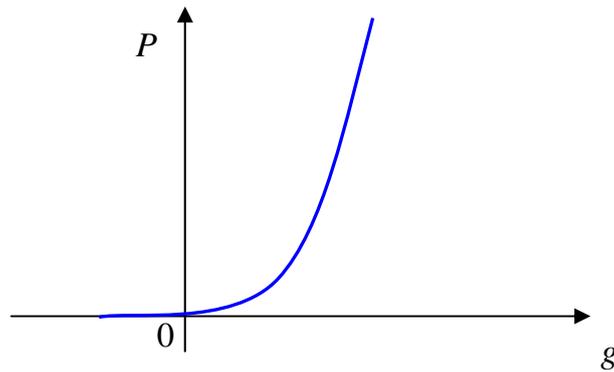
$$P(x, r) \geq 0 \quad \forall x$$

$$P(x, r) = 0 \quad \text{para todo } x \text{ viável}$$

O caso mais comum de funções penalidades é a penalidade quadrática:

$$P(x, r) = r \left\{ \sum_{j=1}^m (h_j(x))^2 + \sum_{j=1}^p [\max(0, g_j(x))]^2 \right\}$$

Esta função, ilustrada abaixo, causa uma descontinuidade da matriz Hessiana quando algum $g_j(x) = 0$, dificultando o uso de algoritmos quasi-Newton.



Além disto, o condicionamento do problema, que é responsável pelo desempenho de qualquer algoritmo de descida, piora quando r cresce. Um aumento de r gera uma melhor aproximação da fronteira da região viável, mas torna-se mais difícil satisfazer as condições de KKT. Uma estimativa para os multiplicadores de Lagrange (e Kuhn-Tucker) é dada por:

$$\lambda = r h[x(r)]$$

$$\mu = r \max\{0, g[x(r)]\}$$

Uma estratégia possível, para minimizar estes problemas, seria um auto-ajuste do parâmetro r , iniciando com valores baixos e se $P(x^k, r^k) > \varepsilon$ então fazer $r^{k+1} = \gamma r^k$ com $\gamma > 1$. O fator de escala r pode também ser individualizado para cada restrição, em função da importância ou da ordem de grandeza da mesma.

Outra função penalidade usada, não suave, é dada por:

$$P(x, r) = r \left\{ \sum_{j=1}^m |h_j(x)| + \sum_{j=1}^p \max(0, g_j(x)) \right\}$$

sendo uma penalidade exata, pois existe um valor finito de r , r_{min} , tal que x^* é um mínimo local de $\{S(x) + P(x, r)\}$ quando $r > r_{min}$, ao contrário da penalidade quadrática onde r deve ser infinito para se ter uma penalidade exata.

No caso de restrições de desigualdades é possível usar penalidades interiores à região viável, conhecidas como *funções barreiras*, sendo a forma mais comum dada por:

$$B(x, r) = -\sum_{i=1}^p \frac{1}{r g_i(x)}$$

sendo as soluções sucessivas estritamente viáveis. Obviamente, as dificuldades computacionais aumentam quando se aproxima da fronteira.

Uma penalização mais satisfatória e exata, no sentido de que existe um valor finito de r , acima do qual a solução do problema modificado satisfaz as condições de

otimalidade do problema original, é o *método do Lagrangeano aumentado* (Powell, 1969 e Hestenes, 1969). Neste caso a função de Lagrange toma a forma:

$$L_A(x, \lambda; r) = S(x) + \lambda^T h(x) + \mu^T f(x) + P(x, r)$$

onde $f_j(x) = g_j(x) + (v_j)^2$, v_j ($j = 1, 2, \dots, p$)

$$P(x, r) = \frac{1}{2} r \{h^T(x) h(x) + f^T(x) f(x)\}$$

Neste caso, a parte da matriz Hessiana de L_A , correspondente a função penalidade é dada por:

$$r \{A^T(x)A(x) + Q(x)\}$$

onde $A = \nabla^T w(x)$, $w^T = [h^T(x) \quad f^T(x)]$, $Q(x) = \sum_{i=1}^{m+p} w_i(x) H_i(x)$ e $H_i(x)$ é a matriz

Hessiana da função $w_i(x)$. Como $w_i(x^*) = 0$, então $Q(x^*) = 0$, reduzindo a Hessiana da função penalidade para $r A^T(x^*)A(x^*)$, que é uma matriz positiva semi-definida. Portanto, se um vetor d pertence ao núcleo de $A(x^*)$, isto é, $A(x^*) d = 0$, então:

$$d^T \nabla_x^2 L_A(x^*, \lambda^*; r) d = d^T \nabla_x^2 L(x^*, \lambda^*) d + r d^T A^T(x^*)A(x^*) d = d^T \nabla_x^2 L(x^*, \lambda^*) d \geq 0$$

E se d pertence ao *range space* de $A(x^*)$, isto é, $d = A^T(x^*) \mu$, então:

$$d^T \nabla_x^2 L_A(x^*, \lambda^*; r) d = d^T \nabla_x^2 L(x^*, \lambda^*) d + r \mu^T A(x^*)A^T(x^*) A(x^*)A^T(x^*) \mu \geq 0$$

se r for suficientemente grande, aumentando os valores característicos de $\nabla_x^2 L(x^*, \lambda^*)$ no *range space* de $A(x^*)$, que podem ser negativos, deixando os demais inalterados. Logo, se o problema original possuir uma matriz Hessiana positiva semi-definida (ou definida), então existe um r finito tal que valores acima dele resultarão em uma matriz Hessiana aumentada positiva semi-definida (ou definida), o que implica que x^* minimiza $L_A(x, \lambda^*; r)$.

No caso de usar um vetor de pesos, r , ao invés de um escalar, a análise acima também se aplica, bastando trocar a expressão $r A^T(x^*)A(x^*)$ por $A^T(x^*) R A(x^*)$, onde $R = \text{diag}(r_i)$.

Exemplo 6.3: $\min S(x) = x^3$

$$\text{sujeito a: } h(x) = x + 1 = 0$$

Obviamente a solução deste problema é $x^* = -1$. O valor ótimo do multiplicador de Lagrange pode ser obtido por:

$$\nabla_x L(x, \lambda) = 3x^2 + \lambda = 0 \rightarrow \lambda^* = -3(x^*)^2 = -3.$$

Definindo o Lagrangeano aumentado:

$$L_A(x, \lambda; r) = x^3 + \lambda(x + 1) + r(x + 1)^2 / 2$$

tem-se os seguintes resultados:

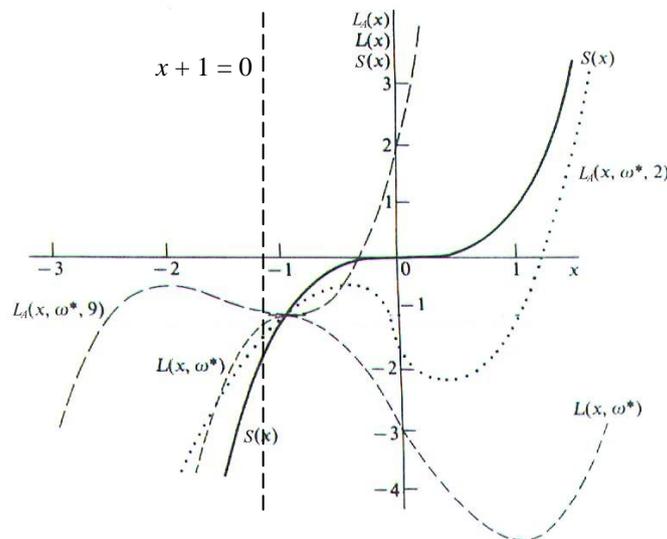
$$\nabla_x L_A(x^*, \lambda^*; r) = 3(x^*)^2 + \lambda^* + r(x^* + 1) = 0$$

$$\nabla_x^2 L(x^*, \lambda^*) = 6x^* = -6 \text{ (negativa definida)}$$

$$d^T \nabla h(x^*) = 0 \rightarrow d = 0 \text{ (} m = n, \text{ grau de liberdade nulo)}$$

$$\nabla_x^2 L_A(x^*, \lambda^*; r) = 6x^* + r = r - 6 \text{ (positiva definida se } r > 6\text{)}.$$

A figura abaixo ilustra este exemplo, fixando $\lambda^* = -3$ e usando dois valores diferentes para r ($r = 2 < 6$ e $r = 9 > 6$). Observe que para $x^* = -1$ o valor de $L(x, \lambda^*)$ é máximo, e é onde o valor de $S(x)$ intercepta $L(x, \lambda^*)$. Também neste ponto a função $L_A(x, \lambda^*; 9)$ apresenta um mínimo local. O ínfimo de $L_A(x, \lambda^*; 9) = -\infty$. Observe também que quando $r < 6$ o mínimo de $L_A(x, \lambda^*; r)$ é deslocado para um ponto inviável.



6.4 Programação linear seqüencial (SLP)

A técnica mais direta para resolver problema de programação não linear em geral é a linearização do problema e a aplicação de técnicas de programação linear, tais como:

1. Linearizar o problema (função objetivo e restrições) em torno de um ponto nominal de operação e obter a solução do problema de programação linear resultante;
2. Linearizar sucessivamente o problema não linear na medida em que melhores soluções viáveis são obtidas por métodos de programação linear. No caso de obter uma solução inviável na programação linear, então é localizado um ponto viável próximo a este;

3. Linearizar por partes a função objetivo e as restrições gerando um problema aproximado, formado por séries de segmentos de reta.

Contudo, não há garantias de convergência para estes métodos. A segunda estratégia é a base do método da programação linear seqüencial (SLP - *Sequential Linear Programming*), onde a cada estágio um problema de programação linear é resolvido pelo método simplex. Este método também é conhecido como método da programação aproximada (MAP - *Method of Approximate Programming*), denominado por Griffith & Stewart, 1961.

A linearização do problema de programação não linear em torno de um ponto x^k resulta em:

$$\min_{x \in K_L} S(x^k) + \nabla^T S(x^k) (x - x^k) \approx S(x)$$

$$\text{sujeito a: } h_j(x) \approx h_j(x^k) + \nabla^T h_j(x^k) (x - x^k) = 0, \quad j = 1, 2, \dots, m$$

$$g_j(x) \approx g_j(x^k) + \nabla^T g_j(x^k) (x - x^k) \leq 0, \quad j = 1, 2, \dots, p$$

sendo $(x - x^k)$ arbitrariamente restrito por limites inferior e superior para que os pontos permaneçam em uma vizinhança próxima a x^k .

algoritmo

- 1) Escolher um ponto inicial x^0 viável e seus limites L^0 e U^0 , $k = 0$
- 2) Linearizar a função objetivo e as restrições em torno do ponto x^k
- 3) Resolver o problema de programação linear resultante
- 4) **Se** x^{k+1} (solução do LP) não for viável para o NLP, **então** reduzir o intervalo $[L^k, U^k]$ para $[L^{k+1}, U^{k+1}]$ e ir para (3)
- 5) **Se** o critério de convergência não foi satisfeito, **então** $k \leftarrow k + 1$ (ir para 2)
- 6) FIM.

As maiores desvantagens do SLP são:

- a) pode convergir lentamente se o ótimo estiver no interior da região viável, ou se o problema for muito não linear;
- b) geralmente viola as restrições não lineares por uma quantidade razoável, necessitando de muitas iterações para a redução dos intervalos.

Por outro lado, em problemas de grandes dimensões e não linearidades moderadas, o método SLP pode ser competitivo. A convergência pode ser rápida em problemas cuja solução está em um vértice das restrições.

6.5 Gradientes reduzidos generalizados (GRG)

O método dos gradientes reduzidos generalizados tem se mostrado o melhor método de linearizações sucessivas. É uma extensão do método de Wolfe (1962), aplicado a problemas com restrições lineares, para acomodar restrições não lineares. Em essência, o método define novas variáveis que são normais às restrições lineares ou linearizadas, e expressa o gradiente (ou outra direção de busca) em termos desta base normal.

O problema não linear, para ser resolvido pelo método GRG, deve estar na forma padrão, ou seja:

$$\begin{aligned} \min \quad & S(x) \\ \text{sujeito a:} \quad & h_j(x) = 0, \quad j = 1, 2, \dots, m \\ & L_i \leq x \leq U_i, \quad i = 1, 2, \dots, n \end{aligned}$$

onde as restrições de desigualdade estão incluídas em $h(x)$ pelo uso das variáveis de folga:

$$\begin{aligned} h_j(x) &= g_j(x) + (v_j)^2, \quad -\infty \leq v_j \leq \infty \quad \text{ou} \\ h_j(x) &= g_j(x) + v_j, \quad 0 \leq v_j \end{aligned}$$

ou ainda usando a estratégia dos conjuntos ativos (*active set*), onde a cada estágio de busca as restrições ativas são adicionadas ao conjunto das restrições de igualdade e as inativas são removidas, baseado nas estimativas dos multiplicadores de Lagrange.

Lembrando que o vetor direção d pode ser decomposto em $d = d_r + d_p$, onde d_p é um vetor do espaço nulo (\aleph) da matriz $A = \nabla^T h$, d_r é um vetor do “range space” (R) de A e $(d_p)^T d_r = 0$, ou seja, d_p é a projeção ortogonal de d em \aleph . Portanto, qualquer vetor $d \in \mathfrak{R}^n$ pode ser representado unicamente por uma combinação linear da base Y de R e da base Z de \aleph :

$$d = Y d_r + Z d_p$$

onde Y é uma matriz com m colunas independentes e Z é uma matriz com $n-m$ colunas independentes e ortogonais a Y , tendo ambas n linhas.

Por exemplo, se x é um ponto viável de restrições lineares (ou linearizadas), $Ax = b$, a sua representação nas base acima: $x = Y x_r + Z x_p$, resulta em:

$$A x = A Y x_r + A Z x_p = b$$

mas como $A Z = 0$, tem-se $A Y x_r = b$ e portanto:

$$x_r = (A Y)^{-1} b$$

ou seja, todos os pontos viáveis possuem os mesmos componentes no “*range space*” de A e a solução do problema de otimização é determinada somente pelos componentes de x no espaço nulo de A . Em outras palavras, partindo de um ponto viável, x^k , o vetor direção deve permanecer no espaço nulo:

$$x^{k+1} = x^k + \alpha_k d^k \quad , \quad \text{com } A d^k = 0$$

ou seja $d^k = Z d_p^k$ ($Y d_r^k = 0$). Então, o vetor direção tem dimensão reduzida $n-m$.

No caso das programações linear e quadrática tem-se, respectivamente:

$$Z^T = P = I - A^T (A A^T)^{-1} A$$

$$Z^T = P = Q^{-1} - Q^{-1} A^T (A Q^{-1} A^T)^{-1} A Q^{-1}$$

Observe que $Z^T = Z^{-1}$, pois a base é ortogonal.

No método GRG a matriz Z é obtida pela seguinte forma:

$$Z = \begin{bmatrix} -B^{-1} C \\ I_{n-m} \end{bmatrix}$$

onde B é uma matriz não singular de dimensão $m \times m$ e C é uma matriz $m \times (n-m)$, ambas com colunas independentes, obtidas da decomposição da matriz A :

$$A = [B \mid C]$$

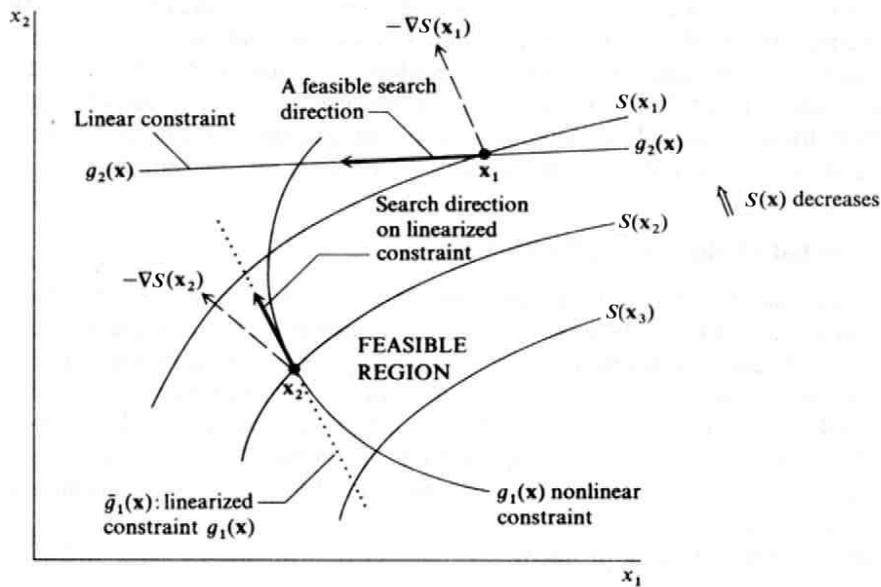
Do mesmo modo o vetor x é decomposto em $x^T = [x_D \mid x_I]$, onde $B = \nabla_{x_D}^T h(x^k)$ e $C = \nabla_{x_I}^T h(x^k)$. Deste modo, $x_D = B^{-1} b - B^{-1} C x_I$, que é conhecido como vetor das variáveis dependentes (ou vetor básico) e x_I é o vetor das variáveis independentes (ou vetor superbásico). Aquelas variáveis mantidas fixas em um de seus limites são chamadas de variáveis não básicas.

O termo gradiente reduzido, g_R , é dado ao seguinte vetor:

$$g_R(x^k) = Z^T \nabla S(x^k) = \begin{bmatrix} -B^{-1} C \\ I_{n-m} \end{bmatrix}^T \begin{bmatrix} \nabla_{x_D} S(x^k) \\ \nabla_{x_I} S(x^k) \end{bmatrix} = \nabla_I S(x^k) - C^T (B^{-1})^T \nabla_D S(x^k)$$

onde $\nabla_D S(x^k) = \nabla_{x_D} S(x^k)$ (para simplificar a notação), sendo que o vetor direção de busca é dado por $-g_R$ nas direções que não violam os limites das variáveis.

Naturalmente, como as restrições foram linearizadas, a direção de busca é viável localmente, como ilustra a figura abaixo.



e a busca pode parar em um ponto inviável, em relação as restrições originais.

A relação entre os multiplicadores de Lagrange e os gradientes reduzidos pode ser obtida da seguinte forma:

$$L(x, \lambda) = S(x) + \lambda^T h(x)$$

$$\nabla_D L(x, \lambda) = \nabla_D S(x) + \nabla_D h(x) \lambda = \nabla_D S(x) + B^T \lambda = 0$$

$$\nabla_I L(x, \lambda) = \nabla_I S(x) + \nabla_I h(x) \lambda = \nabla_I S(x) + C^T \lambda = 0$$

então,

$$\lambda = -(B^{-1})^T \nabla_D S(x)$$

e ao substituí-lo em $\nabla_I L(x, \lambda)$, resulta em:

$$\nabla_I L(x, \lambda) = \nabla_I S(x) - \nabla_I h(x) \lambda = \nabla_I S(x) - C^T (B^{-1})^T \nabla_D S(x) = g_R(x) = 0$$

quando x for um ponto ótimo.

Exemplo 6.4: $\min S(x) = 2x_1^2 + 2x_2^2$

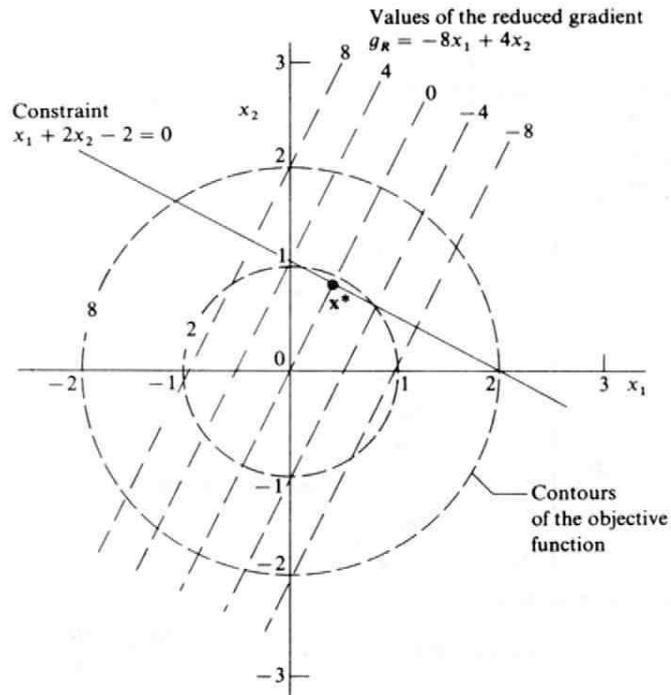
$$\text{sujeito a: } h_1(x) = x_1 + 2x_2 - 2 = 0$$

arbitrando $x_D = x_1$ e $x_I = x_2$, tem-se:

$$B = \frac{\partial h}{\partial x_D} = 1 \quad C = \frac{\partial h}{\partial x_I} = 2$$

$$\nabla_D S(x) = 4x_1 \quad \nabla_I S(x) = 4x_2$$

portanto, $g_R(x) = \nabla_I S(x) - C^T (B^{-1})^T \nabla_D S(x) = 4x_2 - 8x_1$, que pode ser visualizado na figura abaixo.



algoritmo

- 1) Escolher um ponto x^0 viável e selecionar $n-m$ variáveis independentes, $k = 0$
- 2) Linearizar a função objetivo e as restrições em torno do ponto x^k
- 3) Calcular g_R^k

4) Determinar a direção de busca no espaço das variáveis independentes:

se $x_{I,i}^k = U_i$ e $g_{R,i}^k < 0$ ou $x_{I,i}^k = L_i$ e $g_{R,i}^k > 0$, **então** $d_{I,i}^k = 0$

senão $d_{I,i}^k = -g_{R,i}^k$

5) Determinar a direção de busca no espaço das variáveis dependentes:

$$d_D^k = -(B^k)^{-1} C^k d_I^k$$

(isto é, $dh = \frac{\partial h}{\partial x_D} dx_D + \frac{\partial h}{\partial x_I} dx_I = 0$)

6) Encontrar α_k tal que $S(x^k + \alpha_k d^k) = \min_{\alpha > 0} g_k(\alpha) = S(x^k + \alpha d^k)$, sujeito a $L \leq x \leq U$

onde $d^k = \begin{bmatrix} d_D^k \\ d_I^k \end{bmatrix}$

7) Calcular $x_I^{k+1} = x_I^k + \alpha_k d_I^k$

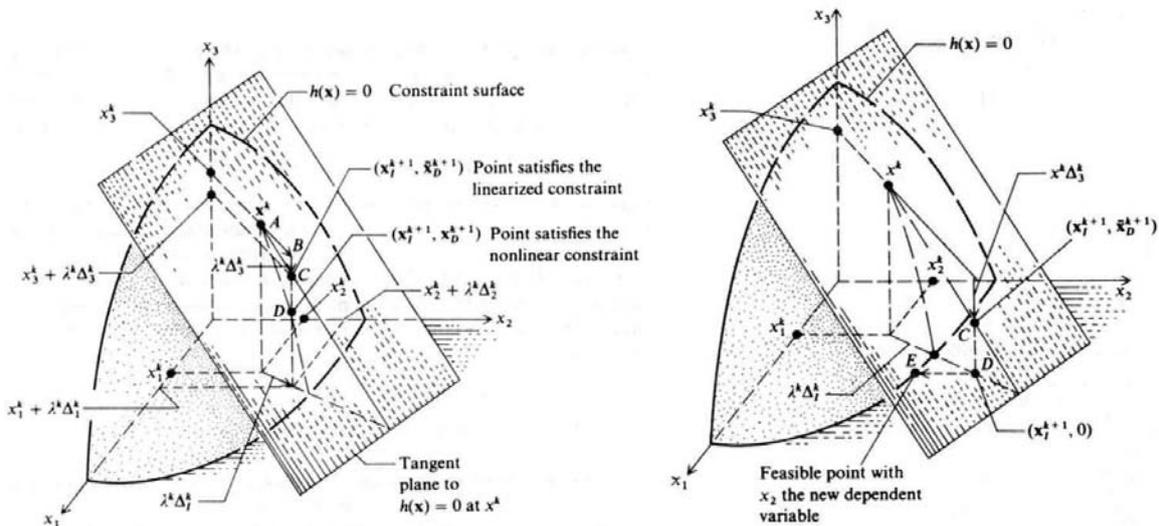
$$\tilde{x}_D^{k+1} = x_D^k + \alpha_k d_D^k \text{ (ponto tentativa ou predição)}$$

- 8) Usar o método de Newton para tornar as variáveis dependentes em um ponto viável $h(x_D^{k+1}, x_I^{k+1}) = 0$ (correção), usando \tilde{x}_D^{k+1} como estimativa inicial.
- 9) Se x^{k+1} é viável e $S(x^{k+1}) \leq S(x^k)$, então $k \leftarrow k + 1$ e se o critério de convergência não foi satisfeito (ir para 2) **senão** FIM.
- 10) Se x^{k+1} é viável e $S(x^{k+1}) > S(x^k)$ ou se o método de Newton não convergiu, então reduzir α_k e (ir para 7).

NOTA: se a obtenção de um ponto viável x^{k+1} não foi possível, após um determinado número de iterações, então deve-se trocar alguns elementos da base e reiniciar o algoritmo. Também aquelas variáveis independentes que atingirem seus limites devem ser substituídas por outras variáveis dependentes. A determinação da direção de busca no espaço das variáveis independentes pode também ser obtida pelo método dos gradientes conjugados:

$$d_I^{k+1} = -g_R^{k+1} + d_I^k \frac{(g_R^{k+1})^T g_R^{k+1}}{(g_R^k)^T g_R^k}$$

As figuras abaixo ilustram o procedimento de predição-correção. No primeiro caso a correção foi bem sucedida, mas no segundo foi necessário mudar de base. As variáveis independentes na iteração k são x_1 e x_2 . Do ponto A para o ponto B tem-se a correção somente nas variáveis independentes. O ponto C é a predição da variável dependente (retorno ao plano tangente, ou restrições linearizadas). O ponto D corresponde a correção da variável dependente para satisfazer as restrições originais. No caso da segunda figura, as correções de Newton não levam o ponto C para dentro das restrições. Então no ponto D (fora das restrições) é feita uma mudança de base tornando x_2 como variável dependente e x_3 como variável independente, sendo que uma nova correção leva o ponto D para o ponto E, que satisfaz as restrições.



6.6 Programação quadrática seqüencial (SQP)

Programação quadrática seqüencial, ou sucessiva, ou recursiva, ou iterativa, ou ainda métodos da métrica variável com restrição, são as várias formas de referenciar o método SQP (Wilson, 1963), que resume-se, basicamente, a resolver as equações de Karush-Kuhn-Tucker (KKT), ou condições necessárias de primeira ordem. A idéia é chegar o mais próximo possível do método de Newton empregado na solução de problemas sem restrição, que apresenta convergência quadrática.

Ao nível mais externo das iterações são feitas aproximações da Hessiana da função de Lagrange, usando métodos de atualização tipo quasi-Newton. O problema de programação quadrática resultante, associado a uma busca em linha (*linesearch*) ou a uma técnica de região de confiança (*trust region*), é então resolvido para prover uma direção de busca.

Aplicando as condições de KKT ao problema de programação não linear:

$$\begin{aligned} & \min_{x \in K} S(x) \\ \text{sujeito a: } & h_j(x) = 0, \quad j = 1, 2, \dots, m \\ & g_j(x) \leq 0, \quad j = 1, 2, \dots, p \end{aligned}$$

com o uso da função de Lagrange:

$$L(x, \lambda, \mu) = S(x) + \lambda^T h(x) + \mu^T g(x)$$

chega-se no seguinte conjunto de equações:

$$\begin{aligned} \nabla_x L(x, \lambda, \mu) &= \nabla S(x) + \lambda^T \nabla h(x) + \mu^T \nabla g(x) = 0 \\ h(x) &= 0 \\ g_j(x) &= 0, \quad j \in \{\text{restrições ativa}\} \\ \mu_i &\geq 0, \quad i \in \{\text{restrições inativas}\} \end{aligned}$$

que é transformado nos sucessivos problemas de programação quadrática:

$$\begin{aligned} & \min_{d \in \mathbb{R}^n} q(d) = \nabla^T S(x^k) d + \frac{1}{2} d^T H(x^k, \lambda^k, \mu^k) d \\ \text{sujeito a: } & h(x^k) + \nabla^T h(x^k) d = 0 \\ & g(x^k) + \nabla^T g(x^k) d \leq 0 \end{aligned}$$

para determinar a melhor direção de busca a partir do ponto x^k e então proceder a atualização para o próximo ponto $x^{k+1} = x^k + \alpha_k d^k$. Assim como no método de Newton, apesar do valor ótimo do passo ser $\alpha_k = 1$, valores de $\alpha_k \in (0, 1]$ são determinados para garantir as propriedades de convergência global do método. Este procedimento geralmente é feito com o uso de técnicas de *linesearch* (minimização unidirecional de

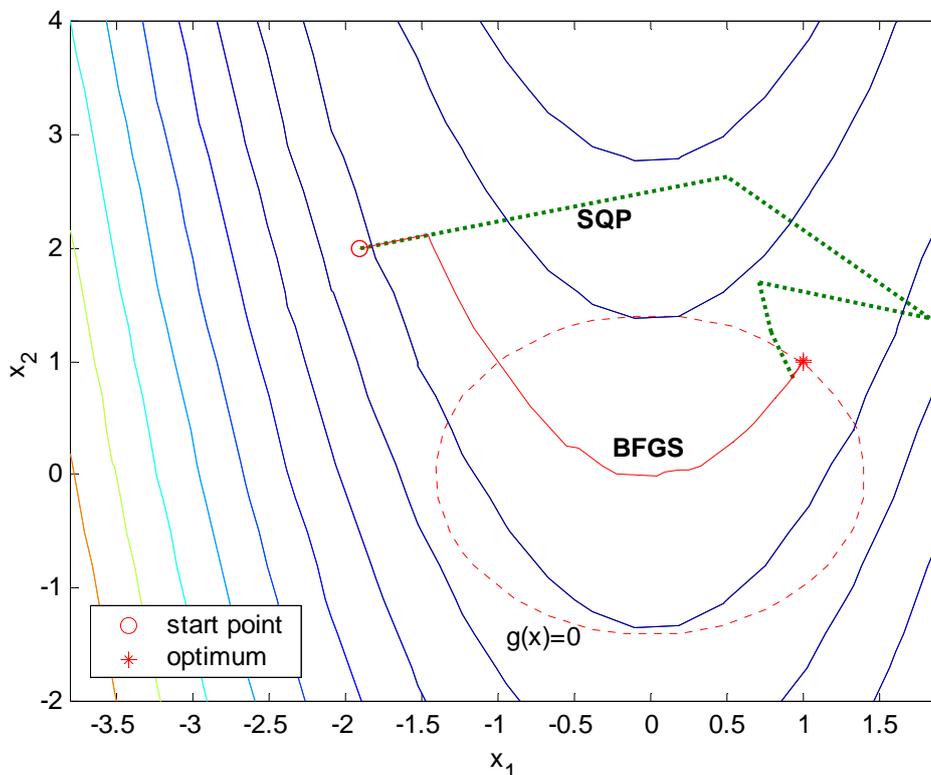
uma função de mérito) ou pela limitação do problema quadrático acima na *trust region*. A matriz $H(x^k, \lambda^k, \mu^k)$ é uma aproximação positiva definida da matriz Hessiana da função de Lagrange, que pode ser atualizada por qualquer método da métrica variável (DFP, BFGS, etc.). Como parte da solução do problema de programação quadrática tem-se também os multiplicadores de Lagrange λ^{k+1} e μ^{k+1} .

Se método de otimização estiver bem implementado, é provável que a solução do problema com restrição seja obtida em um número menor de iterações que o respectivo problema sem restrição. Isto geralmente ocorre porque as restrições podem gerar informações adicionais para a determinação da melhor direção de busca e do tamanho do passo mais apropriado.

Exemplo 6.5: $\min S(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ (Rosenbrock, 1960)

sujeito a: $g(x) = x_1^2 + x_2^2 - 2 \leq 0$

Partindo do ponto $x^0 = [-1,9 \ 2]^T$, a solução do problema sem restrição, usando o método BFGS chega na solução $x^* = [1 \ 1]^T$ após 165 avaliações da função objetivo, ao passo que o método SQP (com atualização BFGS) obtém a mesma solução com apenas 84 avaliações da função objetivo, usando a mesma tolerância (10^{-6}). A figura abaixo ilustra a trajetória dos dois métodos.



A solução do problema interno de programação quadrática, que pode ser escrito na seguinte forma:

$$\min_{d \in \mathbb{R}^n} q(d) = c^T d + \frac{1}{2} d^T H(x^k, \lambda^k, \mu^k) d$$

sujeito a: $A_e d = b_e$

$$A_I d \leq b_I$$

onde $c = \nabla S(x^k)$, $A_e = \nabla^T h(x^k)$, $A_I = \nabla^T g(x^k)$, $b_e = -h(x^k)$ e $b_I = -g(x^k)$, é geralmente realizada em duas etapas. A primeira destina-se a encontrar um ponto viável e a segunda obtém um seqüência de pontos viáveis que convergem para a solução.

A obtenção de um ponto viável pode ser realizada pela inclusão de variáveis artificiais, do mesmo modo que no caso da programação linear, isto é, resolve-se o seguinte problema auxiliar de programação linear:

$$\min_{v \in \mathbb{R}, d \in \mathbb{R}^n} v$$

sujeito a: $A_e d = b_e$

$$A_I d - v \leq b_I$$

Se uma solução viável não é encontrada, então a direção de busca é aquela que minimiza o valor de v .

Usando a estratégia dos conjuntos ativos para a solução da segunda etapa da programação quadrática, é construída a matriz das restrições ativas, A , e determina-se a base, Z , de seu espaço nulo ($A Z = 0$). A matriz Z é formada pelas últimas $m-w$ colunas da decomposição QR (ou decomposição ortogonal-triangular, ou decomposição de Householder) da matriz A^T , ou ainda pelas últimas $m-w$ colunas da matriz V da decomposição em valor singular da matriz $A (= U S V^H)$, onde w é o número de restrições ativas do problema quadrático ($w < m$), isto é:

$$Z_{i,j} = Q_{i,j+w} \quad i = 1, 2, \dots, n \quad , \quad j = 1, 2, \dots, m-w$$

onde $Q^H A^T = \begin{bmatrix} R \\ 0 \end{bmatrix}$ (Q é uma matriz unitária e $Q^{-1} = Q^H$ é a transposta conjugada)

Uma vez encontrada a base Z , determina-se a direção de busca que minimiza $q(d)$ no espaço nulo das restrições ativas. Escrevendo a função objetivo quadrática, $q(d)$, em termos da projeção de d no espaço nulo, $d_p = Z^T d$ (ou $d = Z d_p$), tem-se:

$$q(d_p) = c^T Z d_p + \frac{1}{2} (d_p)^T Z^T H(x^k, \lambda^k, \mu^k) Z d_p$$

que diferenciado em relação a d_p , resulta em:

$$\nabla q(d_p) = Z^T c + Z^T H(x^k, \lambda^k, \mu^k) Z d_p$$

ou seja, o gradiente projetado de $q(d)$. O termo $Z^T H(x^k, \lambda^k, \mu^k) Z$ é chamado de Hessiana projetada. Como a matriz Hessiana é mantida sempre positiva definida no método SQP, então o mínimo de $q(d_p)$ ocorre quando $\nabla q(d_p) = 0$, ou seja a solução do problema linear:

$$Z^T H(x^k, \lambda^k, \mu^k) Z d_p = -Z^T c$$

e a correção em x_k é então realizada nesta direção: $x^{k+1} = x^k + \alpha_k d^k$, com $d^k = Z d_p$.

Devido a natureza quadrática da função objetivo, $q(d_p)$, existe somente duas escolhas possíveis para α_k . O valor exato, $\alpha_k = 1$, que é usado quando nenhuma restrição é violada em x^{k+1} . Caso contrário, o passo ao longo de d^k até a restrição mais próxima é $\alpha_k < 1$, sendo esta restrição incluída no conjunto ativo da próxima iteração. Tal distância é dada por (multiplicando $x^{k+1} = x^k + \alpha_k d^k$ por A_i):

$$\alpha_k = \min_{i \in \{1, \dots, m\}} \frac{-(A_i x^k - b_i)}{A_i d^k}, \text{ com } A_i d^k > 0$$

que é definida para as restrições que não estão no conjunto ativo, e na direção de sua fronteira ($A_i d^k > 0$).

Quando n restrições independentes estão incluídas no conjunto ativo e o mínimo ainda não foi localizado, então os multiplicadores de Lagrange são calculados de modo a satisfazer o sistema não singular de equações:

$$A^T \lambda^k = c$$

Se todos os elementos de $\lambda^k \geq 0$, então x^k é a solução ótima do problema quadrático. Entretanto, se algum componente de λ^k é negativo e este não corresponder a uma restrição de igualdade, então a restrição associada é removida do conjunto ativo para a próxima iteração.

A solução do problema de programação quadrática gera um vetor direção, d^k , que é usado para calcular o próximo ponto no nível mais externo das iterações SQP, ou seja:

$$x^{k+1} = x^k + \alpha_k d^k$$

sendo α_k determinado de modo a minimizar uma função de mérito, escrita genericamente como:

$$\Psi(\alpha, x^k, d^k, \lambda^k, \mu^k, \lambda^{k+1}, \mu^{k+1}, r_k)$$

onde λ^{k+1} e μ^{k+1} são os multiplicadores de Lagrange obtidos na solução do problema quadrático que gerou d^k , e r_k é um escalar (ou vetor) usado na penalização das restrições. Por exemplo, a função implementada no SQP do MATLAB tem a forma (Han, 1977 e Powell, 1978):

$$\Psi(\alpha) = S(x) + r_e^T h(x) + r_l^T \max\{0, g(x)\} \quad , \quad x = x^k + \alpha d^k$$

onde $r_{e,i} = r_{e,i}^{k+1} = \max_i \left\{ \lambda_i^k, \frac{1}{2}(\lambda_i^k + r_{e,i}^k) \right\}$ e $r_{l,i} = r_{l,i}^{k+1} = \max_i \left\{ \mu_i^k, \frac{1}{2}(\mu_i^k + r_{l,i}^k) \right\}$, que

permite uma contribuição para a função de mérito das restrições que não estavam ativas na última programação quadrática, mas que estiveram ativas em iterações adjacentes. O valor inicial do fator de penalização pode ser calculado da seguinte forma:

$$r_{e,i}^0 = \frac{\|\nabla S(x)\|}{\|\nabla h_i(x)\|} \quad \text{e} \quad r_{l,i}^0 = \frac{\|\nabla S(x)\|}{\|\nabla g_i(x)\|}$$

que assegura uma elevada penalização das restrições com menor gradiente, isto é, as que estiverem mais perto de suas fronteiras (este mesmo procedimento pode ser usado para obter uma estimativa inicial para os multiplicadores de Lagrange).

Outra forma muito utilizada é a do Lagrangeano aumentado (Powell, 1969 e Hestenes, 1969). Neste caso a função de mérito é dada por:

$$\Psi(\alpha) = S(x) + \lambda^T h(x) + \mu^T f(x) + P(x, r_k)$$

onde $f_j(x) = \max\{0, g_j(x)\}$, $j = 1, 2, \dots, p$

$$P(x, r_k) = \frac{1}{2} r_k \{h^T(x) h(x) + f^T(x) f(x)\}$$

$$x = x^k \quad \text{ou} \quad x = x^k + \alpha d^k$$

$$\lambda = \lambda^k \quad \text{ou} \quad \lambda = \lambda^k + \alpha (\lambda^{k+1} - \lambda^k)$$

$$\mu = \mu^k \quad \text{ou} \quad \mu = \mu^k + \alpha (\mu^{k+1} - \mu^k)$$

Exercícios de fixação

Resolver os seguintes problemas de programação não linear usando SQP:

1. $\min S(x) = x_1 x_2$

sujeito a: $g_1(x) = x_1^2 + x_2^2 - 25 \leq 0$

2. $\min S(x) = (x_1 - 2)^2 + (x_2 - 1)^2$

sujeito a: $h_1(x) = x_1 - 2x_2 + 1 = 0$

$$g_1(x) = x_1^2 / 4 + x_2^2 - 1 \leq 0$$

3. $\min S(x) = x_1^3 - 3x_1x_2 + 4$

sujeito a: $h_1(x) = -2x_1 + x_2^2 - 5 = 0$

$$g_1(x) = 18 - 5x_1 - 2x_2 \leq 0$$

6.7 Otimização multi-objetivo

O projeto ou operação ótima de um sistema geralmente implica na satisfação de várias especificações e requisitos, tais como elevada confiabilidade, baixo custo, alta produtividade, baixo risco de operação, alta qualidade e baixa poluição. Estes aspectos, freqüentemente, estão em conflito uns com os outros e, portanto, uma solução compromisso deve ser obtida. Conseqüentemente, a formulação de uma única função objetivo com restrições pode não representar de forma adequada o problema. Sendo necessária a definição de um conjunto de funções objetivos, que devem ser balanceadas de alguma maneira.

Em geral, a importância relativa destas funções objetivos não são conhecidas até que se tenha idéia das potencialidades do sistema e do grau de compromisso que pode ser admitido. As preferências entre as funções objetivos recaem muito sobre a experiência e intuição do projetista. Existem pelo menos três estratégias diferentes para expressar estas preferências:

- 1) Método da soma ponderada das funções objetivos;
- 2) Método das restrições ϵ ;
- 3) Método da perseguição dos objetivos.

O método da soma ponderada transforma o problema de otimização multi-objetivo:

$$\begin{aligned} \min_{x \in K} F(x) &= [F_1(x) \ F_2(x) \ \dots \ F_q(x)]^T \\ \text{sujeito a: } h_j(x) &= 0, \quad j = 1, 2, \dots, m \\ g_j(x) &\leq 0, \quad j = 1, 2, \dots, p \end{aligned}$$

em um problema de otimização escalar pela definição da soma ponderada das funções objetivos:

$$\begin{aligned} \min_{x \in K} S(x) &= w^T F(x) \\ \text{sujeito a: } h_j(x) &= 0, \quad j = 1, 2, \dots, m \\ g_j(x) &\leq 0, \quad j = 1, 2, \dots, p \end{aligned}$$

onde w é o vetor dos pesos relativos, geralmente definido como uma combinação convexa:

$$\sum_{i=1}^q w_i = 1 \quad \text{e} \quad w \geq 0$$

Conseqüentemente, este problema pode ser resolvido por qualquer uma das técnicas de otimização de um funcional escalar. Uma das dificuldades desta estratégia é a homogeneização das diferentes unidades de medida tratadas em cada função objetivo

(tais como grau de confiabilidade, custo, segurança, qualidade), ou seja, como escolher os pesos adequadamente. Outra dificuldade está associada ao tratamento das não convexidades que podem existir no espaço das funções objetivos, ilustrado a seguir.

O método das restrições ε consiste em minimizar uma das funções objetivos, expressando as demais como restrições de desigualdades, ou seja:

$$\begin{aligned} & \min_{x \in K} F_r(x) \\ \text{sujeito a: } & h_j(x) = 0, \quad j = 1, 2, \dots, m \\ & g_j(x) \leq 0, \quad j = 1, 2, \dots, p \\ & F_i(x) \leq \varepsilon_i, \quad i = 1, 2, \dots, q, \quad i \neq r \end{aligned}$$

As dificuldades naturais deste método estão associadas à seleção da função objetivo a ser minimizada e a definição do nível de restrição para as demais funções (ε_i). Pode-se também associar a este método uma estratégia de priorização das funções objetivos durante o processo de otimização.

No método da perseguição dos objetivos (*goal attainment*) é definido um conjunto de objetivos $F^* = \{F_1^*, F_2^*, \dots, F_q^*\}$ a ser perseguido pelas respectivas funções $F(x) = \{F_1(x), F_2(x), \dots, F_q(x)\}$. A formulação do problema é dada por:

$$\begin{aligned} & \min_{\gamma \in \mathfrak{R}, x \in K} \gamma \\ \text{sujeito a: } & h_j(x) = 0, \quad j = 1, 2, \dots, m \\ & g_j(x) \leq 0, \quad j = 1, 2, \dots, p \\ & F_i(x) - w_i \gamma \leq F_i^*, \quad i = 1, 2, \dots, q \end{aligned}$$

o termo $w_i \gamma$ permite uma certa folga para mais ou para menos no alcance dos objetivos. Valores nulos de w_i implicam em restrições rígidas para os objetivos. Como nesta formulação os objetivos podem ser super-alcançados ($\gamma < 0$) ou sub-alcançados ($\gamma > 0$), os valores de F^* não necessitam de elevada precisão. O grau relativo deste alcance é controlado pelo vetor de pesos, w . Quando $w_i = |F_i^*|$, é assegurado um mesmo percentual de super- e sub-alcance na respectiva função objetivo, pois neste caso $\gamma \geq [F_i(x) - F_i^*] / |F_i^*|$. Observe que γ representa a quantidade máxima relativa de desvio do objetivo (para mais ou para menos).

O grande problema existente na otimização multi-objetivo é que se existe conflito entre as funções objetivos, então a solução do problema original não é única. Conseqüentemente, a solução obtida por qualquer um dos métodos acima vai estar associada aos pesos ou restrições impostas em cada alternativa. Para tratar da não

unicidade de solução é definido o conceito de *não inferioridade*, também conhecido como *otimalidade de Pareto*.

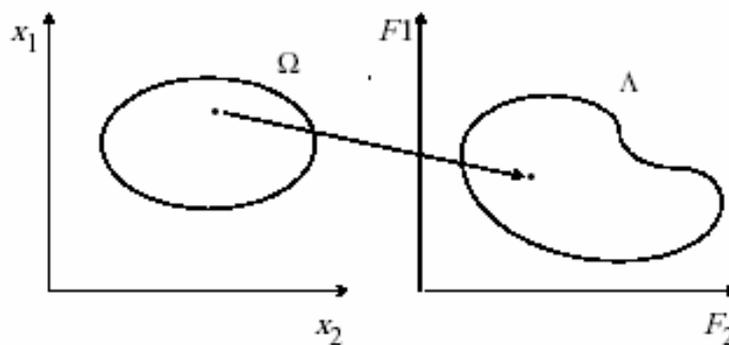
Uma solução não inferior é aquela que uma posterior melhoria em uma das funções objetivos causa a degradação de outras. Para entender melhor este conceito, define-se a região viável, Ω , no espaço das variáveis de decisão $x \in \mathbb{R}^n$, que satisfaz todas as restrições, isto é:

$$\begin{aligned} \Omega &= \{x \in K \subseteq \mathbb{R}^n\} \\ \text{sujeito a: } h_j(x) &= 0, \quad j = 1, 2, \dots, m \\ g_j(x) &\leq 0, \quad j = 1, 2, \dots, p \end{aligned}$$

e a correspondente região viável, Λ , no espaço das funções objetivos:

$$\Lambda = \{y = F(x) \in \mathbb{R}^q \mid x \in \Omega\}$$

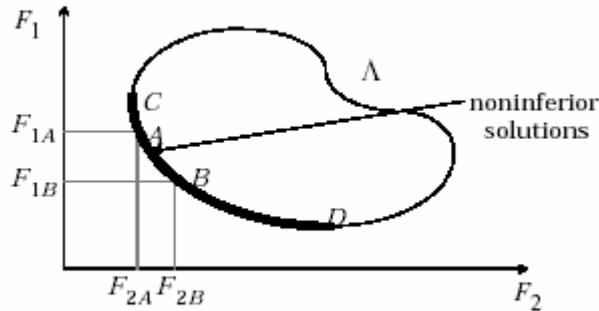
Ou seja, o vetor de desempenho $F(x)$ é um mapeamento do espaço das variáveis de decisão no espaço das funções objetivos, como ilustra a figura abaixo.



Então, um ponto $x^* \in \Omega$, é uma solução não inferior se para alguma vizinhança de x^* não existe um Δx tal que $(x^* + \Delta x) \in \Omega$ e:

$$\begin{aligned} F_i(x^* + \Delta x) &\leq F_i(x^*), \quad i = 1, 2, \dots, q \\ F_j(x^* + \Delta x) &< F_j(x^*), \quad \text{para algum } j \end{aligned}$$

Na representação bidimensional, o conjunto de soluções não inferiores (ou *conjunto de Pareto*) está sobre a curva entre os pontos C e D da figura abaixo. Os pontos A e B representam dois pontos não inferiores distintos, ou seja, pontos onde a melhoria em uma função objetivo requer a degradação de outra: $F_{1B} < F_{1A} \rightarrow F_{2B} > F_{2A}$.

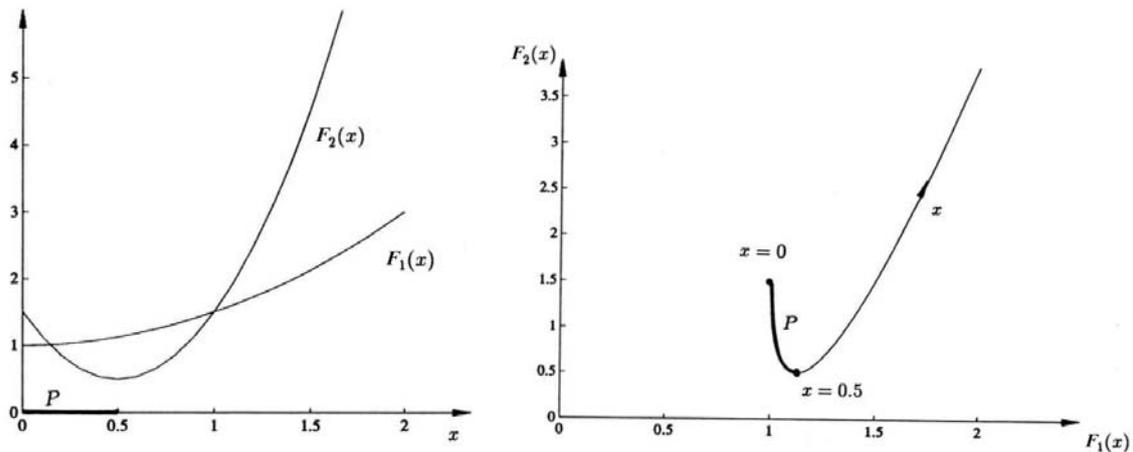


Como qualquer ponto em Ω que não seja um ponto não inferior há a possibilidade de obter um novo ponto com melhoria em todas as funções objetivos, o problema de otimização multi-objetivo resume-se em gerar e selecionar pontos do conjunto de Pareto.

Exemplo 6.6: $\min F(x) = [F_1(x) \ F_2(x)]^T$

$$\begin{aligned} \text{sujeito a: } & F_1(x) = 1 + \frac{1}{2} x^2 \\ & F_2(x) = 4(x - \frac{1}{2})^2 + \frac{1}{2} \\ & x \geq 0 \end{aligned}$$

Para este exemplo o conjunto de Pareto (P) está ilustrado nos gráficos abaixo.



Resolvendo este exemplo pelo método das restrições ϵ , com a escolha de $F_2(x)$ como função objetivo, tem-se o seguinte problema de otimização:

$$\begin{aligned} \min_{x \in \mathbb{R}} & S(x) = F_2(x) \\ \text{sujeito a: } & F_1(x) \leq \epsilon \\ & x \geq 0 \end{aligned}$$

cujas soluções para vários valores de $\epsilon \geq 1$ (observe que $\min F_1(x) = 1, x \geq 0$, ocorrendo em $x = 0$) são dadas por:

$$x^*(\varepsilon) = \begin{cases} 1/2 & \varepsilon \geq 1,125 \\ \sqrt{2(\varepsilon-1)} & 1 \leq \varepsilon \leq 1,125 \end{cases}$$

o valor de $\varepsilon = 1,125$ corresponde a $F_1(1/2)$, sendo que para valores de $\varepsilon \geq 1,125$ a restrição $F_1(x) \leq \varepsilon$ é inativa, e valores entre $1 \leq \varepsilon \leq 1,125$, a restrição está ativa. Portanto, o conjunto de Pareto para este problema corresponde a $P = \{x^*(\varepsilon) \mid 1 \leq \varepsilon \leq 1,125\} = \{0 \leq x \leq 1/2\}$.

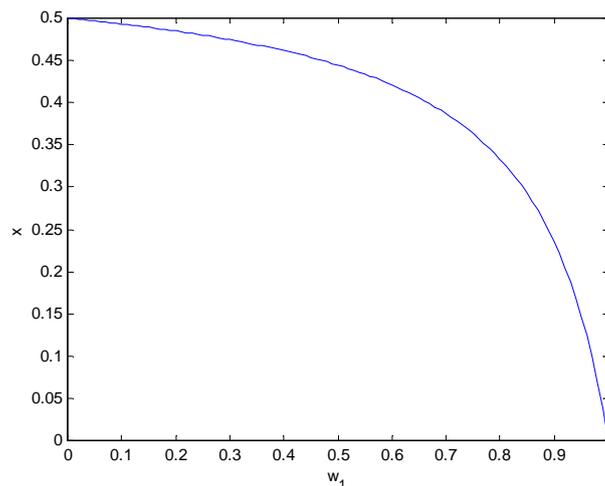
A solução do exemplo acima pelo método das somas ponderadas recai no seguinte problema de otimização:

$$\min_{x \in \mathbb{R}} S(x) = w^T F(x) = w_1 F_1(x) + (1 - w_1) F_2(x)$$

sujeito a: $x \geq 0$

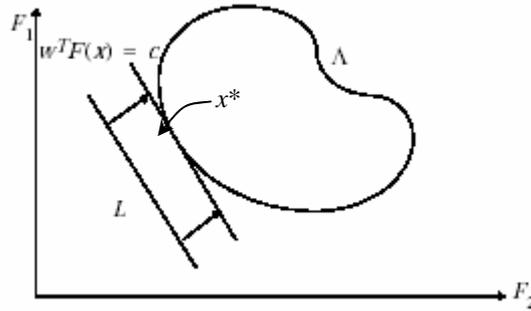
cujas soluções em função do peso w_1 são dadas por:

$$x^*(w_1) = \frac{4(1 - w_1)}{8 - 7w_1}$$

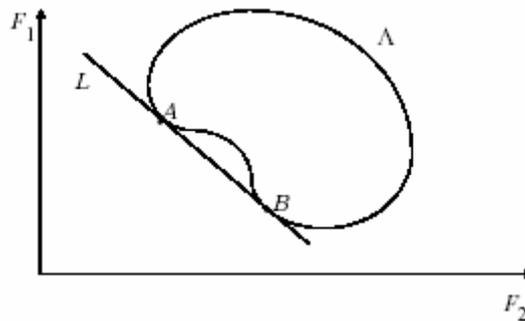


Quando $w_1 = 0$, a solução é o mínimo de $F_2(x)$, ou seja, $x^* = 1/2$, e para $w_1 = 1$ a solução é o mínimo de $F_1(x)$, isto é, $x^* = 0$. Para valores entre $0 \leq w_1 \leq 1$, a figura acima mostra que $P = \{x^*(w_1) \mid 0 \leq w_1 \leq 1\} = \{0 \leq x \leq 1/2\}$.

A existência de soluções não concorrentes dentro do conjunto de Pareto pode gerar um conjunto não convexo, dificultando a obtenção de soluções, principalmente pelo método das somas ponderadas. Neste método, a função objetivo é linear dentro do espaço das funções objetivos: $S(y) = w^T y$ e, portanto, as curvas de níveis são hiperplanos dentro deste espaço. Conseqüentemente, a minimização de $S(y)$ pode ser interpretada como encontrar a menor curva de nível que toca no conjunto Λ , como pode ser visto na figura abaixo.



onde \$L\$ é a linha de nível. A seleção dos pesos, \$w\$, define a inclinação do hiperplano e, conseqüentemente, o ponto onde o conjunto \$\Lambda\$ será tocado. O problema de não convexidade surge quando a fronteira inferior de \$\Lambda\$ é não convexa, como ilustra a figura abaixo.

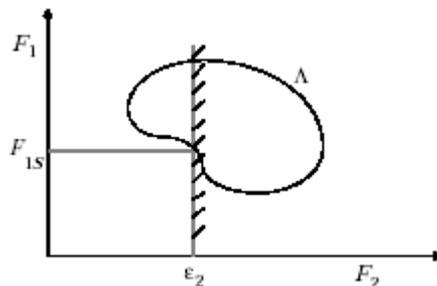


Neste caso, o conjunto de soluções não inferiores entre os pontos \$A\$ e \$B\$ não estão acessíveis. Este problema de não convexidade pode ser tratado pelo método das restrições \$\epsilon\$, conforme ilustra a figura a seguir relativa ao problema abaixo:

$$\min_{x \in \Omega} S(x) = F_1(x)$$

sujeito a: $F_2(x) \leq \epsilon_2$

onde verifica-se que o método é capaz de identificar pontos não inferiores dentro da região de não convexidade de \$\Lambda\$.

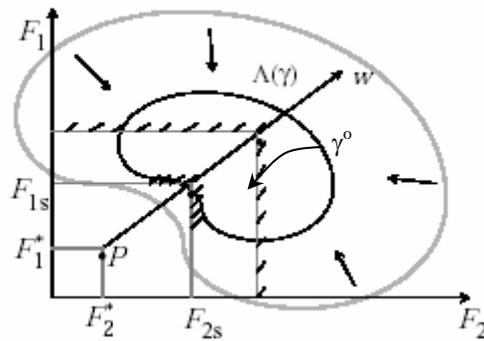


Uma informação importante do método das restrições \$\epsilon\$ é a sensibilidade da solução no ponto \$x^*(\epsilon)\$. Ou seja, o quanto varia o valor de \$F_r(x^*)\$ em relação as variações nas demais funções objetivos, usadas como restrições. Esta informação é diretamente dada pelos multiplicadores de Kuhn-Tucher no ponto ótimo:

$$\frac{\partial F_r^*(\varepsilon)}{\partial F_i} = -\mu_i^* \quad , \quad \forall i \neq r$$

representando uma ajuda importante na tomada de decisão em estabelecer os compromissos entre as funções objetivos.

O método da perseguição dos objetivos apresenta uma forma mais conveniente de expressar os compromissos entre as funções objetivos, dando uma maior flexibilidade para o algoritmo de otimização buscar a melhor solução. A figura abaixo ilustra a evolução deste método na direção da solução.



onde F_1^* e F_2^* são os objetivos estabelecidos, representados pelo ponto P, e (F_{1s}, F_{2s}) é o ponto ótimo obtido para um determinado conjunto de pesos $\{w_1, w_2\}$, que define a direção de busca. A região viável, $\Lambda(\gamma)$, vai encolhendo à medida que o valor de γ é reduzido. Este método pode ser visto como uma generalização do método das restrições ε , que é resgatado quando $w_i = 0 \quad \forall i \neq r$ e $w_r = 1$.

A formulação do problema de otimização pelo método da perseguição dos objetivos também pode ser escrito na forma minimax, ou seja:

$$\min_{x \in \Omega} \max_i \Lambda_i$$

onde $\Lambda_i = \frac{F_i(x) - F_i^*}{w_i}$, $i = 1, 2, \dots, q$, ou seja, deseja-se minimizar o pior caso.

Ao resolver este problema usando métodos tipo SQP, uma forma apropriada de representar a função de mérito é a seguinte:

$$\Psi(x) = \max_i \Lambda_i$$

pois funções tipo Lagrange:

$$\Psi(x, \gamma) = \gamma + \sum_{i=1}^q r_i \max\{0, F_i(x) - w_i \gamma - F_i^*\}$$

podem apresentar problemas devido as possíveis concorrências entre as funções objetivos. Contudo, como $\Lambda_i = \infty$ para $w_i = 0$ (restrições rígidas), uma combinação entre as duas formas acima é mais recomendada, ou seja:

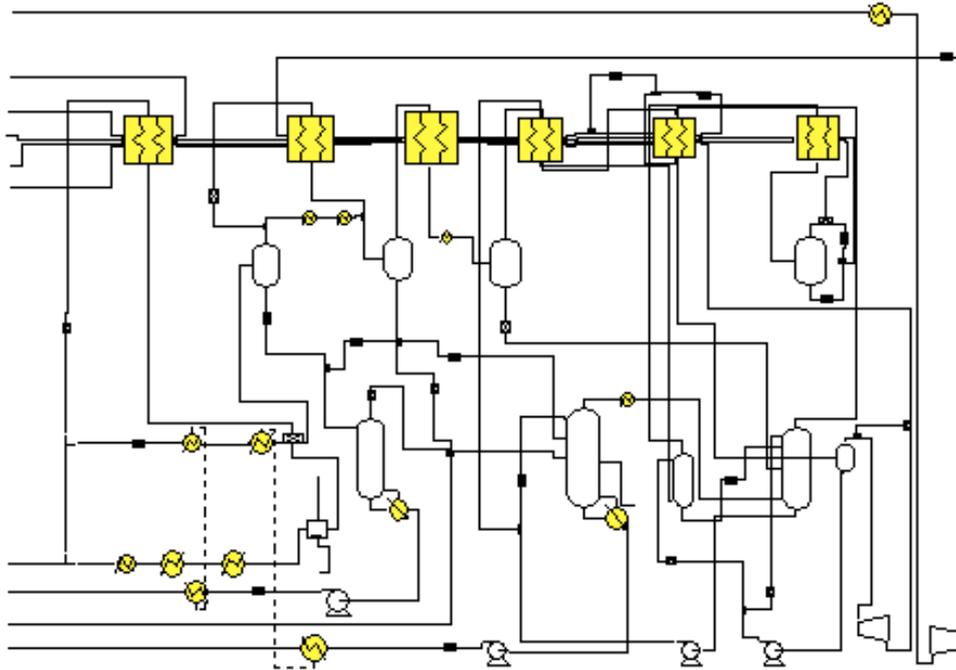
$$\Psi(x, \gamma) = \begin{cases} \sum_{i=1}^q r_i \max\{0, F_i(x) - w_i \gamma - F_i^*\} & \text{se } w_i = 0 \\ \max_i \Lambda_i & \text{se } w_i \neq 0 \end{cases}$$

Finalmente, ao usar métodos quasi-Newton na solução deste problema, devido a linearidade de γ na função objetivo, a contribuição desta variável é nula para a matriz Hessiana. Portanto, ao inicializar a aproximação da Hessiana com a matriz identidade, o elemento da diagonal relacionado com γ deve ser anulado (ou próximo a zero para manter a matriz positiva definida), de modo a acelerar a convergência da programação quadrática interna.

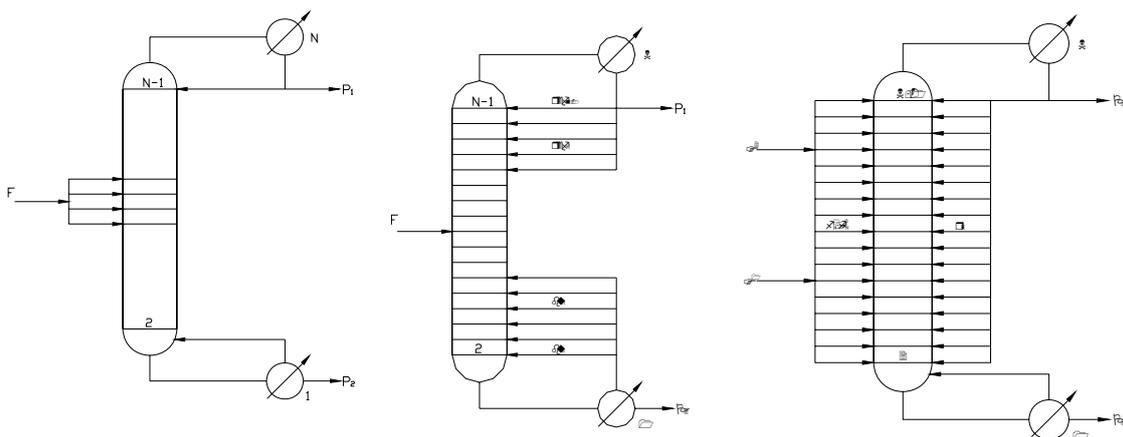
7. Programação Inteira Mista

A programação inteira mista (MIP – *Mixed-Integer Programming*), caracterizada pela existência de variáveis contínuas e inteiras, surge em uma grande variedade de aplicações, tais como:

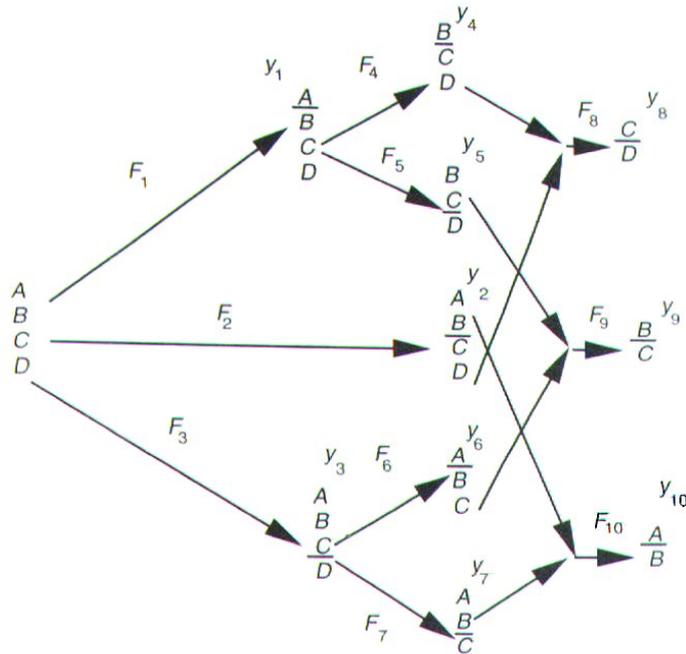
- síntese de processos (rede de trocadores de calor e/ou massa, superestruturas, sistemas de utilidades, etc.);



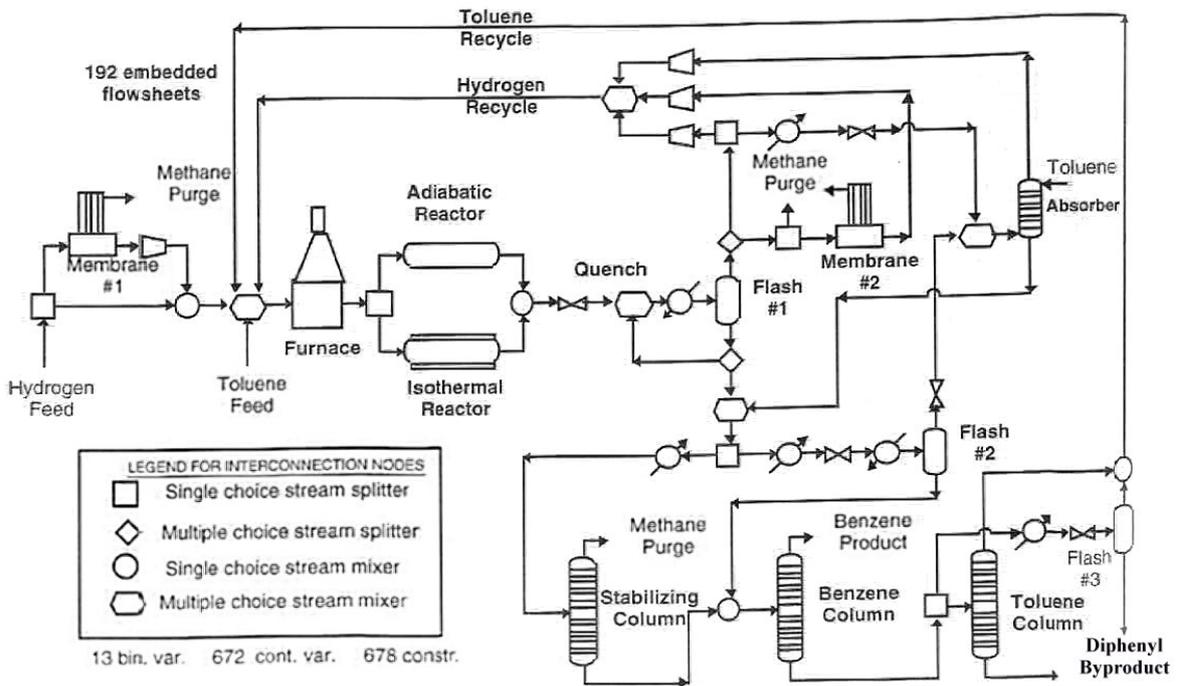
Rede de trocadores de calor.



Localização de alimentação e/ou refluxo de colunas de destilação.



Sequenciamento de destilações.

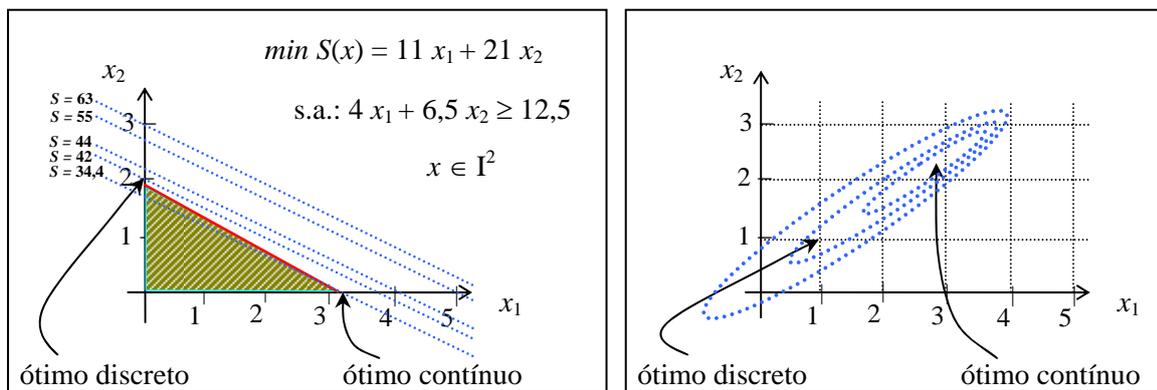


Superestrutura para a hidrodealquilação do tolueno.

- projeto, programação e planejamento de sistemas descontínuos com múltiplos produtos e/ou múltiplos propósitos;
- localização, alocação, planejamento e programação de facilidades;
- interação entre projeto e controle de sistemas;

- projeto de produtos;
- síntese de topologias (redes de transporte e distribuição, estruturas, etc.).

As variáveis inteiras de decisão cujos níveis formam uma dicotomia (por exemplo, instalar ou não instalar um equipamento) são chamadas de *variáveis binárias* (ou variáveis 0–1), assumindo os valores de 0 ou 1. As variáveis inteiras que assumem valores elevados, podem as vezes ser tratadas como variáveis contínuas (*relaxamento*), sendo a solução ótima arredondada para o número inteiro mais próximo, podendo resultar em uma solução sub-ótima. Por exemplo, as figuras abaixo apresentam situações onde a solução relaxada difere significativamente da solução ótima.



As variáveis inteiras, z , com limites superiores e inferiores dados, $z^l \leq z \leq z^u$, podem ser expressas como um vetor de variáveis binárias, $y \in Y = \{0,1\}^q$, pela seguinte fórmula:

$$z = z^l + y_1 + 2y_2 + 4y_3 + \dots + 2^{q-1}y_q$$

onde q é o número mínimo de variáveis 0–1 necessárias para representar z . Este número mínimo é dado por:

$$q = 1 + \text{int} \left\{ \frac{\log(z^u - z^l)}{\log 2} \right\}$$

sendo que a função “int” trunca o argumento real para um valor inteiro. Esta aproximação, entretanto, pode não ser prática quando os limites são grandes.

A maioria dos problemas de programação mista são inerentemente de natureza combinatorial, isto é, as variáveis inteiras do problema de otimização correspondem a combinações de n elementos tomando m de cada vez, resultando em um número de

$\binom{n}{m}$ combinações, ou correspondem a permutações de n elementos, levando a $n!$ possibilidades.

Os problemas MIP que envolvem somente variáveis inteiras são denotados por IP (*Integer Programming*), tendo como caso particular a programação inteira binária (BIP), onde todas as variáveis inteiras são binárias. Quando um problema MIP é linear na função objetivo e nas restrições, tem-se o caso MILP (*Mixed-Integer Linear Programming*). Finalmente, quando há relações não lineares na função objetivo ou nas restrições, tem-se o problema de programação não linear inteira mista (MINLP).

Exemplos clássicos de programação inteira são o do caixeiro viajante e o da mochila. O problema da mochila consiste em colocar q artigos valiosos dentro de uma mochila, não ultrapassando um peso (ou volume) máximo W , de modo a maximizar o valor total dos artigos. Designando cada artigo como y_i , seu respectivo peso (ou volume) de w_i e seu valor de c_i , o problema BIP pode ser descrito como:

$$\begin{aligned} \max S(y) &= c^T y \\ \text{sujeito a: } &w^T y \leq W \\ &y \in Y = \{0,1\}^q \end{aligned}$$

sendo $y_i = 1$ quando o i -ésimo artigo for selecionado e $y_i = 0$ caso contrário.

O problema do caixeiro viajante consiste em fazer um vendedor visitar n cidades somente uma vez, retornando para a cidade de origem, percorrendo o menor caminho possível. Designando como $y_{ij} = 1$ a viagem da cidade i para a cidade j e c_{ij} a distância entre estas cidades, o problema BIP pode ser formulado como:

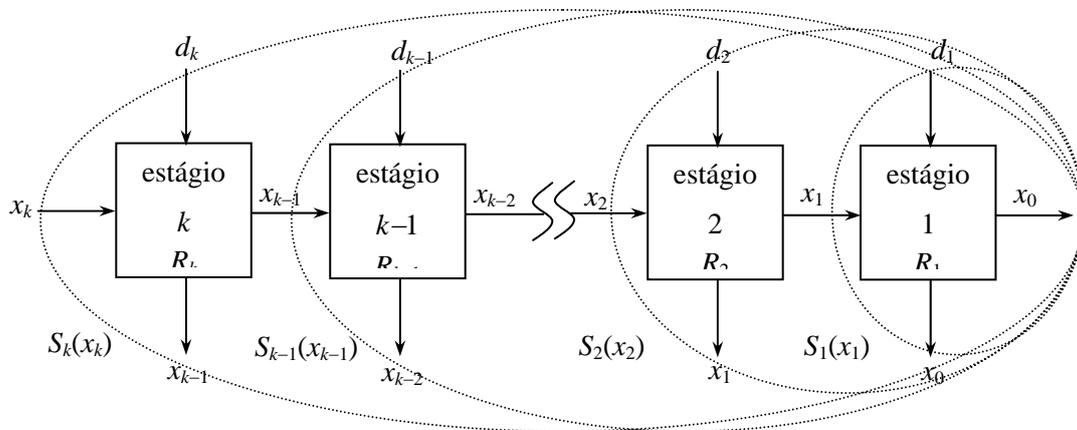
$$\begin{aligned} \min S(y) &= \sum_{i=1}^n \sum_{j=1}^n c_{ij} y_{ij} \\ \text{sujeito a: } &\sum_{i=1}^n y_{ij} = 1, \quad \sum_{j=1}^n y_{ij} = 1, \quad y_{ii} = 0, \quad i, j = 1, 2, \dots, n \\ &y \in Y = \{0,1\}^{q=n \times n} \end{aligned}$$

Este problema também pode ser interpretado como um problema de associação, como por exemplo, a associação de correntes de processo a trocadores de calor, limitando um trocador por corrente.

7.1 Programação dinâmica

Embora a programação dinâmica não esteja, necessariamente, enquadrada dentro da programação mista, ela é descrita neste capítulo por se tratar de um método de solução

de problemas que podem ser decompostos em um número inteiro de estágios. Muitos processos e equipamentos podem ser representados (física ou imaginariamente) por múltiplos estágios (ou períodos de tempo), como ilustra a figura abaixo.

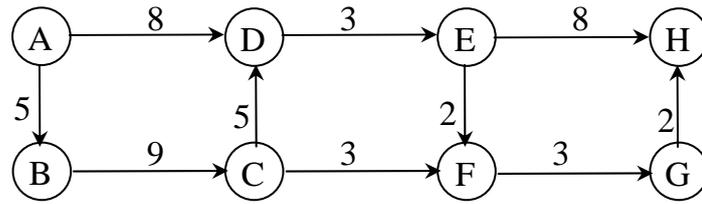


- onde R_k é o valor da função objetivo do estágio k ,
- x_k é o vetor de variáveis de entrada do estágio k ,
- x_{k-1} é o vetor de variáveis de saída do estágio k ,
- d_k é o vetor de variáveis de decisão (independente) do estágio k ,
- $S_k(x_k)$ é a função objetivo global dos estágios 1 a k .

A essência da programação dinâmica é decompor um problema de otimização multivariável interconectado em uma seqüência de subproblemas que podem ser resolvidos serialmente. Cada subproblema contém um subconjunto de variáveis de decisão, que pode ser otimizado por qualquer uma das técnicas de otimização descritas nos capítulos anteriores.

A programação dinâmica está baseada no *princípio da otimalidade de Bellman* (1957): “Se uma decisão forma uma solução ótima em um estágio de um processo, então qualquer decisão remanescente deve ser ótima com respeito ao resultado desta decisão tomada”. Este conceito pode ser melhor implementado se as tomadas de decisão de um estágio para outro são realizadas na direção do fluxo de informação nos estágios, que pode estar no sentido oposto ao fluxo de material.

Exemplo 7.1: Para ilustrar este princípio, seja o problema de distribuição de combustível através de uma rede de tubulações, cujo escoamento se dá somente através do terminal H, segundo a figura abaixo, onde os valores representam os custos de transporte.



Deseja-se determinar o menor custo de transporte da estação de bombeamento C para o terminal H. Ao invés de tentar todos os caminhos possíveis (isto é, uma busca exaustiva), pode-se aplicar o princípio da otimalidade. Partindo do terminal H, obtém-se o custo mínimo de transporte de G para H: $C_{GH}^* = 2$, e de E para H:

$$C_{EH}^* = \min \{C_{EH}, (C_{EF} + C_{FH}^*)\}$$

como $C_{FH}^* = 3 + 2 = 5$, tem-se $C_{EH}^* = \min \{8, 2 + 5\} = 7$. O custo mínimo de D para H é dado por $C_{DH}^* = C_{DE} + C_{EH}^* = 3 + 7 = 10$. Portanto, o menor custo de bombeamento de C para H é obtido de:

$$C_{CH}^* = \min \{(C_{CD} + C_{DH}^*), (C_{CF} + C_{FH}^*)\} = \min \{(5 + 10), (3 + 5)\} = 8.$$

O princípio de Bellman pode ser formulado como segue:

$$S_k^*(x_k) = \min_{d_k} [R_k(x_k, d_k) + S_{k-1}^*(x_{k-1})]$$

Como cada função objetivo otimizada, $S_k^*(x_k)$, depende de suas variáveis de entrada, x_k , estas formas funcionais devem ser obtidas antes de passar para o próximo estágio, ou devem ser calculadas recursivamente. Portanto, a programação dinâmica só é atrativa para problemas de baixa dimensão. Por exemplo, se um estágio k possuir m variáveis de entrada, cada uma podendo assumir r_j níveis viáveis, $j = 1, 2, \dots, m$, então o número de combinações possíveis onde a função objetivo otimizada deste estágio deve ser calculada é $\prod_{j=1}^m r_j$, ou r^m se $r_j = r$.

Vários métodos tem sido propostos para reduzir a dimensionalidade da programação dinâmica. Contudo, não existe uma formulação matemática padrão para esta programação, ou seja, o problema é formulado em função das características do processo.

7.2 Programação linear inteira mista (MILP)

Como na grande maioria das aplicações que recaem em problemas de programação linear inteira mista, as variáveis inteiras podem ser representadas por variáveis binárias, somente este caso é tratado nesta seção, sem perda de generalidade.

A formulação MILP com variáveis 0–1 é dada por:

$$\min S(x,y) = c^T x + d^T y$$

$$\text{sujeito a: } A x + B y \leq b$$

$$x \geq 0$$

$$x \in X \subseteq \mathbb{R}^n, \quad y \in Y = \{0,1\}^q$$

Para um dado valor do vetor y , o problema acima recai em um problema de programação linear (LP). Uma busca exaustiva (ou *enumeração*), enumerando todas as possíveis combinações de variáveis 0–1 para os elementos do vetor y , poderia ser utilizada para a solução do problema. Contudo, para problemas de médio e grande porte este procedimento torna-se proibitivo, devido a natureza combinatorial.

Os algoritmos existentes para problemas MILP podem ser classificados como:

- 1) Métodos “*branch and bound*”;
- 2) Métodos dos planos de corte;
- 3) Métodos de decomposição;
- 4) Métodos baseados em lógica e disjunções.

Nos algoritmos “*branch and bound*” (Land e Doig, 1960), uma árvore binária é empregada para representar as combinações 0–1, a região viável é sistematicamente particionada em subdomínios e limites inferiores e superiores são gerados em diferentes níveis da árvore binária.

Nos métodos dos planos de corte (Gomory, 1958), a região viável não é dividida em subdomínios, mas novas restrições (chamadas de planos de cortes) são geradas a cada estágio e adicionadas ao conjunto de restrições, reduzindo a região viável até obter uma solução ótima.

Nos métodos de decomposição (Benders, 1962), a estrutura matemática do modelo é explorada através de partições das variáveis, dualidade e métodos de relaxação.

Nos métodos baseados em lógicas e disjunções, técnicas de inferência simbólica, em termos das variáveis binárias, são utilizadas para restringir a região viável.

Nesta seção é dada ênfase ao método de “*branch and bound*”, que é mais comumente utilizado nos algoritmos de MILP e MINLP. O método da decomposição de Benders é tratado em detalhe na próxima seção.

7.2.1 Princípio dos métodos dos planos de corte

A idéia básica destes métodos é a seguinte:

- 1) Resolve-se o problema MILP com as variáveis inteiras relaxadas (ou seja, consideradas como contínuas), transformando-se em um problema LP. Se a solução do problema LP possuir todas as variáveis relaxadas inteiras, então a solução ótima do problema original foi encontrada;
- 2) Caso contrário, é sempre possível eliminar a solução obtida pela adição de restrições extras ao problema, sem contudo eliminar qualquer solução inteira. Tais restrições são chamadas de *planos de corte*;
- 3) Após a adição de um plano de corte (ou possivelmente mais do que um), o problema MILP aumentado é novamente resolvido, com as variáveis inteiras relaxadas. Observa-se que, como a solução eliminada é viável no dual, é vantajoso usar algoritmos duais para a solução deste problema, pois parte-se de um ponto inicial viável;
- 4) Se a solução do problema aumentado apresentar todas as variáveis relaxadas inteiras, então a solução ótima do problema original foi encontrada. Caso contrário, o procedimento é repetido até a convergência.

Se os planos de corte forem adequadamente escolhidos a cada estágio, então o poliedro inicial será progressivamente reduzido até coincidir com a casca convexa da solução inteira (pelo menos na vizinhança da solução ótima) e, conseqüentemente, a solução contínua do problema aumentado se tornará inteira nas variáveis relaxadas. Portanto, a maneira em que os planos de corte são escolhidos é decisiva para a convergência do método. Infelizmente, não há um método sistemático para gerar todas as restrições que definem a casca convexa dos pontos inteiros contidos dentro de um poliedro convexo. Além disto, seria custoso e supérfluo gerar todas as faces de uma casca convexa, pois a maioria das restrições estariam inativas, não contribuindo para a definição do ponto ótimo. Contudo, existem vários métodos na literatura que procuram definir os planos de corte de modo a obter uma convergência finita do algoritmo.

Exemplo 7.2: Seja o problema:

$$\begin{aligned} \min S(x) &= -10 x_1 - 11 x_2 \\ \text{sujeito a: } & 10 x_1 + 12 x_2 \leq 59 \\ & x \in \mathbb{I}^2 \quad \text{e} \quad x \geq 0 \end{aligned}$$

A solução do problema original relaxado é obtida no ponto $x = [5,9 \ 0]^T$. Portanto, a restrição adicional:

$$x_1 \leq 5$$

eliminará este ponto sem excluir qualquer solução inteira. Similarmente, qualquer restrição suplementar da forma:

$$x_1 + x_2 \leq \alpha$$

com $5 \leq \alpha \leq 5,9$ é um plano de corte. De fato, muitos outros planos de corte são possíveis, existindo, em princípio infinitas possibilidades. Em particular, se as seguintes restrições fossem utilizadas como planos de corte:

$$x_1 + x_2 \leq 5 \quad \text{e} \quad x_2 \leq 4$$

então o problema aumentado:

$$\begin{aligned} \min S(x) &= -10 x_1 - 11 x_2 \\ \text{sujeito a: } & 10 x_1 + 12 x_2 \leq 59 \\ & x_1 + x_2 \leq 5 \\ & x_2 \leq 4 \\ & x \geq 0 \end{aligned}$$

teria como solução a ponto ótimo do problema original $x = [1 \ 4]^T$, pois as restrições formam uma casca convexa de suas soluções inteiras.

7.2.2 Método de *branch and bound*

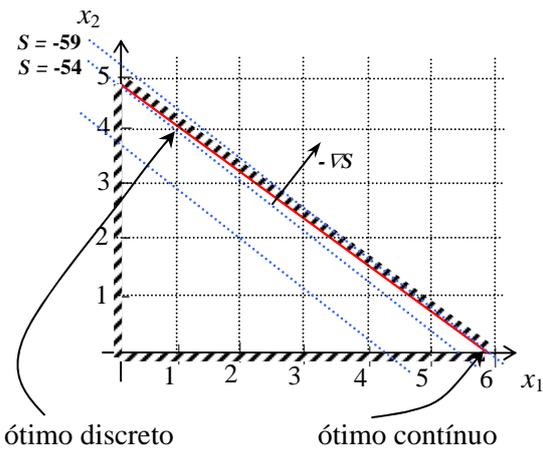
O método de *branch and bound* está baseado nas idéias chaves de *separação*, *relaxação* e *sondagem*.

Separação:

Denotando o problema MILP abaixo por (P) e o conjunto de suas soluções viáveis por FS(P).

$$P \left\{ \begin{aligned} \min S(x,y) &= c^T x + d^T y \\ \text{sujeito a: } & A x + B y \leq b \\ & x \geq 0 \\ & x \in X \subseteq \mathfrak{R}^n, \quad y \in Y = \{0,1\}^q \end{aligned} \right.$$

Então o conjunto de subproblemas $(P_1), (P_2), \dots, (P_n)$ de (P) é uma separação de (P) se:



- (i) uma solução viável de qualquer subproblema $(P_1), (P_2), \dots, (P_n)$ é também uma solução viável de (P) ;
- (ii) cada solução viável de (P) é uma solução viável de exatamente um de seus subproblemas.

Neste caso, o problema (P) é chamado de problema *Pai* e os subproblemas $(P_1), (P_2), \dots, (P_n)$ são chamados de problemas *Filhos*.

Uma questão importante no método de *branch and bound* é de como gerar uma separação do problema (P) . A maneira geralmente utilizada é a introdução de restrições contraditórias em uma única variável binária (ou inteira) a cada estágio. Por exemplo, selecionando a variável binária y_1 de (P) , pode-se separá-lo, ou ramificá-lo (*branching*), em dois subproblemas (P_1) e (P_2) :

$$\left\{ \begin{array}{l} \min S(x,y) = c^T x + d^T y \\ \text{sujeito a: } A x + B y \leq b \\ x \geq 0 \\ y_1 = 0 \\ x \in X \subseteq \mathfrak{R}^n \\ y \in Y = \{0,1\}^q \end{array} \right. \quad P_1 \qquad \left\{ \begin{array}{l} \min S(x,y) = c^T x + d^T y \\ \text{sujeito a: } A x + B y \leq b \\ x \geq 0 \\ y_1 = 1 \\ x \in X \subseteq \mathfrak{R}^n \\ y \in Y = \{0,1\}^q \end{array} \right. \quad P_2$$

Construindo-se desta maneira uma *árvore binária*. Outra maneira de gerar separações, aplicável para problemas com restrições generalizadas de limite superior, ou seja:

$$\sum_{j \in J} y_j = 1$$

é igualar a zero parte desta soma e gerar outra soma mutuamente excludente. Por exemplo, se o problema (P) possuir a seguinte restrição:

$$y_1 + y_2 + y_3 + y_4 = 1$$

então ao subproblema (P_1) seria adicionado a restrição $y_1 + y_2 = 0$ e ao subproblema (P_2) a restrição $y_3 + y_4 = 0$.

Relaxação:

Um problema de otimização, denotado por (RP) , é uma relaxação do problema (P) se o conjunto de soluções viáveis de (P) é um subconjunto de soluções viáveis de (RP) , isto é,

$$FS(P) \subseteq FS(RP)$$

Deste modo, se (RP) não tem solução viável, então (P) também não tem. Além disto, se z_P é a solução ótima de (P) e z_{RP} é a solução ótima de (RP) , então:

$$z_{RP} \leq z_P$$

ou seja, a solução do problema relaxado fornece um *limite inferior* para a solução do problema original. Naturalmente, se a solução ótima de (RP) é viável para (P), então ela é a solução ótima de (P).

A forma de relaxação mais freqüentemente utilizada em problemas MILP é tornar as variáveis binárias em variáveis contínuas: $0 \leq y \leq 1$, gerando um problema LP relaxado. Outra forma de relaxação é remoção de algumas restrições de (P). Porém, existe um compromisso entre a relaxação e a qualidade do limite inferior (z_{RP}) para a solução ótima. Em geral, quanto mais fácil for a solução do problema relaxado (maior relaxamento), maior será a diferença entre z_{RP} e z_P .

Sondagem:

Seja (CS) um subproblema candidato para a solução de (P). Deseja-se então determinar se a região viável de (CS), $FS(CS)$, contém uma solução ótima de (P), para então encontrá-la. Este subproblema (CS) será considerado *sondado* se uma das seguintes condições for satisfeita:

- (i) estiver garantido que $FS(CS)$ não pode conter uma solução melhor que a melhor solução já encontrada em estágios anteriores (ou solução titular, z^*). Se nenhuma solução viável havia sido encontrada, então $z^* = \infty$;
- (ii) uma solução ótima de (CS) foi encontrada, z_{CS} .

Em qualquer uma destas situações o subproblema (CS) não necessita de novas separações.

Denotando por (RCS) uma relaxação do subproblema (CS), e z_{RCS} a sua solução ótima, então os critérios gerais de sondagem em um algoritmo de *branch and bound*, baseado em relaxação, são:

- 1) Se (RCS) não possui solução viável, então (CS) também não possui e pode ser considerado sondado;
- 2) Se $z_{RCS} \geq z^*$ então (CS) está sondado;
- 3) Se uma solução ótima de (RCS) é viável para (CS), então ela é também uma solução ótima de (CS), portanto o problema (CS) pode ser considerado sondado. Neste caso, a solução também é viável para (P) e se $z_{RCS} < z^*$, então a solução titular é substituída por esta nova solução, senão z_{RCS} é um *limite superior* para o problema.

Note que é possível ter $z_{CS} \geq z^* > z_{RCS}$, e neste caso (CS) não pode ser considerado sondado, sendo z_{RCS} um *limite inferior* para o problema. Portanto, quanto menor a

diferença entre a solução do problema (RCS) e o problema (CS), mais freqüentemente estes critérios serão utilizados para eliminar ramificações. O sucesso dos algoritmos de *branch and bound* está baseado no percentual de eliminação de subproblemas e no esforço requerido para resolver os subproblemas candidatos.

Um algoritmo genérico para os métodos de *branch and bound* pode ser descrito como segue:

algoritmo

- 1) Inicializar a lista de subproblemas (CS) = (P), ou árvore binária, $z^* = \infty$;
- 2) **Se** a lista de subproblemas (CS) estiver vazia, **então** FIM (se $z^* = \infty$, então não existe solução viável);
- 3) Selecionar um candidato da lista, tornado-o candidato (CS) corrente;
- 4) Selecionar e resolver uma relaxação (RCS) do (CS) corrente, obtendo a solução z_{RCS} ;
- 5) Aplicar os três critérios de sondagem:
 - (i) **Se** (RCS) é inviável, **então** o (CS) corrente não tem solução viável e (ir para 2);
 - (ii) **Se** $z_{RCS} \geq z^*$, **então** o (CS) corrente não tem solução viável melhor que z^* e (ir para 2);
 - (iii) **Se** a solução ótima de (RCS) é viável para (CS) e $z_{RCS} < z^*$, **então** $z^* \leftarrow z_{RCS}$ e (ir para 2);
- 6) Separar o (CS) corrente e adicionar os seus subproblemas filhos na lista de subproblemas (CS). (ir para 2).

Existem três alternativas principais para selecionar os candidatos da árvore binária:

- a) Busca em primeira profundidade (*depth-first search*) com retrocesso (LIFO: *Last-In-First-Out*). Técnica padrão da maioria dos algoritmos;
- b) Busca em primeira largura (*breadth-first search*);
- c) Busca pelo melhor limite.

Na técnica de busca em primeira profundidade o próximo (CS) selecionado é um dos filhos do (CS) corrente. Quando um nó é considerado sondado, a busca é retrocedida até o ancestral mais próximo que tenha um filho que ainda não foi selecionado. Esta técnica tem como característica uma maior facilidade de re-otimização e compactação de informação, mas pode requerer um número elevado de subproblemas (CS).

Na técnica de busca em primeira largura, todos os nós de um mesmo nível da árvore binária são considerados antes dos nós de níveis subsequentes. Esta técnica é geralmente utilizada como base de heurísticas para a seleção de nós para a melhoria dos limites inferiores.

Na técnica de busca pelo melhor limite, o próximo (CS) escolhido da lista é aquele que possui o menor limite inferior. Esta técnica resulta em um menor número de subproblemas (CS), mas apresenta uma maior heterogeneidade no caminho para o ótimo. Nota-se que existe um compromisso entre o número de (CS) gerados e o controle da otimização, dificultando a seleção da melhor estratégia.

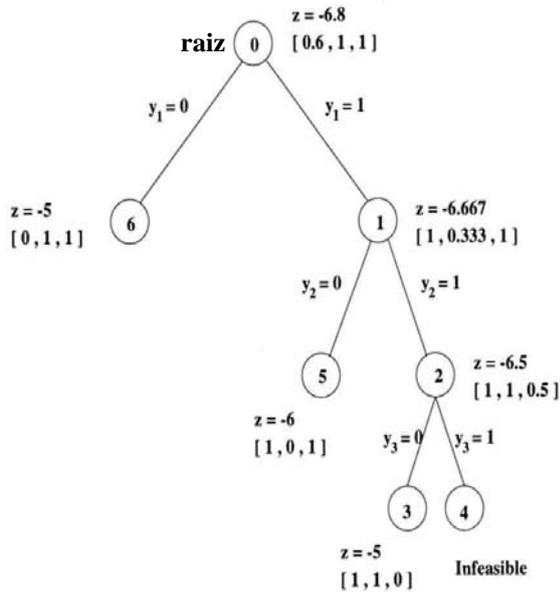
Exemplo 7.3: Para ilustrar o método de *branch and bound* com relaxação das variáveis binárias, seja o seguinte problema:

$$\begin{aligned} \min S(x,y) &= 2x_1 - 3y_1 - 2y_2 - 3y_3 \\ \text{sujeito a: } &x_1 + y_1 + y_2 + y_3 \geq 2 \\ &10x_1 + 5y_1 + 3y_2 + 4y_3 \leq 10 \\ &x_1 \geq 0 \\ &y \in Y = \{0,1\}^3 \end{aligned}$$

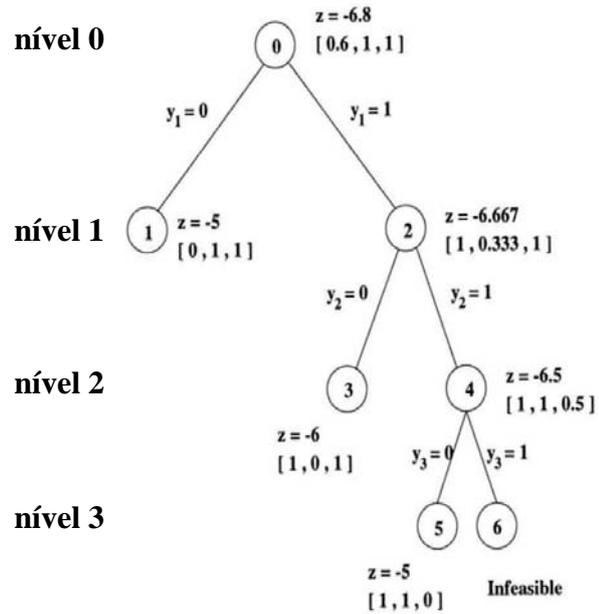
onde a solução ótima ocorre no ponto $x^* = 0$ e $y^* = [1 \ 0 \ 1]^T$, com o valor da função objetivo $S(x^*,y^*) = z^* = -6$. Resolvendo o problema LP relaxado na raiz da árvore binária, tem-se a solução:

$$x = 0, \quad y = [0,6 \ 1 \ 1]^T \quad \text{e} \quad z = -6,8$$

que representa um limite inferior para o problema MILP. As árvores binárias para as técnicas de busca em primeira profundidade e busca em primeira largura são mostradas abaixo, onde o número dentro do nó indica a seqüência do subproblema (CS).



Busca em primeira profundidade.



Busca em primeira largura.

Observa-se que na busca em primeira profundidade, a solução ótima é obtida no nó 5, sendo necessário considerar 6 nós da árvore, além da raiz. Por outro lado, na busca em primeira largura, o ótimo é obtido no nó 3, mas é necessário considerar o mesmo número de nós que no caso anterior para certificar-se da otimalidade do nó. Os limites inferiores e superiores em cada nó da árvore binária, para as duas técnicas, são apresentados na tabela abaixo.

nó	primeira profundidade				primeira largura			
	nível	inferior	superior	z^*	nível	inferior	superior	z^*
0	0	-6,8	∞	∞	0	-6,8	∞	∞
1	1	-6,667	∞	∞	1	-6,8	-5	-5
2	2	-6,667	∞	∞	1	-6,667	-5	-5
3	3	-6,5	-5	-5	2	-6,667	-6	-6
4	3	-6,5	-5	-5	2	-6,5	-6	-6
5	2	-6,667	-6	-6	3	-6,5	-6	-6
6	1	-6	-6	-6	3	-6	-6	-6

7.3 Programação não linear inteira mista (MINLP)

A formulação MINLP com variáveis 0–1 é dada por:

$$\begin{aligned} \min S(x,y) \\ \text{sujeito a: } h_j(x,y) = 0 \quad , \quad j = 1, 2, \dots, m \\ g_j(x,y) \leq 0 \quad , \quad j = 1, 2, \dots, p \\ x \in X \subseteq \mathfrak{R}^n, \quad y \in Y = \{0,1\}^q \end{aligned}$$

Os algoritmos existentes para resolver problemas MINLP podem ser classificados em quatro categorias principais:

- *Branch and Bound*
- Decomposição Generalizada de Benders (GBD)
- Aproximações Externas (OA)
- Lógica e disjunções

O método de *branch and bound*, aplicado ao problema MINLP, inicia resolvendo o correspondente problema NLP relaxado, para então realizar a enumeração implícita e seleção dos subproblemas através das técnicas descritas na seção anterior (busca em primeira profundidade ou busca em primeira largura). Os critérios de sondagem são aplicados às soluções dos subproblema NLP relaxados a cada nó, que geram limites inferiores e superiores.

7.3.1 Decomposição Generalizada de Benders (GBD)

Geoffrion (1972) generalizou a aproximação proposta por Benders (1962), para explorar a estrutura de problemas de programação MINLP. A idéia básica é gerar, para cada iteração, um limite superior e um limite inferior sobre a busca da solução do problema MINLP. O limite superior resulta a partir de um problema primal, enquanto o limite inferior resulta a partir de um problema mestre.

Com o decorrer das iterações verifica-se que a seqüência de limites superiores não cresce e a seqüência de limites inferiores não decresce, logo, as seqüências convergem em um número finito de iterações.

Problema primal (NLP)

O problema primal corresponde ao MINLP com as variáveis y fixas em valores inteiros:

$$P(y^k) \begin{cases} \min S(x, y^k) \\ \text{sujeito a: } h(x, y^k) = 0 \\ \phantom{\text{sujeito a: }} g(x, y^k) \leq 0 \\ \phantom{\text{sujeito a: }} x \in X \subseteq \mathfrak{R}^n \end{cases}$$

onde $y^k \in \{0,1\}^q$ corresponde ao vetor y fixo na iteração k .

Se o problema primal é viável, então sua solução fornece um *limite superior*, $S(x^k, y^k)$, para a função objetivo e os respectivos multiplicadores de Lagrange das restrições, λ^k e μ^k . A função de Lagrange para este problema é dada por:

$$L(x, y, \lambda^k, \mu^k) = S(x, y) + (\lambda^k)^T h(x, y) + (\mu^k)^T g(x, y)$$

Se o problema primal é inviável, então o seguinte problema de viabilidade geral, FP, (Fletcher e Leyffer, 1994) é resolvido:

$$F \begin{cases} \min \sum_{i \in I^l} w_i g_i^+(x, y^k) \\ \text{sujeito a: } h(x, y^k) = 0 \\ \phantom{\text{sujeito a: }} g_i(x, y^k) \leq 0, \quad i \in I \\ \phantom{\text{sujeito a: }} x \in X \subseteq \mathfrak{R}^n \end{cases}$$

onde $w_i \geq 0$ são pesos associados a cada restrição, $g_i^+(x, y^k) = \max [0, g_i(x, y^k)]$, I é o conjunto de restrições de desigualdade viáveis, e I^l é o conjunto de restrições de desigualdade inviáveis. A solução do problema FP fornece informações sobre os multiplicadores de Lagrange, $\bar{\lambda}^k$ e $\bar{\mu}^k$, sendo a função de Lagrange para este caso dada por:

$$\bar{L}(x, y, \bar{\lambda}^k, \bar{\mu}^k) = (\bar{\lambda}^k)^T h(x, y) + (\bar{\mu}^k)^T g(x, y)$$

Problema mestre

O problema mestre é derivado via teoria de dualidade não linear, faz uso dos multiplicadores de Lagrange obtido no problema primal e sua solução produz informação sobre o limite inferior, caracterizada pelas seguintes idéias chaves:

- (i) projeção do problema MINLP sobre o espaço y (i.e., $y \in Y \cap V$);
- (ii) representação dual de V , onde $V = \{y \mid h(x, y) = 0, g(x, y) \leq 0, x \in X\}$;
- (iii) representação dual da projeção do problema MINLP sobre o espaço y .

$$M \left\{ \begin{array}{l} \min \alpha \\ \text{sujeito a: } \alpha \geq \inf_{x \in X} L(x,y,\lambda,\mu) , \quad \forall \lambda , \forall \mu \geq 0 \\ \\ 0 \geq \inf_{x \in X} \bar{L}(x,y,\bar{\lambda},\bar{\mu}) , \quad \forall (\bar{\lambda} , \bar{\mu}) \in \Lambda \\ \\ y \in Y = \{0,1\}^q , \quad \alpha \in \mathfrak{R} \end{array} \right.$$

onde $\Lambda = \{ \bar{\lambda} \in \mathfrak{R}^m, \bar{\mu} \in \mathfrak{R}^p \mid \bar{\mu} \geq 0, \sum_{i=1}^p \bar{\mu}_i = 1 \}$. Observa-se que o problema mestre

(M) tem como restrição dois problemas de otimização, que precisam ser considerados para todos λ e $\mu \geq 0$ (primal viável) e para todos $(\bar{\lambda} , \bar{\mu}) \in \Lambda$ (primal inviável), resultando em um problema com um número elevado de restrições. A estratégia natural de resolver este problema é através de relaxação das restrições, gerando um problema mestre relaxado (RM).

$$RM \left\{ \begin{array}{l} \min \alpha \\ \text{sujeito a: } \alpha \geq \inf_{x \in X} L(x,y,\lambda^k,\mu^k) , \quad k = 1,2,\dots,N_p \\ \\ 0 \geq \inf_{x \in X} \bar{L}(x,y,\bar{\lambda}^k,\bar{\mu}^k) , \quad k = 1,2,\dots,N_{FP} \\ \\ y \in Y = \{0,1\}^q , \quad \alpha \in \mathfrak{R} \end{array} \right.$$

onde N_p é o número de problemas primais viáveis e N_{FP} é o número de problemas primais inviáveis resolvidos.

7.3.2 Aproximações Externas (OA)

Duran e Grossmann (1986) propuseram um algoritmo para tratar de problemas MINLP na forma abaixo, cuja idéia básica é similar ao GBD, onde a cada iteração gera-se um limite superior e um limite inferior da solução MINLP. O limite superior resulta a partir da solução viável do respectivo problema primal com y fixo.

$$\begin{array}{l} \min S(x,y) = c^T y + f(x) \\ \text{sujeito a: } g(x) + B y \leq 0 \\ x \in X \subseteq \mathfrak{R}^n, y \in Y = \{0,1\}^q \end{array}$$

O limite inferior resulta a partir da solução do problema mestre. O problema mestre é derivado pela projeção do MINLP no espaço y e pela aproximação externa (linearização) das restrições e função objetivo não linear ao redor da solução primal x^k . Ou seja, consiste de suportes lineares válidos, e, conseqüentemente, relaxamentos de funções não lineares para todos pontos x^k que resultam fixando $y = y^k \in Y$.

Se o problema primal é viável, então as aproximações externas são dadas pela linearização das funções convexas $f(x)$ e $g(x)$ no ponto x^k :

$$f(x) \geq f(x^k) + \nabla^T f(x^k) (x - x^k)$$

$$g(x) \geq g(x^k) + \nabla^T g(x^k) (x - x^k)$$

Se o problema primal é inviável, então resolve-se o respectivo problema de viabilidade geral, para então linearizar as restrições no ponto obtido, \bar{x}^k :

$$g(x) \geq g(\bar{x}^k) + \nabla^T g(\bar{x}^k) (x - \bar{x}^k)$$

Então o problema mestre pode ser formulado como:

$$M \left\{ \begin{array}{l} \min c^T y + \alpha \\ \text{sujeito a: } \alpha \geq f(x^k) + \nabla^T f(x^k) (x - x^k), \quad \forall k \in P(y^k) \\ 0 \geq g(x^k) + \nabla^T g(x^k) (x - x^k) + B y, \quad \forall k \in P(y^k) \\ 0 \geq g(\bar{x}^k) + \nabla^T g(\bar{x}^k) (x - \bar{x}^k) + B y, \quad \forall k \in FP(y^k) \\ x \in X, \quad y \in Y = \{0,1\}^q, \quad \alpha \in \mathfrak{R} \end{array} \right.$$

A aproximação natural da resolução do problema mestre é o relaxamento, que considera para cada iteração os suportes lineares das restrições e função objetivo ao redor de todos os pontos linearizados previamente. Deste modo, para cada iteração um novo conjunto de restrições e suportes lineares são adicionadas, os quais melhoram o relaxamento e por esta razão melhoram o limite inferior. Observe que o problema mestre corresponde a um problema MILP.

Com as iterações ocorridas, duas seqüências de limites superiores e inferiores são gerados, que não aumentam nem diminuem, respectivamente. Então, é mostrado que essas duas seqüências convergem dentro de uma tolerância ϵ com um número finito de iterações.

7.3.3 Aproximações Externas com Relaxamento de Igualdade (OA/ER)

Para lidar com restrições de igualdade não lineares explicitamente da forma $h(x) = 0$, Kocis e Grossmann (1987) propuseram uma aproximação externa com relaxamento de igualdade para a seguinte classe de problemas MINLP.

$$\min S(x,y) = c^T y + f(x)$$

$$\text{sujeito a: } A x + B y \leq b$$

$$h(x) = 0$$

$$g(x) \leq 0$$

$$x \in X \subseteq \mathfrak{R}^n, \quad y \in Y = \{0,1\}^q$$

A idéia básica do OA/ER é relaxar as restrições de igualdade não lineares em desigualdades e subsequente aplicar o algoritmo OA. O relaxamento das igualdades não lineares é baseada no sinal dos multiplicadores de Lagrange associados

a este relaxamento quando o problema primal (com y fixo) é resolvido. Se um multiplicador λ_i é positivo então a correspondente igualdade não linear $h_i(x)=0$ é relaxada como $h_i(x) \leq 0$. Se um multiplicador λ_i é negativo, então a igualdade não linear é relaxada como $-h_i(x) \leq 0$. Entretanto, se $\lambda_i = 0$, então a restrição de igualdade não linear associada é escrita como $0 \cdot h_i(x) = 0$, a qual implica que esta restrição pode ser eliminada. Tendo transformado as igualdades não lineares em desigualdades, na seqüência formula-se o problema mestre baseado sobre os princípios da aproximação OA discutida anteriormente.

$$\text{M} \left\{ \begin{array}{l} \min c^T y + \alpha \\ \text{sujeito a: } \alpha \geq f(x^k) + \nabla^T f(x^k) (x - x^k), \quad \forall k \in \text{P}(y^k) \\ \quad 0 \geq g(x^k) + \nabla^T g(x^k) (x - x^k), \quad \forall k \in \text{P}(y^k) \\ \quad 0 \geq T^k [h(x^k) + \nabla^T h(x^k) (x - x^k)], \quad \forall k \in \text{P}(y^k) \\ \quad 0 \geq g(\bar{x}^k) + \nabla^T g(\bar{x}^k) (x - \bar{x}^k), \quad \forall k \in \text{FP}(y^k) \\ \quad 0 \geq \bar{T}^k [h(\bar{x}^k) + \nabla^T h(\bar{x}^k) (x - \bar{x}^k)], \quad \forall k \in \text{FP}(y^k) \\ Ax + By \leq b \\ x \in X, \quad y \in Y = \{0,1\}^q, \quad \alpha \in \mathfrak{R} \end{array} \right.$$

onde T^k é a matriz diagonal dos sinais dos multiplicadores de Lagrange.

O algoritmo OA/ER tem a vantagem de exigir um número menor de iterações que o GBD, porém o tamanho de seus problemas mestres MILP é considerado grande em relação ao GBD.

7.3.4 Aproximações Externas com Relaxamento de Igualdade e Penalidade Aumentada (OA/ER/AP)

Os algoritmos *branch and bound*, GBD e OA/ER requerem que algumas formas de suposições de convexidade sejam satisfeitas de modo a garantir a otimização global do problema MINLP. Viswanathan e Grossmann (1990) propuseram o algoritmo OA/ER/AP (conhecido como DICOPT - *DI*crete and *CO*ntinuous *OPT*imizer), que é um variante do algoritmo OA/ER, com objetivo de evitar as limitações impostas pela suposição de convexidade feita no algoritmo OA/ER.

Este algoritmo recorre a mesma formulação do OA/ER e introduz a função penalidade aumentada nos subproblemas de limite inferior da aproximação OA/ER.

Como a suposição de convexidade não é imposta em OA/ER/AP, então:

- (i) a equivalência de $h(x) = 0$ e $T^k h(x) \leq 0$ pode não se manter, onde $h(x)$ e $T^k h(x)$ são as igualdades não lineares e as igualdades relaxadas, respectivamente (T^k é a matriz diagonal dos sinais dos multiplicadores de Lagrange);

- (ii) as linearizações podem não representar suportes válidos;
- (iii) o problema mestre pode não produzir um limite inferior válido sobre a solução do MINLP.

A idéia básica em OA/ER/AP é limitar (i), (ii) e (iii) pelo relaxamento das linearizações no problema mestre, permitindo-os de serem violados e utilizando uma aproximação tipo penalidade que penaliza essas violações das funções suportes. Então, o problema mestre para o método AO/ER/AP pode ser formulado como:

$$\begin{array}{l}
 \text{M} \left\{ \begin{array}{l}
 \min \quad c^T y + \alpha + w_s^T s + \sum_j (w_{pj})^T p_j + \sum_j (w_{qj})^T q_j \\
 \text{sujeito a: } \alpha + s_k \geq f(x^k) + \nabla^T f(x^k) (x - x^k), \quad \forall k \in P(y^k) \\
 p_k \geq g(x^k) + \nabla^T g(x^k) (x - x^k), \quad \forall k \in P(y^k) \\
 q_k \geq T^k [h(x^k) + \nabla^T h(x^k) (x - x^k)], \quad \forall k \in P(y^k) \\
 0 \geq g(\bar{x}^k) + \nabla^T g(\bar{x}^k) (x - \bar{x}^k), \quad \forall k \in FP(y^k) \\
 0 \geq \bar{T}^k [h(\bar{x}^k) + \nabla^T h(\bar{x}^k) (x - \bar{x}^k)], \quad \forall k \in FP(y^k) \\
 Ax + By \leq b \\
 x \in X, \quad y \in Y = \{0,1\}^q, \quad \alpha \in \mathfrak{R}, \quad s_k, p_k, q_k \geq 0
 \end{array} \right.
 \end{array}$$

onde w_s , w_p e w_q são os pesos para as variáveis de folga das restrições. Ou seja, as violações das linearizações são permitidas pela introdução de variáveis de folga, enquanto a penalidade das violações é introduzida como um conjunto adicional de termos na função objetivo que consiste de variáveis de folga multiplicadas por fatores de peso positivos. Deste modo, por causa do relaxamento de restrições, a região viável é expandida e, conseqüentemente, a possibilidade de partes das regiões viáveis serem cortadas devido a linearizações inválidas é reduzida.

7.3.5 Aproximação Externa Generalizada (GOA)

Fletcher e Leyffer (1994) estenderam a aproximação OA para problemas MINLP do tipo abaixo, introduzindo funções penalidades exatas, criando o algoritmo GOA.

$$\begin{array}{l}
 \min S(x,y) \\
 \text{sujeito a: } g(x,y) \leq 0 \\
 x \in X \subseteq \mathfrak{R}^n, \quad y \in Y = \{0,1\}^q
 \end{array}$$

para S e g convexas e continuamente diferenciáveis e X convexo, compacto e não vazio. O GOA é similar ao OA, com as principais diferenças sendo:

- (i) tratamento da inviabilidade;

- (ii) problema mestre considera as inviabilidades explicitamente;
- (iii) tratamento unificado de funções penalidades exatas.

Se o problema primal $P(y^k)$ é inviável, então o seguinte problema de viabilidade geral é dado por:

$$\begin{aligned} \min_x \quad & \sum_{i \in I^l} w_i g_i^+(x, y^k) \\ \text{sujeito a} \quad & g_i(x, y^k) \leq 0, \quad i \in I \\ & x \in X \subseteq \mathcal{R}^n \end{aligned}$$

onde w_i são os pesos, $g_i^+(x, y^k) = \max [0, g_i(x, y^k)]$, I é o conjunto de restrições de desigualdade viáveis, e I^l é o conjunto de restrições de desigualdade inviáveis. Note que o problema acima tem dentro da função objetivo um somatório somente das restrições de desigualdade inviáveis.

Se o problema primal $P(y^k)$ é inviável de modo que x^k resolve o problema de viabilidade geral e tem $\sum_{i \in I^l} w_i g_i^+(x^k, y^k) > 0$, então y^k é inviável dentro das restrições:

$$0 \geq g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix}, \quad \forall i \in \{I^l \cup I\} \quad \forall x \in X$$

Observe que as restrições acima são as linearizações das restrições de desigualdade não lineares em torno do ponto (x^k, y^k) . Então a inviabilidade no problema primal $P(y^k)$ significa que os cortes de linearizações de desigualdade viável e inviável em torno de (x^k, y^k) são violados.

A formulação do problema mestre que considera a inviabilidade explicitamente, é baseada nas seguintes idéias:

- a) projeção do problema MINLP sobre o espaço y ;
- b) aproximação externa da função objetivo e da região viável, com tratamento das inviabilidades do problema primal.

7.3.6 Decomposição Cruzada Generalizada (GCD)

Holmberg (1990) generalizou a aproximação proposta por Van Roy (1983) para a classe de problemas de otimização da forma:

$$\begin{aligned} \min S(x,y) \\ \text{sujeito a: } \quad & h(x,y) = 0 \\ & g(x,y) \leq 0 \end{aligned}$$

$$x \in X \subseteq \mathfrak{R}^n, \quad y \in Y = \{0,1\}^q$$

para S , g e h convexas para cada y fixo, limitadas e Lipschitz contínuas e X convexo, compacto e não vazio. Outra condição é que a otimização com respeito a x da função de Lagrange pode ser realizada independentemente de y .

A decomposição cruzada generalizada consiste de duas fases. A fase I é composta de subproblemas primal e dual e a fase II é constituída de problemas mestres e testes de convergência apropriados.

Na fase I, o subproblema primal produz um limite superior sobre a solução procurada do MINLP e multiplicadores de Lagrange λ^k , μ^k para o subproblema dual. O subproblema dual produz um limite inferior sobre a solução do MINLP e fornece y^k para o subproblema primal. Ambos subproblemas, primal e dual, geram cortes para o problema mestre na fase II. Para cada iteração do GCD, os problemas primal e dual são resolvidos, e um teste de convergência primal é aplicado sobre y^k , enquanto o teste de convergência dual é aplicada sobre λ^k , μ^k . Se qualquer teste de convergência falhar, então entra-se na fase II, que caracteriza a solução do problema mestre e retorna subseqüentemente para a fase I. A idéia chave é fazer uso extenso da fase I e limitar tanto quanto possível o uso da fase II para a aplicação de testes de convergência apropriados. Isto é porque o problema mestre é mais difícil e demanda um maior tempo computacional que os problemas da fase I.

7.3.7 Branch and Bound Espacial (SBB)

Os algoritmos GBD e OA têm a limitação do tamanho do problema mestre (MILP) que cresce a medida que as iterações ocorrem, devido as restrições adicionais, sendo este o maior problema quando o MINLP original tem um número grande de variáveis inteiras. O tempo usado para resolver o problema mestre cresce conforme as iterações ocorrem, enquanto o tempo para os subproblemas NLP mantém-se na mesma ordem de magnitude.

Branch and Bound Espacial Linearizado

Quesada e Grossmann (1992) propuseram um algoritmo para evitar os problemas relatados acima, melhorando a eficiência da solução dos problemas MINLP convexas e reduzindo o trabalho computacional exigido para resolver os problemas mestres MILP.

O algoritmo SBB (*Spatial Branch and Bound*) consiste da busca em árvore sobre o espaço de variáveis binárias. O problema mestre MILP é definido dinamicamente durante a busca em árvore para reduzir o número de nós que necessitam ser enumerados. Uma busca *branch and bound* é conduzida para

determinar limites inferiores na solução de subproblemas LP até encontrar soluções inteiras viáveis. Para estes nós, subproblemas NLP são resolvidos, determinando limites superiores e novas aproximações lineares, as quais são usadas para estender a representação linear dos nós abertos na árvore de busca. Estas aproximações lineares podem ser feitas de diversas formas.

A seguir descreve-se detalhadamente o algoritmo de Quesada e Grossmann (1992) com modificações propostas por Pereira e Secchi (2000). Considera-se os problemas MINLP da forma:

$$\begin{aligned} z &= \min S(x,y) = c^T y + f(x) \\ \text{sujeito a: } & g(x) + B y \leq 0 \\ & x \in X \subseteq \mathfrak{R}^n, y \in Y = \{0,1\}^q \end{aligned}$$

Passo 1 – Arbitram-se valores iniciais para as variáveis binárias “y” e variáveis contínuas “x”. Durante a implementação, existe a opção do próprio algoritmo escolher estes valores, caso não sejam arbitrados.

Passo 2 – Fixando $y = y^0$, o problema MINLP transforma-se num subproblema NLP, ou seja:

$$\begin{aligned} z &= \min_x c^T y^0 + f(x) \\ \text{sujeito a } & g(x) + B y^0 \leq 0 \\ & x \in X \subseteq \mathfrak{R}^n \end{aligned}$$

O problema NLP é resolvido, encontrando uma solução (x^0, y^0) , a qual corresponde a um determinado valor “z” da função objetivo. Considera-se $z_u = z$, o limite superior para a solução ótima do problema MINLP.

Se a violação de restrição for maior que ε_c , onde “ ε_c ” é a tolerância de violação de restrição, então faz-se $z_u = \infty$, o que significa que este primeiro NLP é considerado inviável.

Na implementação, quando y^0 não for dado, existe a possibilidade de ser resolvido um NLP com condições de integralidade sobre as variáveis binárias relaxadas. Neste caso é resolvido um NLP relaxado, resultando uma solução (x^0, y^0) . Se y^0 não for inteiro, então y^0 sofre um arredondamento e é resolvido um outro NLP.

Passo 3 – As funções não lineares do problema MINLP são linearizadas, usando a solução ótima x^0 do subproblema NLP, resultando no seguinte problema MILP.

$$\begin{aligned} z &= \min_{x,y,\alpha} \alpha \\ \text{sujeito a } & \alpha \geq c^T y + f(x^0) + \nabla^T f(x^0) (x - x^0) \\ & B y + g(x^0) + \nabla^T g(x^0) (x - x^0) \leq 0 \end{aligned}$$

$$\alpha \in \mathfrak{R}, \quad x \in X \subseteq \mathfrak{R}^n, \quad y \in Y = \{0,1\}^q$$

Passo 4 – O problema MILP transforma-se num problema LP, com as condições de integralidade sobre as variáveis binárias relaxadas. A solução deste LP é considerada um limite inferior, z^l , para a solução ótima do problema MINLP. Caso $z^l + \varepsilon_z \geq z^u$, onde “ ε_z ” é a tolerância da função objetivo, então este problema é desconsiderado. Se $z^l < z^u$, este primeiro problema LP é armazenado como o primeiro nó da árvore de busca e se a solução deste LP para as variáveis binárias “y” for inteira, segue-se para o passo 8.

Passo 5 – Se não existe mais LPs para serem resolvido, ou seja, nenhum nó aberto na árvore, então o limite superior corrente z^u é a solução ótima do problema MINLP e terminou o algoritmo.

Passo 6 – Se a solução do LP para variáveis binárias não for inteira, então haverá uma ramificação na árvore de busca com a criação de dois nós filhos, isto é, dois problemas LPs. A criação desses LPs será feita da seguinte forma, seleciona-se dentro da solução do último $(LP)_j^k$, onde “j” é o número do nó e “k” é o número do nó pai, o “ y_i ” cujo valor fracionário encontra-se mais distante dos extremos 0 e 1 e adiciona-se as restrições $y_i = 0$ e $y_i = 1$ para os subproblemas $(LP)_{j+1}^j$ e $(LP)_{j+2}^j$, respectivamente. Substitui-se o problema pelos dois subproblemas filhos na lista de LPs.

Passo 7 – Se no final da lista de LPs tem-se dois problemas filhos para serem resolvidos que tem o mesmo problema pai, então faz-se o seguinte:

- i) resolve-se o $(LP)_{j+2}^j$. Se o valor da função objetivo $z_{j+2} + \varepsilon_z \geq z^u$, este problema é desconsiderado. Caso z_{j+2} seja uma solução inteira, segue-se para o passo 8;
- ii) resolve-se o $(LP)_{j+1}^j$. Se o valor da função objetivo $z_{j+1} + \varepsilon_z \geq z^u$, este problema é desconsiderado. Caso z_{j+1} seja uma solução inteira, segue-se para o passo 8;
- iii) se $z_{j+2} \geq z_{j+1}$, inverte-se $(LP)_{j+2}^j$ e $(LP)_{j+1}^j$, no armazenamento de LPs.

Caso contrário (isto é, existe somente um problema filho), resolve-se o $(LP)_j^k$ e se $z_j + \varepsilon_z \geq z^u$, este deve ser desconsiderado, mas se z_j é uma solução inteira, segue-se para passo 8.

Retornar ao passo 5.

Passo 8 – Resolve-se um subproblema NLP fixando as variáveis binárias “y” da solução do problema LP para o nível do nó onde o problema LP se encontra. Se $z_{NLP} < z_U$, faz-se $z_U = z_{NLP}$. Se $z_{NLP} \geq z_U$, então continua-se com o limite superior anterior.

Passo 9 – A solução dos NLPs é usada para gerar restrições adicionais, isto é, a medida que é resolvido um subproblema NLP, esta solução é adicionada como restrição para a solução do próximo subproblema LP e assim sucessivamente. Estas restrições adicionais podem ser feitas de diversas formas como será comentado a seguir. Desconsiderar todos os nós de $(LP)_j$ para os quais $z_j \geq z_U$ e retornar para o passo 5.

Restrições Adicionais

As restrições adicionais, comentadas no passo 9 do algoritmo, a partir das informações da solução dos subproblemas NLP podem ser adicionadas para os nós abertos na árvore de busca de diversos modos.

1 - Aproximações Externas (OA)

O algoritmo OA consiste de uma seqüência de subproblemas NLP e problemas mestres MILP, onde as linearizações dos termos não lineares são derivadas para a solução ótima dos subproblemas NLP. O problema mestre MILP é dado por:

$$z = \min_{x,y,\alpha} \alpha$$

$$\text{sujeito a } \alpha \geq c^T y + f(x^k) + \nabla^T f(x^k) (x - x^k)$$

$$B y + g(x^k) + \nabla^T g(x^k) (x - x^k) \leq 0$$

$$\alpha \in \mathfrak{R}, \quad x \in X \subseteq \mathfrak{R}^n, \quad y \in Y = \{0,1\}^q$$

$$k = 1, 2, \dots, K_{(NLPs)}$$

onde x^k é a solução viável ou inviável do k -ésimo subproblema NLP.

2 - Decomposição Generalizada de Benders (GBD)

Na decomposição generalizada de Benders uma seqüência de subproblemas NLP e problemas mestres MILP no espaço das variáveis binárias é resolvido. Os subproblemas NLPs são gerados fixando as variáveis binárias no MINLP original para um dado valor y^k . O problema mestre é gerado projetando o problema original no espaço reduzido das variáveis binárias. Isto é feito derivando, para cada subproblema, uma função Lagrangeana parametrizada nas variáveis discretas “y”.

Aplicando as condições de Kuhn-Tucker em um subproblema NLP, consideradas em relação as variáveis não lineares “x”, tem-se:

$$\nabla f(x^k) + \nabla g(x^k) \lambda^k = 0$$

Multiplicando a equação acima por $(x - x^k)$, tem-se:

$$\nabla^T f(x^k) (x - x^k) + (\lambda^k)^T \nabla^T g(x^k) (x - x^k) = 0$$

Da segunda inequação do problema MILP, tem-se

$$\nabla^T g(x^k) (x - x^k) \leq -[B y + g(x^k)]$$

Então, $\nabla^T f(x^k) (x - x^k) - (\lambda^k)^T [B y + g(x^k)] \geq 0$. Logo, a primeira inequação do MILP reduz-se a:

$$\alpha \geq c^T y + f(x^k) + (\lambda^k)^T [B y + g(x^k)]$$

Desta forma, a formulação do problema mestre para os cortes de Benders é dado por:

$$z = \min_{y, \alpha} \alpha$$

$$\text{sujeito a } \alpha \geq c^T y + f(x^k) + (\lambda^k)^T [B y + g(x^k)], \quad k = 1, 2, \dots, K_{(NLPs \text{ viáveis})}$$

$$(\lambda^k)^T [B y + g(x^k)] \leq 0, \quad k = 1, 2, \dots, K_{(NLPs \text{ inviáveis})}$$

$$\alpha \in \mathfrak{R}, \quad y \in Y = \{0, 1\}^q$$

onde x^k e λ^k são as variáveis ótima primal e dual do subproblema NLP.

A vantagem das aproximações externas é que elas têm a representação estendida da região viável. Entretanto, têm a limitação que o número de colunas dos problemas LPs que são resolvidos para os nós pode tornar-se muito grande. Também em um número de casos, as novas linearizações não determinam aproximações muito reais da região viável não linear. Para evitar este problema, uma opção são os cortes de Benders, que em geral, não provêm cortes fortes.

Por estas razões, Quesada e Grossmann (1992) propuseram uma restrição aproximada, cuja a idéia básica é agregar as linearizações de funções não lineares, enquanto mantém as restrições lineares em ordem para fortalecer os cortes.

3 - Restrição Aproximada proposta por Quesada e Grossmann (1992)

A partir do problema MINLP original, considere a partição das variáveis contínuas em dois subconjuntos “w” e “v” tais que as restrições são divididas em restrições lineares e não lineares e as variáveis contínuas em variáveis lineares e não lineares.

$$z = \min c^T y + a^T w + r(v)$$

$$\text{sujeito a } C y + D w + t(v) \leq 0$$

$$E y + F w + G v \leq b$$

$$y \in Y, \quad w \in W, \quad v \in V$$

Nesta representação $f(x) = a^T w + r(v)$, $B^T = [C \ E]^T$, $g(x) = [D \ w + t(v) \ F \ w + G \ v]$ e $X = W \times V$.

O problema acima é reformulado com a adição de duas variáveis contínuas α e β para representar as partes lineares e não lineares da função objetivo, ou seja:

$$\begin{aligned} z &= \min \alpha \\ \text{sujeito a} \quad & C y + D w + t(v) \leq 0 \\ & r(v) \leq \beta \\ & E y + F w + G v \leq b \\ & c^T y + a^T w + \beta - \alpha = 0 \\ & \alpha, \beta \in \mathfrak{R}, \quad y \in Y, \quad w \in W, \quad v \in V \end{aligned}$$

Fazendo aproximações externas no problema acima para o ponto (w^k, v^k) gerado pelo k -ésimo subproblema NLP tem-se:

$$\begin{aligned} z &= \min \alpha \\ \text{sujeito a} \quad & C y + D w + t(v^k) + \nabla t(v^k) (v - v^k) \leq 0 \\ & r(v^k) + \nabla r(v^k) (v - v^k) \leq \beta \\ & E y + F w + G v \leq b \\ & c^T y + a^T w + \beta - \alpha = 0 \\ & \alpha, \beta \in \mathfrak{R}, \quad y \in Y, \quad w \in W, \quad v \in V \end{aligned}$$

Se as condições de Kuhn-Tucker do subproblema NLP são consideradas em relação as variáveis não lineares “ v ”, então:

$$\nabla r(v^k) + \nabla t(v^k) \lambda^k + G^T \mu^k = 0$$

que multiplicado por $(v - v^k)$, tem-se:

$$\nabla^T r(v^k) (v - v^k) + (\mu^k)^T G (v - v^k) = -(\lambda^k)^T \nabla t^T(v^k) (v - v^k)$$

Do problema linearizado tem-se:

$$\nabla t(v^k) (v - v^k) \leq -[C y + D w + t(v^k)]$$

então:

$$\nabla^T r(v^k) (v - v^k) \geq (\lambda^k)^T [C y + D w + t(v^k)] - (\mu^k)^T G (v - v^k)$$

Finalmente, substituindo esta expressão na segunda inequação do problema linearizado, tem-se o MILP:

$$\begin{aligned}
 z &= \min \alpha \\
 \text{sujeito a } \quad & \beta \geq r(v^k) + (\lambda^k)^T [C y + D w + t(v^k)] - (\mu^k)^T G (v - v^k) \\
 & E y + F w + G v \leq b \\
 & c^T y + a^T w + \beta - \alpha = 0 \\
 & \alpha, \beta \in \mathfrak{R}, \quad y \in Y, \quad w \in W, \quad v \in V \\
 & k = 1, 2, \dots, K_{(NLPs)}
 \end{aligned}$$

Observe que usando as aproximações lineares acima somente a primeira inequação é modificada para os nós abertos na árvore de busca, quando acontece uma solução inteira de um LP.

4 - Restrição Aproximada proposta por Pereira e Secchi (2000)

Considerando o problema MINLP original na forma:

$$\begin{aligned}
 z &= \min S(x,y) = c^T y + f(x) + \mu u \\
 \text{sujeito a: } \quad & g(x) + B y \leq u \\
 & x \in X \subseteq \mathfrak{R}^n, \quad y \in Y = \{0,1\}^q, \quad u \geq 0
 \end{aligned}$$

onde $\mu > 0$. Se $u > 0$ há violação de restrições para problemas inviáveis. Isto é, μu é uma penalização introduzida na função objetivo quando as restrições são violadas.

Fazendo aproximações externas para o ponto x^l , o qual foi gerado pelo primeiro subproblema NLP, tem-se o MILP

$$\begin{aligned}
 z &= \min_{x,y,\alpha,u} \alpha \\
 \text{sujeito a } \quad & \alpha \geq c^T y + f(x^l) + \nabla^T f(x^l) (x - x^l) + \mu u \\
 & B y + g(x^l) + \nabla^T g(x^l) (x - x^l) \leq u \\
 & \alpha \in \mathfrak{R}, \quad x \in X \subseteq \mathfrak{R}^n, \quad y \in Y = \{0,1\}^q, \quad u \geq 0
 \end{aligned}$$

A partir do relaxamento das variáveis binárias no problema acima, resolve-se j -LPs até encontrar uma solução inteira. Quando esta solução é encontrada é resolvido um subproblema k -NLP e a partir desta solução há um acréscimo de uma inequação para todos os nós abertos na árvore de busca, sendo que se não houver violação de restrição, a aproximação feita será do tipo OA e se houver violação de restrição, a aproximação será feita através de cortes de Benders. Assim, o MILP terá a seguinte forma:

$$\begin{aligned}
 z &= \min_{x,y,\alpha,u} \alpha \\
 \text{sujeito a } \quad & \alpha \geq c^T y + f(x^l) + \nabla^T f(x^l) (x - x^l) + \mu u
 \end{aligned}$$

$$\begin{aligned}
& B y + g(x^l) + \nabla^T g(x^l) (x - x^l) \leq u \\
& \alpha \geq c^T y + f(x^k) + \nabla^T f(x^k) (x - x^k) + \mu u, \quad k = 1, 2, \dots, K_{(NLPs \text{ viáveis})} \\
& \alpha \geq c^T y + f(x^k) + (\lambda^k)^T [B y + g(x^k) - u] + \mu u, \quad k = 1, 2, \dots, K_{(NLPs \text{ inviáveis})} \\
& \alpha \in \mathfrak{R}, \quad x \in X \subseteq \mathfrak{R}^n, \quad y \in Y = \{0, 1\}^q, \quad u \geq 0
\end{aligned}$$

onde $K_{(NLPs \text{ viáveis})}$ são todos pontos viáveis obtidos pelo NLP e $K_{(NLPs \text{ inviáveis})}$ são todos os pontos inviáveis resultantes do NLP.

Branch and Bound Espacial Não Linear

A diferença entre o SBB linearizado e o SBB não linear está na solução do problema relaxado. Enquanto que o SBB linearizado resolve problemas LP relaxados para encontrar limites inferiores, o SBB não linear resolve problemas NLP relaxados.

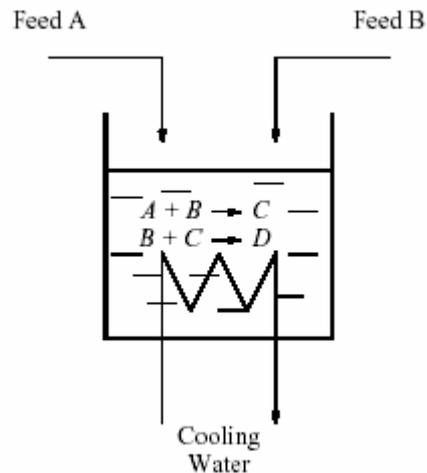
Os algoritmos SBB linearizados partem da suposição de que o LP relaxado pode ser resolvido mais eficientemente que o NLP relaxado. Contudo, se muitos NLPs resultantes de soluções inteiras do LP relaxado forem inviáveis, então os algoritmos SBB linearizados podem encontrar dificuldades de convergência. A introdução de novas restrições, resultantes de sucessivas linearizações, também podem trazer mau condicionamento para os problemas LP relaxados, além de levar a possíveis cortes de soluções ótimas em problemas não convexos.

Por outro lado, os algoritmos SBB não lineares partem da premissa de que os subproblemas NLPs diferem por apenas algumas restrições e, portanto, a solução de um NLP pode servir como uma boa estimativa inicial para outro NLP, acelerando a convergência.

É de se esperar que o SBB não linear seja mais eficiente em problemas com poucas variáveis discretas e com maior não linearidade e não convexidade.

8. Otimização de Processos Dinâmicos

Exemplo 8.1: Para ilustrar o problema de otimização dinâmica, algumas vezes chamado de otimização em *dimensão infinita*, pois envolve variáveis de decisão que são funções, considere o sistema abaixo de um reator semi-batelada, com duas reações exotérmicas:



onde C é o produto de interesse e D é um sub-produto. Iniciando com o reator vazio, as alimentações de A e B, bem com a taxa de água de resfriamento, estão livres para variar ao longo da operação. Para um dado reator, com dimensões fixas, o objetivo operacional pode ser o de determinar o tempo de duração da operação e as taxas de alimentação de reagentes e água, de modo a maximizar a concentração final de C. O projeto do equipamento também pode ser parte do objetivo do problema. Geralmente, quando as variáveis de decisão envolvem somente variáveis operacionais, o problema é denominado de *controle ótimo*.

A formulação matemática do processo acima pode ser genericamente representada por:

$$F(x(t), \dot{x}(t), y(t), u(t), v) = 0$$

onde $x(t)$, $\dot{x}(t)$ são as variáveis diferenciais e suas derivadas em relação a variável independente, t (geralmente o tempo), $y(t)$ são as variáveis algébricas, $u(t)$ são as variáveis de controle e v são os parâmetros invariantes no tempo, a serem determinados pela otimização. Para o exemplo, $u(t)$ representa as taxas de alimentação dos reagentes e da água de refrigeração e v pode ser o volume do reator. As condições iniciais para este problema podem ser descritas genericamente por:

$$I(x(0), \dot{x}(0), y(0), u(0), v) = 0$$

que juntamente com a forma funcional de $u(t)$ e os valores de v determinam completamente a resposta transiente do sistema.

8.1 Formulação da função objetivo e restrições

A formulação da função objetivo para um problema de otimização dinâmica pode estar associada a determinação dos seguintes valores:

- horizonte de tempo da operação, t_f (tempo final);
- parâmetros invariantes no tempo, v ;
- variação temporal das variáveis de controle, $u(t)$, para $t \in [0, t_f]$.

de modo a minimizar (ou maximizar) um funcional $\Phi(u(t), v, t_f)$. A função objetivo é um *funcional* pois depende da escolha da função $u(t)$. No problema do reator semi-batelada Φ seria a concentração final do produto C e t_f seria a duração da batelada, resultando no problema de otimização abaixo:

$$\begin{aligned} \max_{t_f, v, u(t), t \in [0, t_f]} \quad & \Phi = x_3(t_f) \\ \text{sujeito a:} \quad & F(x(t), \dot{x}(t), y(t), u(t), v) = 0 \\ & I(x(0), \dot{x}(0), y(0), u(0), v) = 0 \end{aligned}$$

onde x_3 é a concentração de C.

Se a função objetivo a ser minimizada for a integral de uma função sobre todo o intervalo $[0, t_f]$:

$$\Phi(u(t), v, t_f) = \int_0^{t_f} \varphi(x(t), \dot{x}(t), y(t), u(t), v) dt$$

pode-se adicionar ao sistema de equações algébrico-diferenciais do problema, a seguinte equação:

$$\dot{\Phi} = \varphi(x(t), \dot{x}(t), y(t), u(t), v)$$

com a condição inicial $\Phi(0) = 0$, para ser integrada simultaneamente. Para o caso particular de $\dot{\Phi} = 1$, tem-se o problema de minimização do tempo de operação, t_f .

Uma formulação mais geral para a função objetivo envolve as duas formas acima, ou seja:

$$\Phi(u(t), v, t_f) = \psi(x(t_f), \dot{x}(t_f), y(t_f), u(t_f), v) + \int_0^{t_f} \varphi(x(t), \dot{x}(t), y(t), u(t), v) dt$$

Dentre as mais variadas formas de restrições adicionais em um problema de otimização dinâmica, tem-se:

a) Limites nas variáveis de decisão

$$t_f^{\min} \leq t_f \leq t_f^{\max}$$

$$u^{\min} \leq u(t) \leq u^{\max}, \quad \forall t \in [0, t_f]$$

$$v^{\min} \leq v \leq v^{\max}$$

b) Restrições terminais

de igualdade: $w(t_f) = w^*$

de desigualdade: $w^{\min} \leq w(t_f) \leq w^{\max}$

onde w representa alguma variável do sistema (x ou y), ou uma relação entre elas. Para o exemplo do reator, a quantidade final de material no reator pode ser fixada ou a temperatura final pode estar limitada em uma dada faixa.

c) Restrições interiores

Aparecem quando algumas variáveis devem estar limitadas em alguns pontos no interior do intervalo de operação, ou seja:

$$w^{\min}(t_I) \leq w(t_I) \leq w^{\max}(t_I)$$

onde $t_I \in [0, t_f]$. Como por exemplo, limitar a curva de aquecimento do reator dentro de uma faixa determinada.

d) Restrições de trajetória

São aquelas restrições que devem ser satisfeitas durante todo o intervalo de operação, ou seja:

$$w^{\min} \leq w(t) \leq w^{\max}, \quad \forall t \in [0, t_f]$$

Por exemplo, no reator semi-batelada pode-se impor que a temperatura da reação não deva ultrapassar um determinado valor para evitar a degradação do produto.

8.2 Princípio do Mínimo de Pontryagin

Antes de enunciar o princípio do mínimo de Pontryagin, é necessário introduzir os conceitos do cálculo variacional, que trata da seleção de uma função desconhecida que aparece no integrando de uma integral que deve ser minimizada ou maximizada em função desta seleção.

Considere a seguinte integral:

$$S(x) = \int_{t_1}^{t_2} \varphi[t, x(t), \dot{x}(t)] dt$$

onde $\dot{x}(t)$ é a derivada de $x(t)$ em relação a t . O problema do cálculo variacional consiste na seleção de $x(t)$ de modo a minimizar (ou maximizar) o funcional $S(x)$. Supondo que $x^*(t)$ é a função que minimiza $S(x)$ e $x(t)$ é uma outra função com uma diferença infinitesimal de $x^*(t)$ em cada ponto dentro do intervalo (t_1, t_2) , define-se:

$$\delta x = x(t) - x^*(t)$$

como o operador variação, ou seja, a variação de uma função representa uma mudança infinitesimal arbitrária na função a um dado valor de t . Note que a variação difere da diferenciação, pois esta última corresponde a uma medida da mudança de uma função resultante de uma mudança específica (não arbitrária) na variável independente. A equação acima pode ser escrita como:

$$\delta x = x(t) - x^*(t) = \varepsilon \phi(t)$$

onde $\phi(t)$ é uma função arbitrária contínua e diferenciável e ε é um parâmetro variável que tende a zero.

Como o operador variação acarreta uma mudança infinitesimal em uma função para um valor fixo de t , tem-se:

$$\delta t = 0$$

ou seja, a variável independente não participa do processo de variação. Outra propriedade importante do operador variação é a comutatividade com os operadores de diferenciação e integração:

$$\frac{d}{dt} \delta x = \frac{d}{dt} [\varepsilon \phi(t)] = \varepsilon \frac{d\phi}{dt} \quad \text{e} \quad \delta \frac{dx}{dt} = \frac{dx}{dt} - \frac{dx^*}{dt} = \frac{d}{dt} (x - x^*) = \varepsilon \frac{d\phi}{dt}$$

$$\delta \int_{t_1}^{t_2} x(t) dt = \int_{t_1}^{t_2} x(t) dt - \int_{t_1}^{t_2} x^*(t) dt = \int_{t_1}^{t_2} [x(t) - x^*(t)] dt = \int_{t_1}^{t_2} \delta x(t) dt$$

Como a condição necessária de primeira ordem para o mínimo local de uma função, $S(x)$, pode ser interpretada como a existência de um ponto estacionário da função objetivo, então dentro de uma região infinitesimal em torno deste ponto tem-se que:

$$\delta S = 0$$

Usando a propriedade comutativa chega-se a:

$$\delta S = \int_{t_1}^{t_2} \delta \varphi dt$$

onde $\delta \varphi = \varphi[x, \dot{x}(t), t] - \varphi[x^*, \dot{x}^*(t), t] = \varphi[x^* + \varepsilon \phi, \dot{x}^*(t) + \varepsilon \dot{\phi}(t), t] - \varphi[x^*, \dot{x}^*(t), t]$. Expandindo φ em série de Taylor tem-se:

$$\begin{aligned} \varphi[x^* + \varepsilon \phi(t), \dot{x}^*(t) + \varepsilon \dot{\phi}(t), t] &= \varphi[x^*, \dot{x}^*(t), t] + \nabla_x^T \varphi[x^*, \dot{x}^*(t), t] \varepsilon \phi(t) + \\ &\quad \nabla_{\dot{x}}^T \varphi[x^*, \dot{x}^*(t), t] \varepsilon \dot{\phi}(t) + O[\varepsilon^2 \phi^2(t), \varepsilon^2 \dot{\phi}^2(t)] \end{aligned}$$

que eliminando os termos de ordem $O(\varepsilon^2)$, resulta em:

$$\delta\varphi = \varepsilon \{ \nabla_x^T \varphi[x^*, \dot{x}^*(t), t] \phi(t) + \nabla_{\dot{x}}^T \varphi[x^*, \dot{x}^*(t), t] \dot{\phi}(t) \}$$

Para um ponto estacionário tem-se:

$$\delta S = \varepsilon \int_{t_1}^{t_2} \left(\nabla_x^T \varphi \phi + \nabla_{\dot{x}}^T \varphi \dot{\phi} \right) dt = 0$$

sobre qualquer função arbitrária $\phi(t)$. O segundo termo no integrando pode ser integrado por partes ($u = \nabla_x^T \varphi$ e $v = \phi$), resultando em:

$$\int_{t_1}^{t_2} \nabla_{\dot{x}}^T \varphi \dot{\phi} dt = [\nabla_{\dot{x}}^T \varphi \phi]_{t_1}^{t_2} - \int_{t_1}^{t_2} \frac{d}{dt} [\nabla_{\dot{x}}^T \varphi] \phi dt$$

Se a função $x(t)$ é fixada nos contornos, t_1 e t_2 , então $\phi(t)$ deve se anular nestes pontos, pois não pode existir variação de $x(t)$ nos contornos. Deste modo o primeiro termo da integração por partes é nulo. Para o caso mais geral, onde algum contorno pode estar livre, as seguintes condições de *complementaridade* devem ser satisfeitas:

$$\nabla_{\dot{x}}^T \varphi \phi = 0 \quad \text{para } t = t_1 \text{ e } t = t_2$$

ou seja, se δS é para ser zero para todas as variações admissíveis (restritas somente pela continuidade e diferenciabilidade) então o primeiro termo da integração por partes deve ser sempre nulo. No caso de existir uma contribuição terminal na função objetivo:

$$S(x) = \Psi[x(t_2), t_2] + \int_{t_1}^{t_2} \varphi[t, x(t), \dot{x}(t)] dt$$

a condição de complementaridade terminal seria:

$$[\nabla_x^T \Psi + \nabla_{\dot{x}}^T \varphi] \phi = 0 \quad \text{para } t = t_2$$

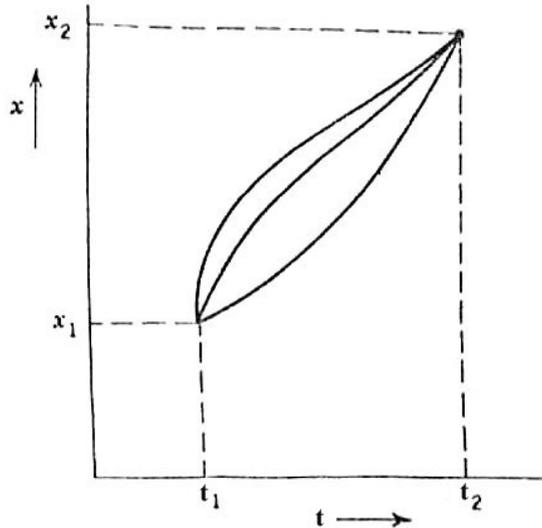
Substituindo a integração por partes em $\delta S = 0$, tem-se então:

$$\delta S = \varepsilon \int_{t_1}^{t_2} \left(\nabla_x^T \varphi - \frac{d}{dt} [\nabla_{\dot{x}}^T \varphi] \right) \phi dt = 0$$

Como a equação acima deve ser satisfeita para qualquer função arbitrária $\phi(t)$, contínua, diferenciável e que satisfaça as condições de contorno, então o termo entre parêntesis deve ser nulo, resultando na equação de *Euler-Lagrange*:

$$\nabla_x \varphi - \frac{d}{dt} [\nabla_{\dot{x}} \varphi] = 0$$

Exemplo 8.2: Para ilustrar, considere o problema de determinar a curva de menor comprimento conectando dois pontos no espaço, conforme a figura abaixo.



O comprimento de uma curva conectando os pontos (t_1, x_1) e (t_2, x_2) é dado por:

$$S(x) = \int_{s(t_1)}^{s(t_2)} ds = \int_{t_1}^{t_2} \sqrt{1 + \dot{x}^2} dt$$

onde $\varphi[t, x(t), \dot{x}(t)] = \sqrt{1 + \dot{x}^2}$ e $ds = \sqrt{dt^2 + dx^2}$ que é o comprimento de um segmento infinitesimal da curva.

Aplicando a equação de Euler-Lagrange para o problema acima, tem-se:

$$\nabla_x \varphi = \frac{\partial \varphi}{\partial x} = 0$$

$$\nabla_{\dot{x}} \varphi = \frac{\partial \varphi}{\partial \dot{x}} = \frac{\dot{x}}{\sqrt{1 + \dot{x}^2}}$$

$$\nabla_x \varphi - \frac{d}{dt} [\nabla_{\dot{x}} \varphi] = 0 - \frac{d}{dt} \left(\frac{\partial \varphi}{\partial \dot{x}} \right) = - \frac{d}{dt} \left(\frac{\dot{x}}{\sqrt{1 + \dot{x}^2}} \right) = 0$$

ou seja: $\frac{\dot{x}}{\sqrt{1 + \dot{x}^2}} = \text{constante}$, o que implica que $\dot{x}^*(t) = \text{constante}$. Portanto,

como não poderia deixar de ser, o caminho mais curto entre dois pontos é uma reta: $x^*(t) = a t + b$.

Naturalmente, para assegurar que a função $x(t)$ minimize o funcional $S(x)$, a condição suficiente de segunda ordem deve ser verificada, que no caso de variações significa que a segunda variação de S deve ser positiva para todas as variações possíveis de x , isto é:

$$\delta^2 S > 0, \forall \delta x$$

que é equivalente a $(\delta x)^T \nabla_x^2 S(x^*) \delta x > 0$, ou ainda pela condição de segunda ordem de Legendre:

$$\nabla_{\dot{x}}^2 \varphi(x^*, \dot{x}^*, t) > 0$$

Aplicando esta condição para o exemplo acima tem-se:

$$\nabla_{\dot{x}}^2 \varphi = \frac{1}{(1 + \dot{x}^2)^{3/2}} > 0$$

logo, a solução ótima $x^*(t)$ minimiza o funcional $S(x)$.

Retornando ao problema de otimização de sistema dinâmicos, seja então o problema:

$$\begin{aligned} \min_{t_f, x(t), u(t), t \in [0, t_f]} \quad & \Phi[x(t), u(t), t_f] = \psi[x(t_f), t_f] + \int_0^{t_f} \varphi[x(t), u(t), t] dt \\ \text{sujeito a:} \quad & \dot{x}(t) = f[x(t), u(t), t], \quad x(0) = x_0 \\ & h[(t_f), t_f] = 0 \end{aligned}$$

Introduzindo os multiplicadores de Lagrange associado às restrições chega-se a:

$$\begin{aligned} L[x(t), u(t), \lambda(t), \eta, t_f] = & \psi[x(t_f), t_f] + \eta^T h[x(t_f), t_f] + \\ & \int_0^{t_f} \{ \varphi[x(t), u(t), t] + \lambda^T(t) (f[x(t), u(t), t] - \dot{x}(t)) \} dt \end{aligned}$$

que pode ser dividida em duas funções:

$$\Theta[x(t_f), \eta, t_f] = \psi[x(t_f), t_f] + \eta^T h[x(t_f), t_f] \text{ e}$$

$$\Lambda[x(t), u(t), \lambda(t), t] = \varphi[x(t), u(t), t] + \lambda^T(t) (f[x(t), u(t), t] - \dot{x}(t))$$

resultando no funcional:

$$L[x(t), u(t), \lambda(t), \eta, t_f] = \Theta[x(t_f), \eta, t_f] + \int_0^{t_f} \Lambda[x(t), u(t), \lambda(t), t] dt$$

que ao ser minimizado resulta nas equações de Euler-Lagrange:

$$\nabla_x \Lambda - \frac{d}{dt} [\nabla_{\dot{x}} \Lambda] = 0 = \nabla_x \varphi + \nabla_x^T f \lambda + \dot{\lambda}$$

$$\nabla_u \Lambda - \frac{d}{dt} [\nabla_{\dot{u}} \Lambda] = 0 \Rightarrow \nabla_{\dot{u}} \Lambda = cte$$

onde $\dot{\omega}(t) = u(t)$, e pelas condições de complementaridade para $\dot{\omega}(t_f)$:

$$[\nabla_{\omega} \Theta + \nabla_{\omega} \Lambda]^T \delta \omega_f = 0 \Rightarrow \nabla_{\omega} \Theta + \nabla_{\omega} \Lambda = 0 \Rightarrow \nabla_{\omega} \Lambda = 0 \text{ para } t = t_f$$

pois $\omega(t_f)$ é livre ($\delta\omega_f \neq 0$) e Θ não depende de ω , chega-se a $\nabla_{\dot{\omega}}\Lambda = 0 \quad \forall t$, ou seja:

$$\nabla_{\dot{\omega}}\Lambda = 0 = \nabla_{\dot{\omega}}\varphi + \nabla_{\dot{\omega}}^T f \lambda = \nabla_u \varphi + \nabla_u^T f \lambda$$

Definindo a seguinte *função de Hamilton* (ou *Hamiltoniano*):

$$H[x(t), u(t), \lambda(t), t] = \varphi[x(t), u(t), t] + \lambda^T(t) f[x(t), u(t), t]$$

então, as condições necessárias de primeira ordem (Euler-Lagrange) descritas acima são equivalentes a:

$$\dot{x} = \nabla_{\lambda} H = f$$

$$\dot{\lambda} = -\nabla_x H = -\nabla_x \varphi - \nabla_x^T f \lambda$$

$$\nabla_u H = \nabla_u \varphi + \nabla_u^T f \lambda = 0$$

que devem ser resolvidas juntamente com as condições iniciais e finais e as condições de complementaridade:

$$x(0) = x_0$$

$$h[x(t_f), t_f] = 0$$

$$[\nabla_{x(t_f)} \Theta - \lambda(t_f)]^T \delta x(t_f) = 0$$

$$\{H[x(t_f), u(t_f), \lambda(t_f), t_f] + \nabla_{t_f} \Theta\} \delta t_f = 0$$

A condição de segunda ordem pode ser verificada pela positividade de $\nabla_u^2 H$. Este conjunto de condições são conhecidos como *Princípio do Mínimo de Pontryagin* (1962), e os multiplicadores de Lagrange, $\lambda(t)$, são conhecidos como *variáveis adjuntas*. Para as partes do caminho $u(t)$ sobre as restrições u^{\min} ou u^{\max} , vale a seguinte condição de otimalidade, no lugar de $\nabla_u H = 0$:

$$H[x^*(t), u^*(t), \lambda^*(t), t] \leq H[x^*(t), u(t), \lambda^*(t), t]$$

para todos as possíveis funções $u(t)$.

Para problemas com restrições de desigualdade do tipo:

$$g[x(t), u(t), t] \leq 0, \quad \forall t \in [0, t_f]$$

onde $g \in \mathfrak{R}^p$ é duas vezes continuamente diferenciável e o subconjunto de restrições ativas, g^a , não deve ser maior que o número de variáveis de controle, u , e deve apresentar uma matriz Jacobiana, $\nabla_u g^a$, de posto completo $\forall t \in [0, t_f]$. Exemplos de restrições de desigualdade que satisfazem estas condições são:

$$u^{\min} \leq u(t) \leq u^{\max}, \quad \forall t \in [0, t_f]$$

$$u^{\min}[x(t), t] \leq u(t) \leq u^{\max}[x(t), t], \quad \forall t \in [0, t_f]$$

$$|u(t)| \leq u^m, \quad \forall t \in [0, t_f]$$

$$\|u(t)\|^2 \leq r^2, \quad \forall t \in [0, t_f], \quad r \in \mathfrak{R}$$

Introduzindo funções de folga $\dot{z}(t)$, isto é:

$$g_i[x(t), u(t), t] + \dot{z}_i^2(t) = 0, \quad i = 1, 2, \dots, p$$

tem-se o integrando da função de Lagrange na forma:

$$\Lambda[x(t), u(t), \lambda(t), t] = \varphi[x(t), u(t), t] + \lambda^T(t) (f[x(t), u(t), t] - \dot{x}(t)) + \mu^T(t) (g[x(t), u(t), t] + \dot{z}^2(t))$$

onde o vetor $\dot{z}^2(t) = [\dots \dot{z}_i^2(t) \dots]^T$. Neste caso as equações de Euler-Lagrange são dadas por:

$$\nabla_x \Lambda - \frac{d}{dt} [\nabla_{\dot{x}} \Lambda] = 0 = \nabla_x \varphi + \nabla_x^T f \lambda + \nabla_x^T g \mu + \dot{\lambda}$$

$$\nabla_{\omega} \Lambda - \frac{d}{dt} [\nabla_{\dot{\omega}} \Lambda] = 0 \Rightarrow \nabla_{\dot{\omega}} \Lambda = cte$$

$$\nabla_z \Lambda - \frac{d}{dt} [\nabla_{\dot{z}} \Lambda] = 0 \Rightarrow \nabla_{\dot{z}} \Lambda = cte$$

De modo análogo às condições de complementaridade para $\omega(t)$, que resultam em $\nabla_{\dot{\omega}} \Lambda = 0 \quad \forall t$, para a função $z(t)$ tem-se:

$$[\nabla_z \Theta + \nabla_z \Lambda]^T \delta z_f = 0 \Rightarrow \nabla_z \Theta + \nabla_z \Lambda = 0 \Rightarrow \nabla_z \Lambda = 0 \quad \text{para } t = t_f$$

logo, $\nabla_z \Lambda = 0 \quad \forall t$ e, pela definição de Λ , $\mu_i \dot{z}_i = 0$, ou ainda $\mu_i g_i = 0, \quad i = 1, 2, \dots, p$.

Definindo o Hamiltoniano para este caso:

$$H[x(t), u(t), \lambda(t), t] = \varphi[x(t), u(t), t] + \lambda^T(t) f[x(t), u(t), t] + \mu^T(t) g[x(t), u(t), t]$$

então o Princípio do Mínimo de Pontryagin (ou condições necessárias de otimalidade para sistemas dinâmicos) é dado por:

$$\dot{x} = \nabla_{\lambda} H = f$$

$$\dot{\lambda} = -\nabla_x H = -\nabla_x \varphi - \nabla_x^T f \lambda - \nabla_x^T g \mu$$

$$\nabla_u H = \nabla_u \varphi + \nabla_u^T f \lambda + \nabla_u^T g \mu = 0$$

$$\mu \geq 0$$

$$\nabla_u^2 H \geq 0 \quad \forall \{\delta u \mid \nabla_u^T g^a \delta u = 0\}$$

juntamente com as condições iniciais, finais e de complementaridade:

$$x(0) = x_0$$

$$h[x(t_f), t_f] = 0$$

$$[\nabla_{x(t_f)} \Theta - \lambda(t_f)]^T \delta x(t_f) = 0$$

$$\{H[x(t_f), u(t_f), \lambda(t_f), t_f] + \nabla_{t_f} \Theta\} \delta t_f = 0$$

$$\mu_i g_i = 0, i = 1, 2, \dots, p.$$

8.3 Controle ótimo

Analisando inicialmente o problema de controle ótimo para sistemas lineares:

$$\dot{x}(t) = A(t) x(t) + B(t) u(t) \quad , \quad x(0) = x_0$$

com a seguinte função objetivo quadrática:

$$S[x(t), u(t), t_f] = \frac{1}{2} \|x(t_f)\|_R^2 + \frac{1}{2} \int_0^{t_f} [\|x(t)\|_{Q(t)}^2 + \|u(t)\|_{W(t)}^2] dt$$

onde $\|x\|_R^2 = x^T R x$ e $R \geq 0$, $Q(t) \geq 0$ e $W(t) \geq 0$ são matrizes simétricas de pesos.

Definindo o Hamiltoniano para este problema:

$$H[x(t), u(t), \lambda(t), t] = \frac{1}{2} \|x(t)\|_{Q(t)}^2 + \frac{1}{2} \|u(t)\|_{W(t)}^2 + \lambda^T [A x + B u]$$

então, pelo princípio do mínimo de Pontryagin, tem-se:

$$\begin{aligned} \dot{x} &= \nabla_{\lambda} H = A x + B u \\ \dot{\lambda} &= -\nabla_x H = -Q x - A^T \lambda \\ \nabla_u H &= W u + B^T \lambda = 0 \\ x(0) &= x_0 \\ \lambda(t_f) &= R x(t_f) \end{aligned}$$

que resolvendo para $u(t)$, chega-se a:

$$\begin{aligned} u &= -W^{-1} B^T \lambda \\ \dot{x} &= A x - B W^{-1} B^T \lambda \end{aligned}$$

gerando um problema de TPBVP (*two-point boundary value problem*) para x e λ , cuja solução minimiza a função objetivo, pois $\nabla_u^2 H = W > 0$, e é um mínimo global devido a convexidade do problema. Se a solução $\lambda(t)$ do problema de TPBVP puder ser expressa como uma função da solução $x(t)$ em termos de uma transformação linear $P(t) \in \mathfrak{R}^{n \times n}$, isto é, $x(t)$ ser uma base de soluções para $\lambda(t)$:

$$\lambda(t) = P(t) x(t)$$

então a solução do TPBVP pode ser facilmente obtida. Para verificar tal possibilidade, primeiro substitui-se esta dependência nas equações para \dot{x} e $\dot{\lambda}$, e subtrai-se uma da outra resultando em:

$$(\dot{P} + PA + A^T P - P B W^{-1} B^T P + Q) x(t) = 0$$

que deve ser satisfeita $\forall t$, logo como $x(t) \neq 0$ então:

$$\dot{P} = -PA - A^T P + P B W^{-1} B^T P - Q$$

que é uma equação diferencial matricial não linear, conhecida como equação de *Riccati* dinâmica, cuja condição de contorno é obtida de $\lambda(t_f) = R x(t_f)$, ou seja:

$$P(t_f) = R$$

Fazendo a mudança de variável $\tau = t_f - t$, tem-se $d\tau = -dt$ e:

$$\dot{P} = PA + A^T P - P B W^{-1} B^T P + Q$$

$$P(0) = R$$

Portanto, o problema de TPBVP foi transformado em um problema de valor inicial de dimensão $n(n+1)/2$, pois $P(t)$ é uma matriz simétrica. Com a solução da equação de *Riccati* dinâmica, a lei de controle ótimo quadrático é dada por:

$$u^* = -K(t) x^*(t)$$

$$K(t) = W(t)^{-1} B(t)^T P(t)$$

que é independente da condição inicial x_0 . O sistema dinâmico em malha fechada pode ser escrito como:

$$\dot{x}^* = (A - B W^{-1} B^T P) x^* = A_K x^*, \quad x(0) = x_0$$

onde $A_K(t) = A - B W^{-1} B^T P$ é a matriz característica da dinâmica do sistema regulado. Observa-se também que o valor ótimo da função objetivo é dado por:

$$S[x^*, u^*, t_f] = \frac{1}{2} x_0^T P(0) x_0$$

onde $P(0)$ é obtido em $t = 0$ ($\tau = t_f$). Este resultado pode ser verificado pela equação:

$$\frac{d}{dt}(x^T P x) = \dot{x}^T P x + x^T \dot{P} x + x^T P \dot{x}$$

que ao substituir as expressões para \dot{P} e \dot{x}^* , resulta em:

$$\frac{d}{dt}(x^T P x) = -x^T Q x - x^T P B W^{-1} B^T P x = -x^T Q x - u^T W u$$

Integrando a equação acima em $[0, t_f]$ tem-se:

$$x(0)^T P(0) x(0) = x(t_f)^T P(t_f) x(t_f) + \int_0^{t_f} (x^T Q x + u^T W u) dt = S[x(t), u(t), t_f]$$

pois $P(t_f) = R$.

Fazendo a integração em $[t, t_f]$ tem-se:

$$x(t)^T P(t) x(t) = x(t_f)^T R x(t_f) + \int_t^{t_f} (x^T Q x + u^T W u) dt \geq 0$$

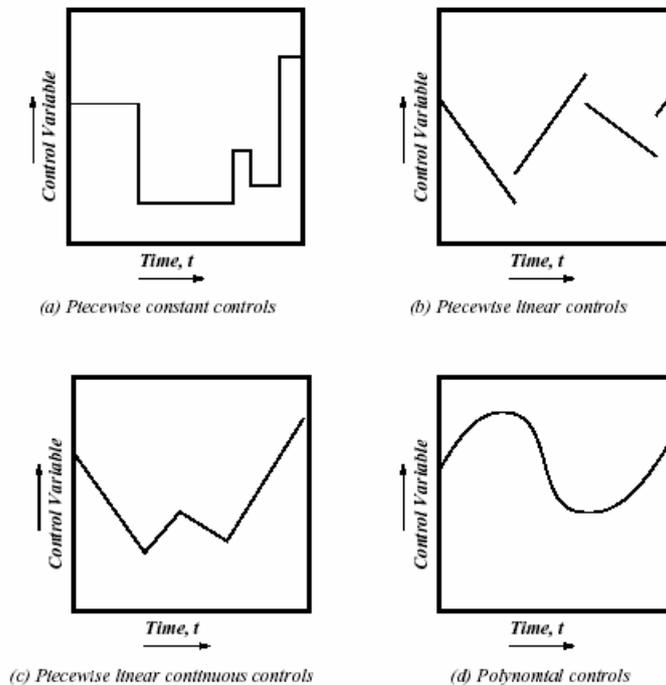
de onde conclui-se que $P(t)$ é positiva semidefinida.

Para o caso particular de um sistema invariante no tempo, com $[A \ B]$ controlável, $t_f = \infty$ e matrizes pesos constantes, a lei de controle ótimo pode ser obtida através da solução da equação de *Ricatti estacionária*:

$$PA + A^T P - P B W^{-1} B^T P + Q = 0$$

que pode ser eficientemente resolvida via decomposição em valores característicos do gradiente do Hamiltoniano associado. Neste caso a matriz de ganhos, K , da lei de controle é invariante no tempo.

Na maioria dos casos de controle ótimo não linear, é mais eficiente tratar o problema como uma programação não linear do que utilizar o princípio do mínimo de Pontryagin, para obter o controle ótimo, $u(t)$. Nestes casos, a escolha da forma das variáveis de controle toma um aspecto mais de engenharia do que matemático, pois dependerá da forma com que as ações de controle serão implementadas na prática. Alguns tipos de variações mais comuns para as variáveis de controle estão ilustrados abaixo.



Sendo as duas primeiras mais frequentemente utilizadas em controle ótimo de processos.

Para qualquer um dos casos acima, o algoritmo abaixo mostra uma maneira de solucionar o problema de otimização dinâmica via NLP, conhecida como método da tentativa-e-erro ou *single-shooting*.

algoritmo

- 1) Escolher um perfil inicial para as variáveis de controle, $u^0(t)$, $k = 0$.
- 2) Integrar o sistema de equações algébrico diferenciais.
- 3) Calcular a função objetivo e restrições.
- 4) Corrigir as variáveis de controle pelo uso de métodos de NLP.
- 5) Verificar a convergência de $u^k(t)$. Caso não esteja satisfeita $k \leftarrow k + 1$ (ir para 2).
- 6) FIM.

Outra forma de resolver o problema é através da discretização das equações diferenciais (ou método de *multiple-shooting*) por técnicas de elementos finitos, diferenças finitas ou aproximação polinomial, gerando um problema algébrico de programação não linear de elevada dimensão, que também pode ser resolvido por qualquer método de NLP com restrição.

Exercícios

1) A figura abaixo representa um extrator onde entram uma solução com uma vazão $F = 10^4$ kg H₂O / h e uma concentração $x_0 = 0,02$ kg A / kg H₂O, e um solvente B puro a uma vazão W kg B / h a ser determinada. A corrente refinada sai com uma vazão $F = 10^4$ kg H₂O / h e uma concentração x kg A / kg H₂O e ainda um pouco de solvente B dissolvido em água. No extrato, a vazão é de W kg B / h e a concentração y kg A / kg B. Calcular a vazão ótima W de B.

dados: preço do soluto no extrato: 0,4 \$ / kg A

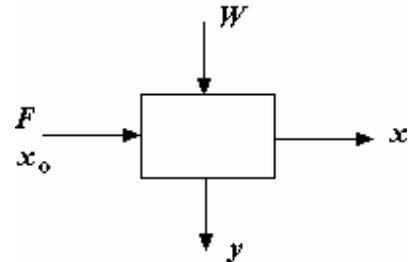
preço do solvente B: 0,01 \$ / kg B

solubilidade de B em água: 0,0007 kg B / kg H₂O

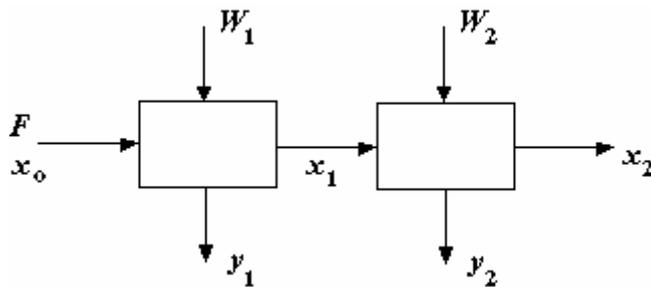
relação de equilíbrio: $y = k x$ ($k = 4$)

o despejo não tem valor comercial.

critério de desempenho: Lucro = \$ soluto extraído - \$ solvente utilizado.



2) Calcular as vazões ótimas de solvente para o exercício anterior quando dois extratores em série são utilizados no lugar de apenas um extrator, conforme figura abaixo. Considere a mesma relação de equilíbrio e a mesma solubilidade de B em água para os dois extratores.



3) A figura abaixo representa um trocador de calor em contra-corrente, onde uma corrente de processo com vazão $W_1 = 1000$ kg / h e temperatura $T_1 = 200$ °C é resfriada a $T_2 = 100$ °C por uma corrente fria a $t_2 = 60$ °C. Calcular a vazão ótima W_2 do refrigerante (kg / h) bem como a área de troca térmica ótima A (m²). Considerar $c_p = 1$ kcal / kg °C para as duas correntes e $U = 500$ kcal / h m² °C.

dados: investimento do trocador de calor: $3200 (A / 50)^{0,48} \$$

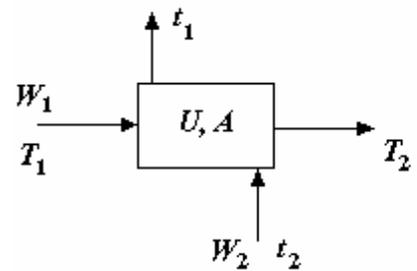
custo do refrigerante: $5 \times 10^{-6} \$ / \text{kg}$

custo de manutenção: 2 % do investimento / ano

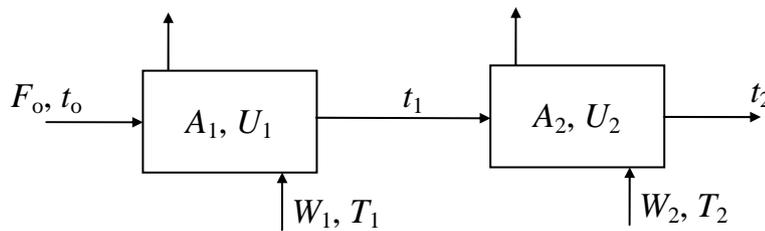
tempo de operação: 8640 h / ano

critério de desempenho: custo anual = 50 % custo

operacional + 10 % ao ano sobre o investimento



4) A figura abaixo representa um sistema de dois trocadores de calor em série, onde uma corrente de processo com vazão $F_o = 1000 \text{ kg / h}$ e temperatura $t_o = 20 \text{ }^\circ\text{C}$ deve ser aquecida a $t_2 = 100 \text{ }^\circ\text{C}$ por correntes de vapor saturado a $T_1 = 100 \text{ }^\circ\text{C}$ e $T_2 = 120 \text{ }^\circ\text{C}$, respectivamente. Calcular as vazões ótimas W_1 e W_2 das correntes de vapor (kg / h) bem como as áreas de troca térmica ótimas A_1 e A_2 (m^2). Considerar $cp = 1 \text{ kcal / kg }^\circ\text{C}$, $U_1 = 300 \text{ kcal / h m}^2 \text{ }^\circ\text{C}$, $U_2 = 400 \text{ kcal / h m}^2 \text{ }^\circ\text{C}$, $\lambda_1 = 500 \text{ kcal / kg}$ e $\lambda_2 = 600 \text{ kcal / kg}$.



dados: investimento dos trocadores de calor: $160 (A_1)^{0,5} \$$ e $320 (A_2)^{0,5} \$$

custo do vapor: (a $100 \text{ }^\circ\text{C}$) $5 \times 10^{-6} \$ / \text{kg}$ e (a $120 \text{ }^\circ\text{C}$) $1 \times 10^{-5} \$ / \text{kg}$

custo de manutenção: 2 % do investimento / ano

tempo de operação: 8640 h / ano

critério de desempenho:

custo anual = 50 % custo operacional + 10 % ao ano sobre o investimento

5) Um reator CSTR converte um reagente A em um produto B através de uma reação catalítica endotérmica. O catalisador pode ser continuamente injetado no reator e a temperatura do reator pode ser controlada pela injeção de vapor através de uma serpentina. Alguma conversão pode ser obtida somente com o aquecimento, sem a adição de catalisador.

Por outro lado, pode-se também obter alguma conversão somente com a adição de catalisador, sem o aquecimento. O produto B pode ser vendido por R\$ 50 / mol de B. O custo do catalisador (R\$ / mol da alimentação) é dado pela expressão:

$$C_c = 2 + 10 X_c + 20 X_c^2$$

onde X_c é a fração molar de catalisador. O custo do vapor (R\$ / mol da alimentação) é dado pela expressão:

$$C_s = 1 + 0,003 S + 2 \times 10^{-6} S^2$$

onde S é a relação de kg de vapor / mol de alimentação. A conversão em termos das quantidades de catalisador e vapor é dada por:

$$X_B = 0,1 + 0,3 X_c + 10^{-4} S - 10^{-4} X_c S$$

onde X_B é a relação de mol de B / mol de alimentação. Deseja-se saber qual o lucro máximo deste processo produtivo por mol de alimentação.

6) Encontre o mínimo global da função

$$f(x) = x_1^2 + x_2^2 - 3x_1 + 15x_2$$

$$\text{sujeito a: } (x_1 + x_2)^2 - 4(x_1 - x_2) = 0$$

Bibliografia Básica

- The Variational Method in Engineering – R. S. Schechter – McGraw-Hill, 1967.
- Optimization: Theory and Practice – G. S. G. Beveridge & R. S. Schechter – McGraw-Hill, 1970.
- Applied Nonlinear Programming – D. M. Himmelblau – McGraw-Hill, 1972.
- Nonlinear Programming – M. S. Bazaraa & C. M. Shetty – John Wiley & Sons, 1979.
- Mathematical Programming. Theory and Algorithms – M. Minoux – John Wiley & Sons, 1986.
- Programação Não-Linear. Introdução à Teoria e aos Métodos – P. Mahey – Editora Campus, 1987.
- Optimization of Chemical Processes – T. F. Edgar & D. M. Himmelblau – McGraw-Hill, 1988.
- Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications – C.A. Floudas – Oxford Press, 1995.
- Optimierung – M. Papageorgiou – Oldenbourg Ed., 1996.
- Systematic Methods of Chemical Process Design – L.T. Biegler, I.E. Grossmann, A.W. Westerberg – Prentice Hall, Inc., 1997.
- Numerical Optimization – J. Nocedal & S. J. Wright – Springer, 1999.
- Numerical Optimization – J. F. Bonnans, J. C. Gilbert, C. Lemaréchal & C. A. Sagastizábal – Springer, 2003.

Referências citadas:

- Secchi, A.R., Perlingeiro, C.A. (1989), “Busca Aleatória Adaptativa”, in *Proc. of XII Congresso Nacional de Matemática Aplicada e Computacional*, São José do Rio Preto, SP, pp. 49-52.
- Aarts, E., Korts, J., Van Laarhoven, P. (1988), “A quantitative analysis of the simulated annealing algorithm: a case study for the travelling salesman problem”, *J. Stat. Phys.*, **50**, 187-206.
- Hasan, M., Alkhamis, T., Ali, J. (2000), “A comparison between simulated annealing, genetic algorithm and tabu search methods for the unconstrained quadratic Pseudo-Boolean function”, *Comp. & Ind. Eng.*, **38**, 323-340.