



Universidade Federal do Rio Grande do Sul
Instituto de Matemática
Programa de Pós-Graduação em Matemática Aplicada

5217
Dissertação (Mestrado)
Matemática
1999

Algoritmo Numérico para a Solução da Programação Mista Não Linear e Inteira

Dissertação de Mestrado

Elaine Corrêa Pereira

Porto Alegre, 1999

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA APLICADA

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
DEPARTAMENTO DE MATEMÁTICA APLICADA
DIVISÃO DE TIRAGEM

ALGORITMO NUMÉRICO
PARA A SOLUÇÃO DA PROGRAMAÇÃO
MISTA NÃO LINEAR E INTEIRA

por

Elaine Corrêa Pereira

Dissertação submetida como requisito parcial
para a obtenção do grau de
Mestre em Matemática Aplicada

Prof. Dr. Argimiro Resende Secchi
Orientador

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
DEPARTAMENTO DE MATEMÁTICA APLICADA
DIVISÃO DE TIRAGEM

Porto Alegre, dezembro de 1999

“O importante não é saber: é nunca
perder a capacidade de aprender”

(anônimo)

Aos meus filhos

e à minha mãe

Elaine

AGRADECIMENTOS

Ao Prof. Dr. Argimiro Resende Secchi pelos conhecimentos transmitidos, orientação e profissionalismo.

Aos professores do Curso de pós-graduação em Matemática Aplicada pelos conhecimentos transmitidos.

Ao Colegiado do Departamento de Matemática da Universidade do Rio Grande pela liberação parcial de minhas atividades.

Aos colegas da turma de mestrado pela amizade e união.

Ao Prof. Tabajara Almeida e Profa. Cleuza Almeida por ter tornado possível a minha estada em Porto Alegre.

Ao Prof. Dinei Gonçalves, Prof. Carlos S. S. Scott Hood e Enga. Débora Cuchiara pela ajuda e apoio no desenvolvimento deste trabalho.

Aos amigos do Departamento de Matemática pelo incentivo e constante apoio.

De modo muito especial ao Prof. Vitor Paulo Leonardo pela amizade e apoio em alguns momentos muito difíceis.

À minha mãe pela dedicação aos meus filhos em todos os momentos durante a realização deste trabalho.

Aos meus filhos Cristian e Vivian pela paciência, compreensão pela ausência e companheirismo.

RESUMO

O presente trabalho apresenta a formulação e implementação de um algoritmo para a solução de problemas de programação mista não linear e inteira (MINLP) convexos. O algoritmo proposto não segue a tradicional solução seqüencial de subproblemas de programação não linear (NLP) e problemas mestres de programação mista linear e inteira (MILP). Em vez disso, o problema mestre é definido dinamicamente durante a busca em árvore para reduzir o número de nós que necessitam ser enumerados. Uma busca “branch” e “bound” é conduzida para determinar limites inferiores das soluções dos subproblemas de programação linear (LP) até encontrar soluções inteiras viáveis. Para estes nós, subproblemas de programação não linear 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. Resultados numéricos em alguns problemas testes são relatados, comparando a eficiência do algoritmo com resultados da literatura estudada. Faz-se também uma análise do comportamento frente a problemas testes não convexos e finalmente a análise de um problema aplicado à rede de trocadores de calor.

ABSTRACT

The current work shows the formulation and implementation of an algorithm for the solution of convex mixed-integer nonlinear programming (MINLP) problems. The proposed algorithm does not follow the traditional sequence solution of nonlinear programming (NLP) subproblems and master problems of mixed-integer linear programming (MILP). Instead, the master problem is defined dynamically during the tree search to reduce the number of nodes that need to be enumerated. A branch and bound search is performed to predict lower bounds by solving linear programming (LP) subproblems until feasible integer solutions are found. For these nodes nonlinear programming subproblems are solved, providing upper bounds and new linear approximations which are used to tighten the linear representation of the open nodes in the search tree. Numerical results on some test problems are reported comparing the efficiency of the algorithm with the results of the studied literature. An analysis of the algorithm behavior for nonconvex test problems is also carried out and finally an analysis of a problem applied to heat exchanger networks is done.

SUMÁRIO

LISTA DE SÍMBOLOS.....	ix
LISTA DE FIGURAS.....	xi
LISTA DE TABELAS.....	xii
1 - INTRODUÇÃO.....	1
1.1 - Formulação do Problema.....	1
1.2 - Teoria de Otimização.....	2
1.2.1 - Mínimos e Máximos.....	4
1.2.2 - Convexidade.....	4
1.2.3 - Condições necessárias e suficientes do ótimo.....	5
1.2.3.1 - Para o caso sem restrição.....	5
1.2.3.2 - Para o caso com restrição.....	6
1.3 - Estrutura da Dissertação.....	13
2 - REVISÃO BIBLIOGRÁFICA.....	15
2.1 - Descrição Matemática.....	15
2.2 - Técnicas de Otimização Global Estocástica.....	17
2.3 - Técnicas de Otimização Global Determinística.....	18
2.3.1 - “Branch” e “Bound” Espacial.....	19
2.3.2 - Decomposição Generalizada de Benders (GBD).....	20

2.3.3 - Aproximações Externas (OA).....	21
2.3.4 - Aproximações Externas com Relaxamento de Igualdade (OA/ER).....	23
2.3.5 - Aproximações Externas com Relaxamento de Igualdade e Penalidade Aumentada (OA/ER/AP).....	24
2.3.6 - Aproximação Externa Generalizada (GOA).....	26
2.3.7 - Decomposição Cruzada Generalizada(GCD).....	29
2.3.8 - Otimização Global (GOP).....	30
2.4 - Aplicações.....	31
2.4.1 - Síntese de Rede de Trocadores de Calor.....	31
2.4.2 - Síntese de Fluxograma de Processos.....	32
2.4.3 - Projeto de Colunas de Destilação.....	33
3 - DESCRIÇÃO DO ALGORITMO.....	35
3.1 - Descrição Detalhada do Algoritmo.....	36
3.2 - Restrições Adicionais.....	41
3.2.1 - OA.....	42
3.2.2 - GBD.....	42
3.2.3 - Restrição proposta por Quesada e Grossmann.....	44
3.2.4 - Restrição proposta neste trabalho.....	46
3.3 - Implementação do Algoritmo.....	47
4 - RESULTADOS.....	48
4.1 - Exemplos convexos.....	48
4.2 - Exemplos não convexos.....	56
4.3 - Aplicações.....	63
5 - CONCLUSÕES E SUGESTÕES.....	73
5.1 - Conclusões.....	73
5.2 - Sugestões de Trabalhos.....	74
5.2.1 - Teste Local.....	76

5.2.2 - Teste Global.....	78
5.2.3 - Problema Mestre na Fase II.....	80
REFERÊNCIAS BIBLIOGRÁFICAS.....	82
ANEXOS.....	88

LISTA DE SÍMBOLOS

\mathbf{x}	vetor de variáveis contínuas
\mathbf{y}	vetor de variáveis binárias
f	função objetivo
\mathbf{h}	restrições de igualdade lineares ou não lineares
\mathbf{g}	restrições de desigualdade lineares ou não lineares
\mathbf{X}	espaço vetorial
\mathbf{x}^*	ponto ótimo
\mathbf{H}	matriz hessiana
\mathbf{L}	função de Lagrange
max	máximo
min	mínimo
\mathbf{w}	função vetorial não dimensional
z_u	limite superior da solução ótima do MINLP
z_l	limite inferior da solução ótima do MINLP
z_{NLP}	solução do NLP
\mathbf{z}	solução ótima ou sub-ótima do MINLP
\mathbf{b}	vetor perturbação
log	logaritmo natural
Σ	somatório
μ	penalização da violação de restrições
λ	vetor dos multiplicadores de Lagrange associado às restrições de igualdades

μ	vetor dos multiplicadores de Lagrange associado às restrições de desigualdades
μ_0	multiplicador de Lagrange associado à função objetivo
α	escalar
ε	tolerância
∂	derivada parcial
ξ	variável de folga
∇	gradiente

LISTA DE FIGURAS

Figura 3.1:	Fluxograma do algoritmo.....	40
Figura 3.2:	Rotina do LP-NLP.....	41
Figura 4.1:	Árvore “branch” e “bound” do problema mestre MILP.....	51
Figura 4.2:	Configuração dos processos.....	52
Figura 4.3:	Árvore “branch” e “bound” do problema mestre MILP.....	54
Figura 4.4:	Região viável e curvas de níveis da função objetivo.....	57
Figura 4.5:	Linearização.....	58
Figura 4.6:	Região viável reduzida.....	58
Figura 4.7:	Região viável e curvas de níveis da função objetivo.....	59
Figura 4.8:	Região viável reduzida.....	60
Figura 4.9:	Superestrutura.....	61
Figura 4.10:	Superestrutura SYNHEAT original para problema com duas correntes quentes e duas frias.....	67
Figura 4.11:	Superestrutura reduzida para cinco correntes frias e cinco correntes quentes.....	68
Figura 4.12:	Superestrutura reduzida para três correntes quentes e uma corrente fria.....	69
Figura 5.1:	Passos na fase II.....	75

LISTA DE TABELAS

Tabela 4.1:	Número de LPs e NLPs.....	49
Tabela 4.2:	Número de LPs e NLPs.....	55
Tabela 4.3:	Número de LPs e NLPs.....	55
Tabela 4.4:	Dados para três correntes quentes e uma corrente fria.....	69
Tabela 4.5:	Valores arbitrários para variáveis binárias.....	70
Tabela 4.6:	Resultados para custo anual.....	70
Tabela 4.7:	Número de NLPs e LPs.....	70
Tabela 4.8:	Resultados para temperaturas.....	71
Tabela 4.9:	Resultados para variáveis binárias.....	72

BRUNO A. M. S. JUNIOR
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE QUÍMICA
LABORATÓRIO DE QUÍMICA DE ALTA TEMPERATURA
AV. BRUNO NEVES, 255 - MARACÁ - RJ 21245-970

1 - INTRODUÇÃO

1.1 - Formulação do Problema

Sistemas de processos em engenharia é uma área rica em problemas de otimização. Muitos problemas de projetos de processos e problemas operacionais podem ser formulados como programação linear (LP), programação não linear (NLP), programação mista linear e inteira (MILP) ou programação mista não linear e inteira (MINLP).

Dentro destas formulações, os problemas de programação não linear formam um subconjunto maior do campo da programação matemática, em especial em exemplos de engenharia química que envolvem redes de trocadores de calor (Yee e Grossmann, 1990), síntese de fluxograma de processos (Kravanja e Grossmann, 1997) e projetos de colunas de destilação (Viswanathan e Grossmann, 1993).

Modelos de engenharia de processos freqüentemente envolvem variáveis discretas. Certas decisões são naturalmente discretas, por exemplo, o número de pratos na coluna de destilação. Problemas de otimização que envolvem variáveis discretas, como os descrito acima, são formulados como programações mista-inteira linear ou não linear e, geralmente, podem ser representados na seguinte forma

$$\begin{aligned} \min \quad & z = c^T y + f(x) \\ \text{sujeito a} \quad & h(x) + By \leq 0 \\ & x \in X \subseteq R^n \\ & y \in \{0,1\}^m \end{aligned} \tag{1.1}$$

onde “ x ” é o vetor de variáveis contínuas, “ y ” é o vetor de variáveis binárias 0-1, B e c são matriz e vetor de coeficientes, respectivamente, $f(x)$ e $h(x)$ são funções lineares ou não lineares e o sinal das desigualdades incluem o caso de igualdade.

Um aspecto comum desta classe de problemas é a não convexidade, os quais implicam dificuldades com a determinação de uma solução ótima global, com o uso de técnicas de programação matemática correntes. Por outro lado é importante como um primeiro passo o desenvolvimento de métodos melhorados para casos convexos, pois estes podem servir como base para desenvolvimento de métodos rigorosos ou aproximados para solução de problemas não convexos estruturados ou não.

Devido às razões mencionadas, associadas a fatores econômicos e ambientais, ultimamente há um crescente interesse em desenvolver e investigar novas técnicas para resolução de problemas MINLP.

1.2 - Teoria de Otimização

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, projeto de processo, controle de processo e análise numérica. Dentro

deste contexto de otimização, os problemas são tratados dentro das seguintes definições.

Função Objetivo

É a função matemática cujo máximo ou mínimo deseja-se determinar.

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, isto é, o grau de liberdade do sistema.

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).

Pontos Viáveis

São todos os pontos que satisfazem as restrições de igualdade e desigualdade.

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 localizam os pontos viáveis.

O método empregado para solucionar um problema de otimização depende, fundamentalmente, da forma da função objetivo e de suas restrições.

1.2.1 - Mínimos e Máximos

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

1.2.2 - Convexidade

Um subconjunto K de um espaço vetorial X é dito convexo se $\forall x_1, x_2 \in K$ e $0 \leq \alpha \leq 1$, tem-se $\alpha x_1 + (1 - \alpha)x_2 \in K$, apresentando as seguintes propriedades:

- i) $\beta K = \{x \in K / x = \beta y, y \in K\}$ é convexo para $\forall \beta \in \mathbb{R}$;
- ii) $K + L$ e $K \cap L$ são convexos para todo subconjunto convexo L de X ("+" opera sobre os elementos de cada subconjunto)

Sendo K um conjunto convexo não vazio do \mathbb{R}^n , a função $f: K \rightarrow \mathbb{R}$ é dita convexa se $\forall x_1, x_2 \in K$ e $0 \leq \alpha \leq 1$ tem-se $f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2)$, apresentando as seguintes propriedades:

- i) $f(x)$ é contínua em qualquer ponto do interior de K ;
- ii) as seções $K_\varepsilon = \{x \in K / f(x) \leq \varepsilon\}$ são conjuntos convexos.

A função $f(x)$ é estritamente convexa se a desigualdade for estrita $\forall x_1 \neq x_2 \in K$; . Uma função $s(x)$ é côncava se a função $f(x) = -s(x)$ for convexa.

Dentro das generalizações de funções convexas e côncavas, tem-se:

- i) $f(x)$ é quase-convexa se $f[\alpha x_1 + (1 - \alpha)x_2] \leq \max[f(x_1), f(x_2)]$
 $\forall 0 \leq \alpha \leq 1$ e $\forall x_1, x_2 \in K$;
- ii) $f(x)$ é quase-côncava se $f[\alpha x_1 + (1 - \alpha)x_2] \geq \min[f(x_1), f(x_2)]$
 $\forall 0 \leq \alpha \leq 1$ e $\forall x_1, x_2 \in K$;

- iii) $f(x)$ é pseudo-convexa se $\forall x_1, x_2 \in K$ e $f(x_1) < f(x_2)$, implicam em $(x_1 - x_2)^T \nabla f(x_2) < 0$;
- iv) $f(x)$ é pseudo-côncava se $\forall x_1, x_2 \in K$ e $f(x_1) > f(x_2)$, implicam em $(x_1 - x_2)^T \nabla f(x_2) > 0$.

1.2.3 - Condições Necessárias e Suficientes do Ótimo

1.2.3.1 - Para o caso sem restrição

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

$$\begin{aligned} \min f(x) \\ x \in R^n \end{aligned} \tag{1.2}$$

Neste caso, o problema de otimização depende somente da forma da função objetivo.

- Condição Necessária de Primeira Ordem

Para que x^* seja um mínimo local da função $f(x)$, diferenciável em x^* , é necessário que $\nabla f(x^*) = 0$

- Condição Necessária de Segunda Ordem

Para que x^* seja um mínimo local da função $f(x)$, duas vezes diferenciável em x^* , é necessário que $\nabla f(x^*) = 0$ e que $H(x^*) \equiv \nabla^2 f(x^*)$ seja positiva semidefinida, onde $H(x^*)$ é chamada de matriz Hessiana, sendo que para $\forall x \neq 0$, se:

$$x^T Hx \geq 0 \rightarrow H \text{ é positiva semidefinida}$$

$$x^T Hx > 0 \rightarrow H \text{ é positiva definida}$$

$$x^T Hx \leq 0 \rightarrow H \text{ é negativa semidefinida}$$

$$x^T Hx < 0 \rightarrow H \text{ é negativa definida}$$

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

- Condição Suficiente

Seja $f(x)$ duas vezes diferenciável em x^* tal que $\nabla f(x^*) = 0$ e $H(x^*)$ seja positiva definida então x^* é um mínimo local estrito de f .

Quando $f(x)$ é convexa, as condições para o ótimo simplificam-se, porque as condições de segunda ordem são equivalentes à convexidade local da função. Além disso, um mínimo local será também global.

1.2.3.2 - Para o caso com restrição

Na otimização com restrição o problema que está sendo resolvido é

$$\begin{aligned} & \min f(x) \\ & \text{sujeito a} \\ & \quad h(x) = 0 \\ & \quad g(x) \leq 0 \\ & \quad x \in X \subseteq R^n \end{aligned} \tag{1.3}$$

Conforme (1.3), a função objetivo $f(x)$ está sujeita às 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 da função objetivo e de suas restrições, os problemas de otimização com restrição são chamadas de

programação linear, programação quadrática, programação não linear, programação inteira e programação mista.

Uma idéia chave no desenvolvimento das condições necessárias e suficientes para problemas de otimização com restrição é transformá-las em problemas sem restrição e aplicar as condições de otimização para a determinação de pontos estacionários da função não restrita. Uma destas transformações envolve a introdução de uma função auxiliar, chamada a função de Lagrange, definida como

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

onde λ e μ são os vetores dos multiplicadores de Lagrange associados às restrições de igualdade e desigualdade, respectivamente. Os multiplicadores λ associados com as restrições de igualdade não são restringidos em sinal, enquanto os multiplicadores μ associados com as restrições de desigualdade são não negativos.

Para encontrar os pontos estacionários para a função de Lagrange, o problema (1.3) torna-se um problema não restrito, ou seja

$$\max_{\lambda, \mu} \min_x L(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x)$$

$$\text{com } \mu \geq 0 \text{ e } \begin{cases} \lambda > 0, & \text{para } h > 0 \\ \lambda < 0, & \text{para } h < 0 \end{cases}$$

Considerando o problema (1.3) somente com restrições de igualdade, ou seja:

$$\begin{aligned} & \min f(x) \\ & \text{sujeito a} \\ & h_i(x) = 0 \quad i = 1, 2, \dots, m < n \\ & x \in X \end{aligned} \tag{1.4}$$

Seja x^* um mínimo global de (1.4) para o qual os gradientes das restrições de igualdade são linearmente independentes. Perturbando o lado direito das restrições de igualdade, tem-se

$$\begin{aligned}
 & \min f(x) \\
 & \text{sujeito a} \\
 & h_i(x) = b_i \quad i = 1, 2, \dots, m < n \\
 & x \in X
 \end{aligned} \tag{1.5}$$

onde $b = (b_1, b_2, \dots, b_m)$ é o vetor perturbação.

Se o vetor perturbação muda, então a solução ótima de (1.5) e seus multiplicadores mudam, pois em geral $x = x(b)$ e $\lambda = \lambda(b)$. Logo, a função de Lagrange toma a forma

$$L(x, \lambda) = L(b) = f(x(b)) + \lambda(b)^T [h(x(b)) - b].$$

Considerando-se que o ponto estacionário de L corresponde ao mínimo global, então, $f(x^*) = L(x^*, \lambda^*)$.

Tomando o gradiente da função de Lagrange com respeito ao vetor perturbação “ b ” e rearranjando os termos obtém-se:

$$\nabla_b L = \left[\frac{\partial x}{\partial b} \right]^T \left(\nabla_x f + \left[\frac{\partial h}{\partial x} \right]^T \lambda \right) + \left[\frac{\partial \lambda}{\partial b} \right]^T (h(x) - b) - \lambda,$$

onde $\left[\frac{\partial x}{\partial b} \right]$ é uma matriz $n \times m$, $\left[\frac{\partial \lambda}{\partial b} \right]$ é uma matriz $m \times m$ e $\left[\frac{\partial h}{\partial x} \right]$ é uma matriz $m \times n$.

Note que os termos dentro do primeiro e segundo parênteses correspondem aos gradientes da função de Lagrange com respeito a x e a λ , respectivamente, e conseqüentemente eles são nulos devido às condições necessárias $\nabla_x L = \nabla_\lambda L = 0$. Então, chega-se a $\nabla_b L(x^*, \lambda^*) = -\lambda^*$ e, desde que x^* é um mínimo global e $\nabla_b h(x) = 0$, tem-se $\nabla_b f(x^*) = -\lambda^*$.

Por esta razão, os multiplicadores de Lagrange λ^* resultam informação sobre a sensibilidade da função objetivo com respeito ao vetor perturbação “ b ” para o ponto ótimo x^* .

- Condições Necessárias de Primeira Ordem - Condições de Fritz-John

Seja $x^* \in X$ uma solução viável de (1.3), sendo $f(x)$ e $g(x)$ diferenciáveis para x^* e $g(x)$ tendo as primeiras derivadas parciais contínuas para x^* . Então, se x^* é uma solução local de (1.3), existem multiplicadores de Lagrange μ_0, λ e μ tais que

$$\mu_0 \nabla f(x^*) + \lambda^T \nabla h(x^*) + \mu^T \nabla g(x^*) = 0$$

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

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

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

$$(\mu_0, \mu_j) \geq (0, 0) \quad j = 1, 2, \dots, p$$

$$(\mu_0, \lambda, \mu) \neq (0, 0, 0)$$

onde $\nabla f(x^*)$ é um vetor de dimensão n , $\nabla h(x^*)$ é uma matriz $m \times n$ e $\nabla g(x^*)$ é uma matriz $p \times n$, μ_0 é um escalar, λ e μ são vetores de dimensões m e p , respectivamente.

As restrições $\{\mu_j g_j(x^*) = 0, j = 1, 2, \dots, p\}$ são chamadas restrições complementares.

Quando $\mu_0 = 0$, as condições necessárias de primeira de ordem de Fritz-John não utilizam o gradiente da função objetivo. Assim sendo, as condições do gradiente representam uma combinação linear das restrições de desigualdade ativas e restrições de igualdade. Em tais casos, as soluções de Fritz-John não são proveitosas na identificação de um ótimo local da função $f(x)$. Um número de condições adicionais são necessárias para garantir $\mu_0 > 0$. Estas são as qualificações das restrições de primeira ordem que são apresentadas a seguir.

Seja x^* um ótimo local de (1.3), X um conjunto aberto, o conjunto J definido como $J = \{j : g_j(x^*) = 0\}$ e $g_j(x), h_i(x)$, com $i = 1, 2, \dots, m$ diferenciáveis continuamente para x^* .

- *Qualificação das Restrições de Independência Linear*

Os gradientes $\nabla h_i(x^*) \forall i$ e $\nabla g_j(x^*)_{j \in J}$ são linearmente independentes.

- *Qualificação da Restrição de Slater*

As restrições $g_j(x^*)$ são pseudo-convexas para x^* , as restrições $h_i(x^*)$ são quase-convexas e quase-côncavas, os gradientes de $\nabla h_i(x^*) \forall i$ são linearmente independentes, e existe um $\bar{x} \in X$, tal que $h_i(\bar{x}) = 0 \forall i$ e $g_j(\bar{x}) < 0 \quad j \in J$.

- *Qualificação das Restrições de Kuhn-Tucker*

Existe um vetor não nulo $z \in \mathbb{R}^n$ para o qual $\{z^T \nabla h_i(x^*) = 0 \text{ e } z^T \nabla g_j(x^*) \leq 0 \quad j \in J\}$, implica que existe uma função vetorial não dimensional $w(\tau)$ no intervalo $[0,1]$, tal que:

i) $w(0) = x^*$

ii) $w(\tau) \in X$ para $0 \leq \tau \leq 1$

iii) w é uma vez diferenciável para $\tau = 0$, e $\frac{dw(0)}{d\tau} = \lambda z$ para alguns $\lambda > 0$.

- *Qualificação da Restrição de Convexo Inverso Fraco*

As restrições $h(x)$ e $g(x)$ são continuamente diferenciáveis para x^* . Cada $g_j(x^*)_{j \in J}$ é pseudo-côncava para x^* ou linear e cada $h_i(x^*) \forall i$ é pseudo-convexa e pseudo-côncava para x^* .

- **Condições Necessárias de Kuhn-Tucker**

Suponha que as funções f , h_i e g_j são continuamente diferenciáveis e que as condições da qualificação de restrição mantém-se para x^* . Então a condição necessária para x^* ser um ótimo local de (1.3) é a existência de números λ_i (não restritos em sinal) e $\mu_j \geq 0$, tal que:

$$\begin{cases} \nabla f(x^*) + \lambda^T \nabla h(x^*) + \mu^T \nabla g(x^*) = 0 \\ \mu_j g_j(x^*) = 0 \quad j = 1, 2, \dots, p \end{cases}$$

- Condições Necessárias de Primeira Ordem - Condições Karush-Kuhn-Tucker (K.K.T.)

Seja $x^* \in X$ a solução viável de (1.3) e $f(x)$, $g(x)$ diferenciáveis para x^* e as primeiras derivadas parciais de $h(x)$ são contínuas para x^* . Se x^* é um ótimo local de (1.3) e uma das seguintes restrições

- i) independência linear;
- ii) Slater;
- iii) Kuhn-Tucker ou
- iv) convexo inverso fraco

é satisfeita, então existem multiplicadores de Lagrange λ , μ tal que

$$\nabla f(x^*) + \lambda^T \nabla h(x^*) + \mu^T \nabla g(x^*) = 0$$

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

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

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

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

- Condições Suficientes de Primeira Ordem - Condições de Karush-Kuhn-Tucker

Seja $x^* \in X$ uma solução viável de (1.3), sendo também x^* um ponto K.K.T e definindo $I^+ = \{i : \lambda_i > 0\}$ e $I = \{i : \lambda_i < 0\}$ para x^* . Se $f(x)$ é pseudo-convexa para x^* , $h_i(x)$ para $i \in I^+$ são quase-convexas para x^* , $h_i(x)$ para $i \in I$ são quase-côncava para x^* e $g_j(x)$ para $j \in J$ são quase-convexas para x^* sendo que todos os outros pontos x são viáveis para $h_i(x)$ e $g_j(x)$, então x^* é um ótimo global de (1.3).

Se as condições acima estão restritas dentro de uma bola de raio ε centrada em x^* , então x^* é um ótimo local de (1.3).

As condições de otimização de primeira ordem utilizam informação somente sobre os gradientes da função objetivo e restrições, sendo que a curvatura das funções, medida pelas segundas derivadas, não é considerada, logo as condições necessárias de primeira ordem não resultam informação completa.

- Qualificação da Restrição de Segunda Ordem

Seja $x^* \in X$ um ponto viável do problema (1.3) e as funções $g(x)$ e $h(x)$ duas vezes diferenciáveis, sendo z um vetor qualquer não nulo, tal que

$$\begin{cases} z^T \nabla h_i(x^*) = 0 & i = 1, 2, \dots, m \\ z^T \nabla g_j(x^*) = 0 & j \in J \equiv \{j : g_j(x^*) = 0\} \end{cases} e$$

então a qualificação da restrição de segunda ordem mantém-se para x^* se z é a tangente de um *arc* $w(\tau)$ duas vezes diferenciáveis que começa para x^* , ao longo da qual $\frac{\partial w}{\partial \tau} = \lambda z$ para algum λ positivo, $w(0) = x^*$, onde $0 \leq \tau \leq \varepsilon$ com $\varepsilon > 0$,

$$h_i(w(\tau)) \equiv 0 \quad \forall i \quad e \quad g_j(w(\tau)) \equiv 0 \quad j \in J.$$

- Condições Necessárias de Segunda Ordem

Seja x^* um ótimo local do problema (1.3), as funções $f(x)$, $h(x)$ e $g(x)$ duas vezes continuamente diferenciáveis e mantida a qualificação de restrição de segunda ordem para x^* . Se existem multiplicadores de Lagrange λ^*, μ^* satisfazendo as condições necessárias de primeira ordem K.K.T e se para cada vetor não nulo z tem-se

$$\begin{cases} z^T \nabla h_i(x^*) = 0 & i = 1, 2, \dots, m \quad e \\ z^T \nabla g_j(x^*) = 0 & j \in J \equiv \{j : g_j(x^*) = 0\} \end{cases}$$

então, $z^T \nabla^2 L(x^*, \lambda^*, \mu^*) z \geq 0$, onde $L(x^*, \lambda^*, \mu^*) = f(x^*) + (\lambda^*)^T h(x^*) + (\mu^*)^T g(x^*)$.

- Condições Suficientes de Segunda Ordem

Sejam $f(x)$, $h(x)$, $g(x)$ duas vezes diferenciáveis, e assume-se que existem multiplicadores de Lagrange λ^* , μ^* satisfazendo as condições necessárias de primeira ordem K.K.T.

Se para todo vetor não nulo z :

$$\begin{cases} z^T \nabla h_i(x^*) = 0 & i = 1, 2, \dots, m, \quad e \\ z^T \nabla g_j(x^*) = 0 & j \in J_1 \equiv \{j : g_j(x^*) = 0, \mu_j > 0\} \\ z^T \nabla g_j(x^*) \leq 0 & j \in J_2 \equiv \{j : g_j(x^*) = 0, \mu_j = 0\} \end{cases}$$

segue que,

$$z^T \nabla^2 L(x^*, \lambda^*, \mu^*) z > 0,$$

então x^* é um mínimo local estrito do problema (1.3).

Nos trabalhos de Cooper e Steinberg(1970), Minoux(1986), Floudas(1995), Secchi(1993), Himmelblau(1972), encontram-se maiores informações sobre a teoria de otimização e aplicações.

1.3 – Estrutura da Dissertação

O capítulo 1 introduziu definições básicas e formulações da parte teórica de otimização para problemas com restrição ou não, discutindo as condições necessárias e suficientes para o ótimo.

Na seqüência, o capítulo 2, faz uma descrição matemática através da formulação geral dos problemas MINLP, tratando a seguir de uma revisão das técnicas

de otimização global estocástica e determinística para resolvê-los, analisando as idéias-chaves de algoritmos determinísticos correntes como “*Branch*” e “*Bound*” Espacial, Decomposição Generalizada de Benders (GBD), Aproximações Externas (OA), Aproximações Externas com Relaxamento de Igualdade (OA/ER), Aproximações Externas com Relaxamento de Igualdade e Penalidade Aumentada (OA/ER/AP), Aproximações Externa Generalizada (GOA), Decomposição Cruzada Generalizada (GCD) e Otimização Global (GOP). Após são descritas algumas aplicações dos problemas MINLP em redes de trocadores de calor, síntese de fluxogramas de processo e projetos de colunas de destilação.

O capítulo 3 descreve o algoritmo proposto para problemas MINLP convexos, o qual faz uma geração dinâmica das aproximações lineares, as quais são derivadas para soluções inteiras na árvore para limitar o problema original, evitando a solução seqüencial de problemas mestres MILP e problemas NLP. Subproblemas NLP são resolvidos para soluções inteiras para as variáveis binárias. Os passos do algoritmo para implementação são apresentados para uma melhor compreensão e juntamente com os fluxogramas do programa principal e das subrotinas. Também é levado em consideração as formas de introdução das restrições adicionais.

Exemplos convexos são resolvidos no capítulo seguinte, pois o algoritmo foi implementado com esta finalidade. Também resolve-se exemplos não convexos para analisar o comportamento do algoritmo. Faz-se uma conexão com o capítulo 2, resolvendo um exemplo aplicado a rede de trocadores de calor

Finalmente o último capítulo trata das conclusões e sugere uma modificação do algoritmo proposto neste trabalho, através da introdução de testes locais e globais para a implementação de um algoritmo para problemas MINLP não convexos.

2 - REVISÃO BIBLIOGRÁFICA

2.1 - Descrição Matemática

A formulação geral dos problemas de programação mista, não linear e inteira (MINLP) é dada por

$$\begin{aligned} & \min_{x,y} f(x,y) \\ & \text{sujeito a} \\ & h(x,y) = 0 \\ & g(x,y) \leq 0 \\ & x \in X \subseteq R^n \\ & y \in Y_{\text{inteiro}} \subseteq I^r \end{aligned} \tag{2.1}$$

onde:

x representa um vetor de "n" variáveis contínuas

y representa um vetor de "r" variáveis inteiras

$h(x,y) = 0$ são "m" restrições de igualdade
 $g(x,y) \leq 0$ são "p" restrições de desigualdade
 $f(x,y)$ é a função objetivo .

As variáveis inteiras y com limites superiores e inferiores dados, $y^l \leq y \leq y^u$, podem ser expressas como variáveis binárias, isto é, 0-1, denotada como Z , pela seguinte fórmula:

$$y = y^l + z_1 + 2z_2 + 4z_3 + \dots + 2^{N-1} z_N$$

onde N é o número mínimo de variáveis 0-1 necessárias. Este número mínimo é dado por:

$$N = 1 + \text{INT} \left\{ \frac{\log(y^u - y^l)}{\log 2} \right\}$$

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.

Logo, a formulação de (2.1) pode ser escrita em termos de variáveis 0-1

como

$$\begin{aligned}
 & \min_{x,y} f(x,y) \\
 & \text{sujeito a} \\
 & h(x,y) = 0 \\
 & g(x,y) \leq 0 \\
 & x \in X \subseteq R^n \\
 & y \in Y = \{0,1\}^q
 \end{aligned} \tag{2.2}$$

onde y é um vetor de "q" variáveis 0-1.

Tratando com modelos de otimização mista não linear e inteira da forma (2.1) ou (2.2) surgem dificuldades que estão associadas com a natureza do problema, isto é, o domínio combinatório (domínio- y) e o domínio contínuo (domínio- x).

A maior parte dos métodos numéricos para a solução dos problemas MINLPs preocupam-se com a determinação do mínimo local. Muitas das aplicações

da engenharia de processos levam para MINLPs não convexos, que possuem múltiplos mínimos locais.

Neste capítulo será feita uma revisão bibliográfica dos métodos estocásticos e determinísticos existentes para localizar o ótimo global de problemas MINLPs.

Floudas(1995) descreve um excelente texto sobre fundamentos de otimização mista não linear e inteira.

2.2 - Técnicas de Otimização Global Estocástica

Técnicas de otimização global estocástica, segundo Smith(1996), empregam alguns elementos de aleatoriedade na busca do ótimo global e através de um argumento estatístico verificam a convergência. Este argumento é baseado na premissa geral que quanto mais tempo permanecer buscando o ótimo global, a probabilidade que este seja encontrado é maior.

A vantagem desta técnica é que não faz suposição a priori sobre a estrutura do problema a ser resolvido. Entretanto, para problemas altamente restritos, não existe garantia que um ponto ótimo local seja encontrado.

Estes algoritmos são aplicados para problemas relativamente restritos, os quais possuem muitos mínimos locais (por exemplo, planejamento de processos) e, para tais problemas, uma solução ótima global próxima é encontrada rapidamente. Esta propriedade conduz tais algoritmos a serem aplicados para resolver problemas organizacionais grandes, produzindo boas soluções em tempo computacional relativamente baixo, fazendo deles candidatos potenciais para aplicações "online" (Matthews,1995).

Os primeiros algoritmos estocásticos de otimização não eram adaptáveis para problemas com muitas restrições e ou variáveis contínuas tais como encontradas em projeto de processos contínuos. Entretanto variantes mais sofisticadas desta técnica estão surgindo com o tempo, como por exemplo, os trabalhos de Secchi e Perlingeiro

(1989) e Secchi e Biscaia Jr. (1989) que utilizam uma técnica de otimização global estocástica adaptativa para problemas NLP de dimensões elevadas.

O trabalho de Paules e Floudas (1992) apresenta uma aproximação de programação estocástica de dois estágios com recurso MINLP para problemas de síntese de seqüências de destilação com integração de calor. A estratégia proposta combina os aspectos de uma aproximação da superestrutura com uma partição das variáveis projetadas em duas classes: estrutural e periódica. A classe estrutural compromete-se com elementos fixos do projeto em todas as condições de operação, enquanto as periódicas são quantidades ajustáveis que podem ser escolhidas para a operação ótimas em cada período.

2.3 - Técnicas de Otimização Global Determinística

Técnicas de otimização global determinística, segundo Smith(1996), garantem localização do ótimo global dentro de uma tolerância especificada ϵ . Esta tolerância define a máxima diferença entre o valor da função objetivo obtida e o verdadeiro ponto ótimo global.

Uma vantagem das técnicas determinísticas é que as restrições são consideradas explicitamente e por esta razão são adaptadas para problemas mais restritos do que técnicas estocásticas.

A desvantagem de muitos algoritmos é que eles exploram aspectos matemáticos específicos da estrutura do problema para garantir a otimização global e devido a isto podem ser somente aplicados para problemas que possuem esta estrutura especial.

Técnicas determinísticas obtiveram sucesso aplicadas a problemas relativamente pequeno em áreas de planejamento, projeto de processos e síntese de rede de trocadores de calor.

A seguir são apresentados algoritmos determinísticos para a solução de problemas MINLP.

2.3.1 – “Branch” e “Bound” Espacial

O método de busca em árvore, chamado de “*branch*” e “*bound*”, teve seus princípios enunciados na década de 60, por Land e Doig(1960). Desde então, muitas implementações práticas surgiram em função da obtenção dos limites (“*bounds*”), da seleção do vértice a ser ramificado (“*branching*”) e da escolha da variável pela qual a ramificação deve ser realizada.

O método de “*branch*” e “*bound*” espacial, descrito no trabalho de Smith e Pantelides(1997), produz limites superiores e inferiores rigorosos para o valor da função objetivo sobre cada subdomínio considerado. Um ponto viável dentro de um subdomínio correspondendo ao limite superior é também identificado. O subdomínio é particionado em dois ou mais subdomínios por ramificação no espaço das variáveis. O procedimento continua até os limites superiores e inferiores convergirem para dentro da requerida tolerância ϵ , ou até ser provado que um subdomínio é inviável globalmente, ou até os limites inferiores rigorosos para o subdomínio excederem o valor da função objetivo do melhor ponto viável que foi encontrado até o momento, indicando que não existe possibilidade de melhorar este valor no subdomínio em avaliação.

Existem muitas implementações do “*branch*” e “*bound*” espacial para otimização global, todos os quais usam o melhor ponto viável disponível obtido via otimização local como o limite superior.

Raman e Grossmann(1994) propuseram uma estrutura para problemas de programação discreta na forma de disjunções lógicas, equações e lógica proposicional. A transformação de formulação lógica em forma de equação não é sempre desejável e que por esta razão existe uma necessidade para tratar a solução de problemas de programação mista inteira com algumas relações mista-inteira expressas como disjunções e outras expressas como restrições algébricas. Uma caracterização teórica das restrições disjuntivas é propor o que pode servir como um critério para decidir se a disjunção pode ser transformada em forma de equação. O algoritmo proposto é uma extensão do método de Raman e Grossmann(1993) para tratar as disjunções com inequações, o qual é restrito para o caso de equações e inequações

lineares e relações lógicas simbólicas que podem ser expressas como lógicas proposicionais. O método também é implementado com um LP baseado no esquema “*branch e bound*” no qual o uso de inequações violadas a partir de relações lógicas são consideradas como discutido em Raman e Grossmann(1993).

2.3.2 - Decomposição Generalizada de Benders (GBD)

Geoffrion(1972) generalizou a aproximação proposta por Benders(1962), para explorar a estrutura de problemas de programação matemática (2.2). A idéia básica de Geoffrion é gerar, para cada iteração, um limite superior e um limite inferior sobre a busca da solução do modelo MINLP. O limite superior resulta a partir do problema primal, enquanto o limite inferior resulta a partir do problema mestre. O problema primal corresponde a (2.2) com as variáveis y fixas, e esta solução produz informação relativa ao limite superior e aos multiplicadores de Lagrange associados com as restrições de igualdade e desigualdade. 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 (2.2) sobre o espaço- y ;
- ii) representação dual de V , onde $V = \{ y : h(x,y) = 0, g(x,y) \leq 0, \text{ para alguns } x \in X \}$;
- iii) representação dual da projeção do problema (2.2) sobre o espaço- y .

Com as iterações ocorridas é mostrado que a seqüência de limites superiores alterados não cresce e a seqüência de limites inferiores não decresce, logo, as seqüências convergem em um número finito de iterações.

Muitas vezes os modelos MINLP exibem estruturas especiais (por exemplo, gráficos, redes, funções separáveis), as quais podem ser exploradas para o desenvolvimento de soluções especializadas. O GBD tem a vantagem de explorar

estruturas especiais nos subproblemas NLP, porém requer um número grande de iterações, onde os subproblemas NLP e problemas mistos MILP necessitam ser resolvidos sucessivamente.

2.3.3 - Aproximações Externas (OA)

Duran e Grossmann(1986a, 1986b) propuseram um algoritmo, 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 do problema (2.3)

$$\begin{aligned} & \min c y + f(x) \\ & \text{sujeito} \\ & g(x) + B y \leq 0 \\ & x \in X = \{x : x \in R^n, A_1 x \leq a_1\} \subseteq R^n \\ & y \in Y = \{y : y \in \{0,1\}^q, A_2 y \leq a_2\} \end{aligned} \tag{2.3}$$

O limite inferior resulta a partir da solução do problema mestre. O problema mestre é derivado usando informação primal, que consiste do ponto solução x^k do primal e é baseado sobre uma 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 $y = y^k \in Y$. 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 as quais melhoram o relaxamento e por esta razão melhoram o limite inferior.

Com as iterações ocorridas, duas seqüências de limites superiores e inferiores modificados 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.

O que distingue o OA do GBD é que o problema mestre é formulado baseado sobre informação primal e linearizações externas.

O trabalho desenvolvido por Quesada e Grossmann(1992) apresenta o método “*branch*” e “*bound*” para problemas MINLP convexos, baseado na solução de subproblemas NLP e LP, evitando a solução seqüencial de subproblemas NLP e problemas mestres MILP que é exigida na implementação padrão dos algoritmos GBD e OA.

Raman e Grossmann (1992) propuseram uma estrutura quantitativa para a integração de conhecimento heurístico e lógico, os quais são expressos em forma de lógica proposicional, em modelos de otimização MINLPs para síntese de processos. Como discutido em Raman e Grossmann(1991), o conhecimento qualitativo heurístico e lógico quando dado em forma de lógica proposicional é expresso matematicamente como restrições mistas inteiras. Baseado nesta representação, dois esquemas básicos são sugeridos para integrar conhecimento qualitativo dentro do modelo de otimização de programação mista inteira (MILP ou MINLP) para problemas de síntese, enquanto o conhecimento quantitativo é integrado ao nível do algoritmo de otimização MINLP, sendo que estes algoritmos de busca referem-se aos métodos GBD e OA.

Turkay e Grossmann(1996) apresentaram um algoritmo OA Lógico Básico no qual o problema discreto-contínuo é modelado como programa disjuntivo generalizado. Este modelo envolve disjunção lógica com equações não lineares e relações lógicas puras. A principal vantagem para programas disjuntivos generalizado está na robustez e eficiência computacionais quando comparados com algoritmos e modelos MINLP algébricos.

Uma revisão dos métodos “*branch*” e “*bound*”, GBD e OA, relações entre eles e referências para aplicações em engenharia de processos pode ser encontrado em Grossmann e Kravanja (1995).

2.3.4 - Aproximações Externas com Relaxamento de Igualdade (OA/ER)

Para lidar com restrições de igualdade não linear 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.

$$\begin{aligned}
 & \min_{x,y} c^T y + f(x) \\
 & \text{sujeito a} \\
 & h(x) = 0 \\
 & g(x) \leq 0 \\
 & Cx + By \leq 0 \\
 & x \in X = \{x : x \in R^n, A_1 x \leq a_1\} \subseteq R^n \\
 & y \in Y = \{y : y \in \{0,1\}^q, A_2 y \leq a_2\}
 \end{aligned} \tag{2.4}$$

A idéia básica do OA/ER é relaxar as restrições de igualdade não lineares em desigualdades e subseqüentemente 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.

O algoritmo OA/ER tem a vantagem de exigir um número menor de iterações, porém o tamanho de seus problemas mestres MILP é considerado grande em relação ao GBD.

Os algoritmos “branch” e “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.

2.3.5 - Aproximações Externas com Relaxamento de Igualdade e Penalidade Aumentada (OA/ER/AP)

Viswanathan e Grossmann(1990) propuseram o algoritmo OA/ER/AP, 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 (2.4) 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 de (2.4).

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. 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.

Vecchiotti e Grossmann(1997), formularam um modelo híbrido para problemas não linear discreto-contínuo para engenharia de sistemas de processos. O modelo envolve variáveis disjuntivas e binárias e restrições inteiras ou mistas inteiras. A formulação algébrica e disjuntiva pode ser deduzida como um caso particular, ao passo que para a formulação híbrida, é introduzido um novo algoritmo para a solução,

que é uma extensão do OA Lógico Básico. Estas idéias foram implementadas no LOGMIP (Programa Misto, Inteiro e Lógico), um código de computador, escrito em C, para resolver problemas de otimização não linear discreto-contínuo, onde os problemas podem ser escritos de três formas diferentes, ou seja:

- problemas algébricos (**PA**)
- problemas disjuntivos (**PD**)
- problemas híbridos (**PH**)

O programa tem uma rotina de reconhecimento de modelos, para analisar o tipo de modelo, a qual trabalha como segue:

- i) se as disjunções não são detectadas no modelo, o problema é MINLP (**PA**), então o algoritmo OA/ER/AP por Viswanathan e Grossmann(1990) é aplicado.
- ii) Se as disjunções são detectadas, então poderá conter variáveis binárias ou não.
 - se não contém variáveis binárias, o problema é **PD** e aplica-se o algoritmo OA Lógico Básico por Turkay e Grossmann(1996).
 - Se contém variáveis binárias, o problema é **PH**, então uma extensão do OA Lógico Básico é usado.

Tal algoritmo para resolver o problema na representação híbrida, decompõe o problema **PH** em dois conjuntos de subproblemas, o subproblema NLP (Problema não Linear) e o de subproblema MILP (Problema Misto Linear e Inteiro). O número de problemas NLP iniciais para serem resolvidos têm que ser determinados a priori. Dependendo do problema, isto pode ser feito de modo sistemático ou especificado pelo usuário.

Identificado os subproblemas NLP iniciais, as variáveis booleanas são fixadas para cada problema, e as variáveis binárias poderão ser fixadas ou relaxadas. Após resolver os subproblemas NLP iniciais, obtém-se um limite superior deste conjunto. Desta forma, o problema mestre MILP é resolvido determinando um limite inferior e os valores das variáveis binárias e booleanas para o próximo NLP. Se os limites inferior e superior ficarem dentro de uma tolerância, pára-se, caso contrário as

iterações continuam até ser encontrado a convergência. A novidade deste programa é a capacidade para lidar com as disjunções.

Alguns resultados mostram que, para determinados casos, o uso de disjunções é a melhor alternativa de modelagem comparados ao MINLP algébrico. O algoritmo para ser aplicado para um dado problema depende do tipo de equações e restrições. Para problemas convexos, os modelos disjuntivo e híbrido mostraram uma melhora significativa comparado ao caso algébrico. Para problemas não convexos a principal vantagem parece existir no fato que os modelos disjuntivo e híbrido resistem às soluções sub-ótimas fracas.

Em geral, modelos disjuntivo e híbrido são superiores à modelos algébricos em termos de tempo computacional e qualidade de soluções.

O trabalho desenvolvido por Kravanja e Grossmann(1994) relata novas capacidades que foram implementadas em PROSYN (ver Kravanja e Grossmann (1990)), um pacote que realiza otimização de MINLPs sobre superestruturas para fluxogramas de processos.

A principal característica do PROSYN é que ele possibilita execução automatizada de topologias simultâneas e otimização de parâmetros de processos. Quando PROSYN é aplicado para vários tipos de problemas, surgem limitações como o efeito da não convexidade que surgem em modelos de processos não lineares, podendo impedir a determinação de solução global ótima e o ambiente flexível para simplificar diferentes tipos de cálculos. O objetivo do trabalho de Kravanja e Grossmann(1994) foi de discutir e implementar estratégias para estas limitações, as quais foram feitas através da extensão da estratégia de modelagem e decomposição, desenvolvida por Kocis e Grossmann(1989) , algoritmo OA/ER/AP.

2.3.6 - Aproximação Externa Generalizada (GOA)

Fletcher e Leyffer(1994) estenderam a aproximação OA para problemas MINLP do tipo (2.5), introduzindo funções penalidades exatas, criando o algoritmo GOA.

$$\begin{aligned} & \min_{x,y} f(x,y) \\ & \text{sujeito a} \\ & g(x,y) \leq 0 \\ & x \in X \subseteq R^n \\ & y \in Y = \{0,1\}^q \end{aligned} \tag{2.5}$$

sobre as seguintes condições:

C_1 : X é um conjunto convexo, compacto, não vazio e as funções

$$f: R^n \times R^q \rightarrow R$$

$$g: R^n \times R^q \rightarrow R^p$$

são convexas.

C_2 : f e g são continuamente diferenciáveis.

O GOA é similar ao OA, com as principais diferenças sendo:

i) tratamento da inviabilidade;

Se o problema primal $P(y^k)$ é inviável, então o seguinte problema viável geral é dado por:

$$\begin{aligned} & \min_x \sum_{i \in I^1} w_i g_i^+(x, y^k) \\ & \text{sujeito a} \\ & g_i(x, y^k) \leq 0, \quad i \in I \\ & x \in X \end{aligned} \tag{2.6}$$

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^1 é o conjunto de restrições de desigualdade inviáveis.

Note que o problema (2.6) tem dentro da função objetivo um somatório somente das restrições de desigualdade inviáveis. Fletcher e Leyffer(1994) propuseram o seguinte lema para o problema (2.6).

Lema: Se o problema primal $P(y^k)$ é inviável de modo que x^k resolve o problema (2.6) e tem $\sum_{i \in I'} w_i g_i^+(x^k, y^k) > 0$

Então, y^k é inviável dentro das restrições

$$0 \geq g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \quad \forall i \in \{I' \cup I\} \quad \forall x \in X$$

Observe que as restrições mencionadas anteriormente são as linearizações das restrições de desigualdade não lineares em torno do ponto (x^k, y^k) . Então o lema acima afirma que 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.

ii) nova formulação do problema mestre que considera a inviabilidade explicitamente, baseado nas seguintes idéias:

- projeção de (2.5) sobre o espaço y , ou seja:

$$\min_y \min_x f(x, y)$$

sujeito a

(2.7)

$$g(x, y) \leq 0$$

$$x \in X$$

$$y \in Y$$

- aproximação externa da função objetivo e da região viável utilizando o lema anterior.

iii) tratamento unificado de funções penalidades exatas

2.3.7 - Decomposição Cruzada Generalizada (GCD)

Holmberg(1990) generalizou a aproximação proposta por Van Roy(1983) para classe de problemas de otimização como

$$\begin{aligned} & \min_{x,y} f(x,y) \\ & \text{sujeito a} \\ & h(x,y) = 0 \\ & g(x,y) \leq 0 \\ & x \in X \subseteq R^n \\ & y \in Y = \{0,1\}^q \end{aligned} \tag{2.8}$$

sob as seguintes condições:

C₁: As funções:

$$f : R^n \times R^q \rightarrow R$$

$$h : R^n \times R^q \rightarrow R^m$$

$$g : R^n \times R^q \rightarrow R^p$$

são funções convexas para cada $y(\text{fixo}) \in Y = \{0,1\}^q$

C₂: X é um conjunto convexo, compacto, não vazio e as funções f, h, g são limitadas e Lipschitziana sobre (X,Y).

C₃: 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 de (2.8) e multiplicadores de Lagrange λ^k, μ^k para o subproblema dual. O subproblema dual produz um limite inferior sobre a solução de (2.8) 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 subsequente 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 consome mais tempo de CPU que os problemas da fase I.

2.3.8 - Otimização Global (GOP)

Um grande número de problemas de programação não linear podem ser escritos da seguinte forma:

$$\begin{aligned}
 & \min_{x,y} f(x,y) \\
 & \text{sujeito a} \\
 & \quad h(x,y) = 0 \\
 & \quad g(x,y) \leq 0 \\
 & \quad x \in X \\
 & \quad y \in Y
 \end{aligned} \tag{2.9}$$

onde $f(x,y)$, $g(x,y)$ e $h(x,y)$ podem ser funções não lineares conduzindo para não convexidade do problema.

Floudas e Visweswaran (1990b) propuseram uma nova aproximação de otimização global determinística para resolver problemas da forma (2.9) que satisfazem condições determinadas por Floudas e Visweswaran (1990a), ou seja:

- i) $f(x,y)$ é convexo em x para todo y fixo, e convexo em y para todo x fixo;
- ii) $g(x,y)$ é convexo em x para todo y fixo, e convexo em y para todo x fixo;
- iii) $h(x,y)$ é afim em x para todo y fixo e afim em y para todo x fixo.

O algoritmo GOP usa teoria de dualidade para decompor (2.9) em subproblemas dual relaxado e primal, os quais são então resolvidos utilizando a convexidade do problema projetado no espaço de subconjuntos x e y . O algoritmo provou ter convergência finita para um ótimo global limitado a problemas convexos nas projeções em X e Y .

Este trabalho discute as propriedades matemáticas de algumas classes de problemas de otimização e apresenta a aplicação do algoritmo GOP para algumas dessas classes que tem estrutura especial.

2.4 - Aplicações

Neste item serão apresentadas referências com pequena descrição de trabalhos desenvolvidos em redes de trocadores de calor, síntese de fluxogramas de processos e projeto de colunas de destilação.

2.4.1 – Síntese de Rede de Trocadores de Calor

Para fazer uma síntese da rede de trocadores de calor(HEN) é necessário formular e solucionar problemas MINLP.

As funções não lineares tais como: diferença de temperatura média logarítmica(LMTD), equação de transferência de calor, balanço de energia para misturadores ou separadores e a função custo, introduzem não convexidade para síntese HEN. Termos não convexos fazem a solução de modelos matemáticos mais difíceis e causam complicações numéricas.

Viswanathan e Grossmann(1990) usaram estratégias heurísticas para reduzir o efeito da não convexidade, mas nenhuma técnica MINLP disponível garante convergência da solução ótima global quando problemas MINLP não são convexos.

O trabalho de Zamora e Grossmann(1997) apresenta um algoritmo de otimização MINLP global para as síntese de redes de trocadores de calor sem divisões

de correntes. Este algoritmo lida com a não convexidade introduzida pela LMTD, a equação da transferência de calor e a função de custo no modelo proposto por Yee e Grossmann(1990). O algoritmo apresenta uma nova classe de planos aproximados que limita a LMTD, e um modelo MINLP convexo que é restringido linearmente, e limites inferiores são encontrados para o custo anual total mínimo da rede. Este modelo MINLP convexo é embutido dentro do algoritmo “branch” e “bound” que realiza uma busca espacial no domínio das temperaturas da rede. A solução do modelo MINLP limitado inferiormente resulta em um conjunto de configurações promissoras da rede que são otimizadas globalmente para busca da configuração da rede ótima global e condições de operações. Este último conjunto de problemas NLP global é resolvido aplicando a versão especializada do algoritmo “branch” e “contract” proposto por Zamora e Grossmann (1996) para a otimização global de problemas com termos bilineares e linear fracionários.

Um outro algoritmo de otimização para rede de trocadores de calor é proposta por Quesada e Grossmann(1993), o qual baseia-se sobre um novo problema subestimado que envolve funções estimadoras não lineares e lineares, que produzem uma aproximação exata nos limites da região viável. O problema subestimado, que dá ascensão para um problema NLP convexo que prediz limites inferiores estendidos do problema original, é acoplado com um procedimento de busca “branch” e “bound” espacial, para encontrar a solução ótima global.

Novak, Kravanja e Grossmann (1996) desenvolveram um esquema de linearização e inicialização especial para síntese simultânea de rede de trocadores de calor.

2.4.2 - Síntese de Fluxograma de Processos

Conceitos de sínteses de fluxograma de processos são baseados sobre geração heurística, modificação evolutiva, otimização de execução de tarefas ou otimização de superestrutura.

A vantagem do conceito de otimização de execução de tarefas sobre o conceito de otimização de superestrutura é que possibilita considerar diferentes fenômenos simultaneamente a partir do ponto de vista do desempenho do processo total.

Por outro lado, o conceito da superestrutura envolve integração de tarefa, e conseqüentemente, a topologia do fluxograma é representada de um modo mais natural e direto. Entretanto, um dos principais problemas com a aproximação da superestrutura é que esta é limitada para problemas de tamanho médio devido a complexidade para a capacidade limitada dos algoritmos de otimização MINLP.

Kravanja e Grossmann(1997) propuseram uma estrutura, a qual é possível postular, pré-selecionar e otimizar as superestruturas de uma forma mais sistemática. Eles propuseram uma aproximação hierárquica de multiníveis para a síntese MINLP de fluxogramas de processos. Seguindo a estratégia hierárquica, foi postulada a superestrutura para diferentes níveis de representação dos alternativos fluxogramas e estes modelados para os níveis correspondentes de agregação e complexidade. Usando um procedimento de pré-seleção, a superestrutura é otimizada de forma mais eficaz e confiável.

Kocis e Grossmann(1989), propuseram uma estratégia de modelagem e decomposição para explorar a estrutura especial de problemas de sínteses de fluxogramas que são resolvidos com o algoritmo OA/ER. O objetivo deste procedimento é reduzir o esforço computacional exigido para resolver problemas de otimização MINLP, e reduzir o efeito que a não convexidade pode ter na rejeição do ótimo global.

2.4.3 – Projeto de Colunas de Destilação

Modelos MINLP para encontrar a localização ótima para as alimentações e número de pratos exigidos para a separação específica em uma coluna de destilação com alimentações múltiplas é apresentada no trabalho de Viswanathan e

Grossmann(1993). O algoritmo OA/ER/AP mostrou nos resultados deste trabalho ser uma ferramenta robusta para resolver tais problemas.

3 - DESCRIÇÃO DO ALGORITMO

Após uma revisão dos métodos de resolução dos problemas MINLP, feita no capítulo 2, concluí-se que os algoritmos para resolver tais problemas podem ser classificados em três categorias principais:

- “*Branch*” e “*Bound*”
- *Decomposição Generalizada de Benders (GBD)*
- *Aproximações Externas (OA)*

Os algoritmos GBD e OA têm a limitação do tamanho do problema mestre(MILP) que cresce a medida que as iterações ocorrem e este é 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.

Quesada e Grossmann(1992) propuseram um algoritmo para evitar os problemas relatados acima, como acontece nos algoritmos GBD e OA, melhorando a

eficiência da solução dos problemas MINLP convexos e reduzindo o trabalho computacional exigido para resolver os problemas mestres MILP.

O algoritmo proposto por Quesada e Grossmann(1992) 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*” e “*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.

O algoritmo proposto neste trabalho baseia-se no algoritmo de Quesada e Grossmann(1992) com modificações na introdução das aproximações lineares.

3.1 - Descrição Detalhada do Algoritmo

Para descrever detalhadamente o algoritmo proposto neste capítulo, baseado no algoritmo de Quesada e Grossmann(1992), com algumas modificações, considera-se os problemas MINLP convexos, modelado como

$$z = \min_{x,y} c^T y + f(x)$$

sujeito a

$$By + g(x) \leq 0$$

$$x \in X = \{x / x \in R^n, x^l \leq x \leq x^u\}$$

$$y \in Y = \{y / y \in \{0,1\}^m, Ay \leq a\}$$

(3.1)

Para implementação do algoritmo, na solução de (3.1), segue-se os seguintes passos:

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 (3.1) transforma-se num subproblema NLP, ou seja:

$$\begin{aligned}
 z &= \min_x c^T y_0 + f(x) \\
 \text{sujeito a} & \\
 &By_0 + g(x) \leq 0 \\
 &x \in X
 \end{aligned}
 \tag{3.2}$$

O problema (3.2) 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 (3.1).

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 (3.1) são linearizadas, usando a solução ótima “ x_0 ” do subproblema NLP (3.2), resultando no seguinte problema MILP.

$$\begin{aligned}
 z &= \min \alpha \\
 \text{sujeito a} & \\
 &\alpha \geq c^T y + f(x_0) + \nabla f(x_0)^T (x - x_0) \\
 &By + g(x_0) + \nabla g(x_0)^T (x - x_0) \leq 0 \\
 &\alpha \in R, y \in Y, x \in X
 \end{aligned}
 \tag{3.3}$$

Passo 4 – O problema (3.3) 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 (3.1). 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 resolvidos, ou seja, nenhum nó aberto na árvore, então o limite superior corrente z_u é a solução ótima do problema (3.1) 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 $(LP)_j$ e se $z_j + \varepsilon_z \geq zu$, 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 $z_{NLP} < zu$, faz-se $zu = z_{NLP}$. Se $z_{NLP} \geq zu$, 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 $z_j \geq zu$ e retornar para o passo 5.

Os passos do algoritmo estão representados através de dois fluxogramas, conforme figuras 3.1 e 3.2. O programa principal, conforme figura 3.1, chama a rotina LP-NLP, conforme a figura 3.2, a qual é chamada a cada problema LP a ser resolvido, onde define-se como:

nó \rightarrow número de nós na árvore.

k \rightarrow nível do nó (nó atual).

flag = 1 \rightarrow solução não inteira.

flag = 0 \rightarrow solução inteira.

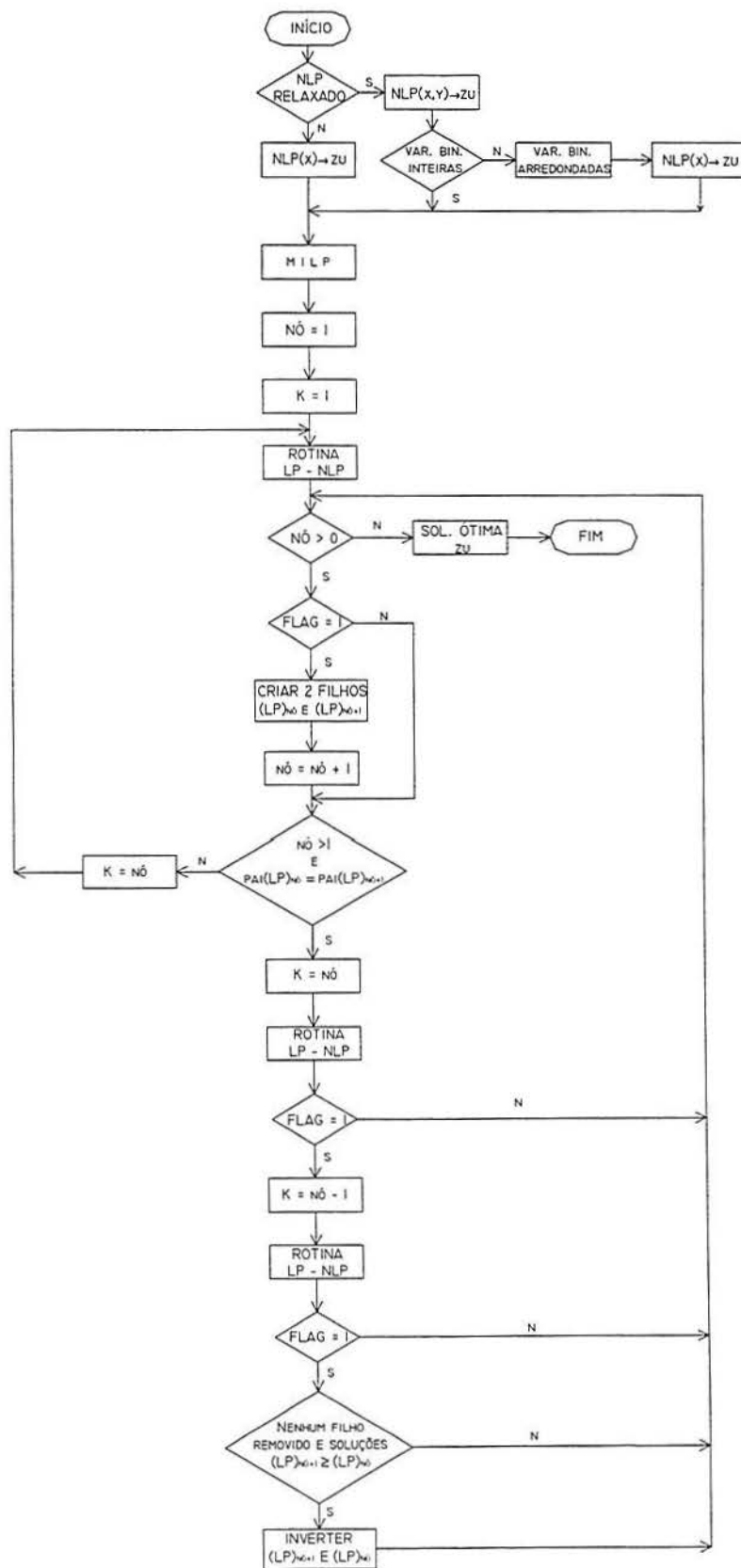


Figura 3.1: Fluxograma do algoritmo

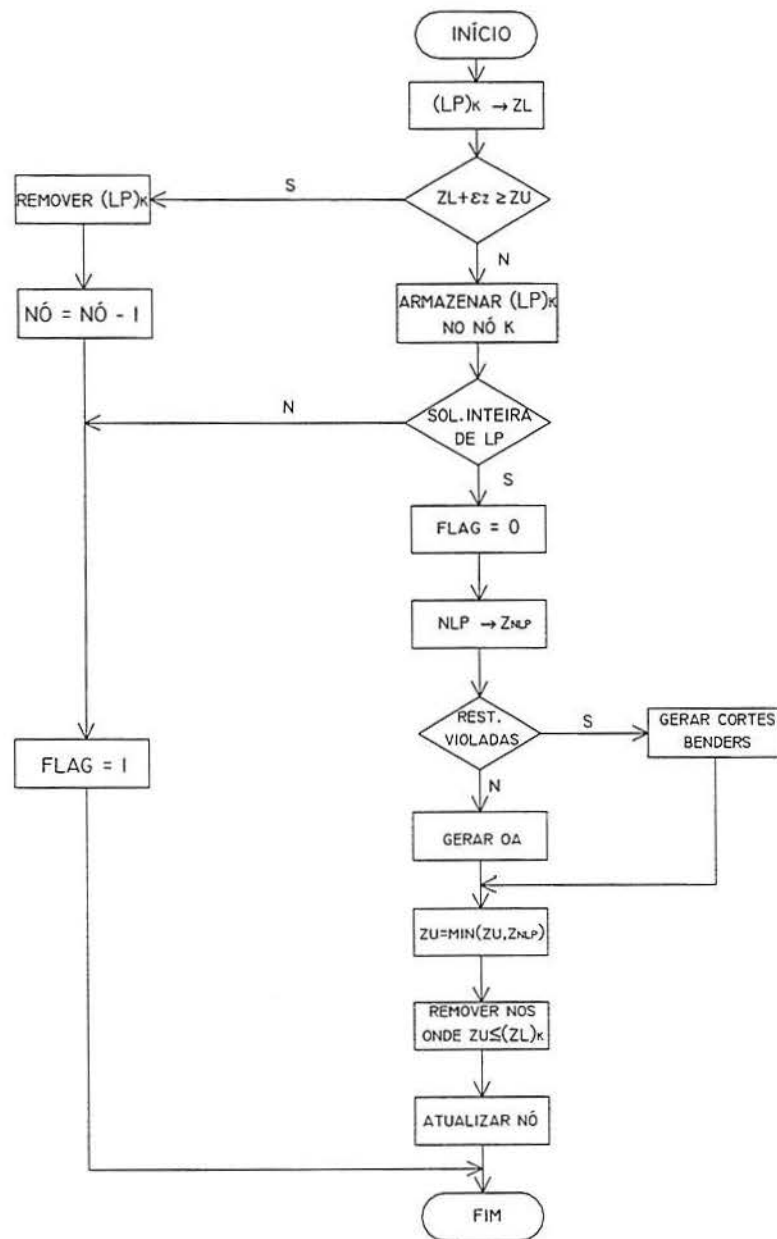


Figura 3.2: Rotina do LP-NLP

3.2 - 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.

3.2.1 - 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:

$$\begin{aligned}
 & z = \min \alpha \\
 & \text{sujeito a} \\
 & \alpha \geq c^T y + f(x^k) + \nabla f(x^k)^T (x - x^k) \\
 & By + g(x^k) + \nabla g(x^k)^T (x - x^k) \leq 0 \\
 & k = 1, 2, \dots, K_{(NLPs)} \\
 & \alpha \in R^l, y \in Y, x \in X
 \end{aligned} \tag{3.4}$$

onde “ x^k ” é a solução viável ou inviável do k-ésimo subproblema NLP.

3.2.2 - 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 de (3.4) 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:

$$\nabla f(x^k)^T (x - x^k) + (\lambda^k)^T \nabla g(x^k)^T (x - x^k) = 0$$

Da segunda inequação de (3.4), tem-se $\nabla g(x^k)^T (x - x^k) \leq -[By + g(x^k)]$

$$\text{Então, } \nabla f(x^k)^T (x - x^k) - (\lambda^k)^T [By + g(x^k)] = 0$$

Logo, a primeira inequação de (3.4), reduz-se a:

$$\alpha \geq c^T y + f(x^k) + (\lambda^k)^T [By + g(x^k)]$$

Desta forma, a formulação do problema mestre para os cortes de Benders é dado por:

$$\begin{aligned} z = \min \quad & \alpha \\ \text{sujeito a} \quad & \\ & \alpha \geq c^T y + f(x^k) + (\lambda^k)^T [By + g(x^k)] \\ & \quad k = 1, 2, \dots, K_{(NLPs \text{ viáveis})} \\ & (\lambda^k)^T [By + g(x^k)] \leq 0 \\ & \quad k = 1, 2, \dots, K_{(NLPs \text{ inviáveis})} \\ & \alpha \in R^1, y \in Y \end{aligned} \tag{3.5}$$

onde (x, λ) 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 nova 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.

3.2.3 - Restrição Aproximada proposta por Quesada e Grossmann(1992)

A partir do problema original MINLP (3.1), 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.

$$\begin{aligned}
 z &= \min c^T y + a^T w + r(v) \\
 \text{sujeito a} & \\
 Cy + Dw + t(v) &\leq 0 \\
 Ey + Fw + Gv &\leq b \\
 y \in Y, w \in W, v \in V &
 \end{aligned}
 \tag{3.6}$$

Nesta representação $f(x) = a^T w + r(v)$, $B^T = [C \ E]^T$, $g(x) = [Dw + t(v) \ Fw + Gv]$ e $X = W \times V$.

O problema (3.6) é reformulado com a adição de duas variáveis contínuas (α, β) para representar as partes lineares e não lineares da função objetivo, ou seja:

$$\begin{aligned}
 z &= \min \alpha \\
 \text{sujeito a} & \\
 Cy + Dw + t(v) &\leq 0 \\
 r(v) &\leq \beta \\
 Ey + Fw + Gv &\leq b \\
 c^T y + a^T w + \beta - \alpha &= 0 \\
 y \in Y, w \in W, v \in V, \alpha \in R, \beta \in R &
 \end{aligned}
 \tag{3.7}$$

Fazendo aproximações externas em (3.7) para o ponto (w^k, v^k) gerado pelo k -ésimo subproblema NLP tem-se

$$\begin{aligned}
 & z = \min \alpha \\
 & \text{sujeito a} \\
 & Cy + Dw + t(v) + \nabla t(v) (v - v^k) \leq 0 \\
 & r(v) + \nabla r(v)^T (v - v^k) \leq \beta \\
 & Ey + Fw + Gv \leq b \\
 & c^T y + a^T w + \beta - \alpha = 0
 \end{aligned} \tag{3.8}$$

Se as condições de Kuhn-Tucker do subproblema NLP em (3.6) são consideradas em relação as variáveis não lineares “v”, então:

$$\nabla r(v) + \nabla t(v) \lambda + G^T \mu = 0 \tag{3.9}$$

Então, multiplicando (3.9) por $(v - v^k)$, tem-se:

$$\nabla r(v^k)^T (v - v^k) + (\mu^k)^T G(v - v^k) = -(\lambda^k)^T \nabla t(v^k)^T (v - v^k) \tag{3.10}$$

De (3.8), tem-se que $\nabla t(v^k) (v - v^k) \leq -Cy - Dw - t(v^k)$, então:

$$\begin{aligned}
 \nabla r(v^k)^T (v - v^k) + (\mu^k)^T G(v - v^k) &\geq (\lambda^k)^T [Cy + Dw + t(v^k)] \\
 \nabla r(v^k)^T (v - v^k) &\geq (\lambda^k)^T [Cy + Dw + t(v^k)] - (\mu^k)^T G(v - v^k)
 \end{aligned} \tag{3.11}$$

Finalmente, substituindo (3.11) na segunda inequação de (3.8), tem-se o MILP:

$$\begin{aligned}
 & z = \min \alpha \\
 & \text{sujeito a} \\
 & \beta \geq r(v^k) + (\lambda^k)^T [Cy + Dw + t(v^k)] - (\mu^k)^T G(v - v^k) \\
 & Ey + Fw + Gv \leq b \\
 & c^T y + a^T w + \beta - \alpha = 0 \\
 & k = 1, 2, \dots, K_{(NLPs)} \quad x \in X, y \in Y, \alpha \in R
 \end{aligned} \tag{3.12}$$

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.

3.2.4 - Restrição Aproximada proposta neste trabalho

Considerando o problema MINLP original modelado com

$$z = \min_{x,y} c^T y + f(x) + \mu u$$

sujeito a

$$By + g(x) \leq u$$

$$x \in X = \{x / x \in R^n, x^l \leq x \leq x^u\}$$

$$y \in Y = \{y / y \in \{0,1\}^m, Ay \leq a\}$$

(3.13)

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 em (3.13) para o ponto " x^l ", o qual foi gerado pelo primeiro subproblema NLP, tem-se o MILP

$$z = \min \alpha$$

sujeito a

$$\alpha \geq c^T y + f(x) + \nabla f(x)^T (x - x^l) + \mu u$$

$$By + g(x) + \nabla g(x)^T (x - x^l) \leq u$$

$$x \in X, y \in Y, \alpha \in R$$

(3.14)

A partir do relaxamento das variáveis binárias em (3.14), 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:

$$\begin{aligned}
 & z = \min \alpha \\
 & \text{sujeito a} \\
 & \alpha \geq c^T y + f(x^i) + \nabla f(x^i)^T (x - x^i) + \mu u \\
 & By + g(x^i) + \nabla g(x^i)^T (x - x^i) \leq u \\
 & \alpha \geq c^T y + f(x^k) + \nabla f(x^k)^T (x - x^k) + \mu u \\
 & \qquad \qquad \qquad k = 1, 2, \dots, K_{(NLPs \text{ viáveis})} \\
 & \alpha \geq c^T y + f(x^k) + (\lambda^k)^T [By + g(x^k) - u] + \mu u \\
 & \qquad \qquad \qquad k = 1, 2, \dots, K_{(NLPs \text{ inviáveis})} \\
 & x \in X, y \in Y, \alpha \in R^1, u \in R^l
 \end{aligned}
 \tag{3.15}$$

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.

3.3 - Implementação do Algoritmo

O algoritmo proposto neste trabalho foi implementado em MATLAB para as versões 4.2 ou superior. Para a solução dos subproblemas de programação não linear utilizou-se o algoritmo de programação quadrática sequencial (SQP) com a fórmula BFGS para atualizar a estimativa da matriz Hessiana (Broyden,1970; Fletcher,1970; Goldfarb,1970; Shanno,1970), através da rotina CONSTR do MATLAB. Para a solução dos problemas de programação linear utilizou-se o algoritmo simplex (Danzig, 1951), através da rotina LP do MATLAB.

Anexo, encontram-se as rotinas MINLP.m e LP_NLP.m, representadas através de seus fluxogramas nas figuras 3.1 e 3.2, respectivamente. A rotina MINLP.m estrutura o algoritmo “branch” e “bound” espacial. A subrotina LP_NLP é chamada para resolver os problemas de LP, com as condições de integralidade relaxadas, e os problemas de NLP quando soluções inteiras são obtidas nos LPs.

4 - RESULTADOS

Neste capítulo serão resolvidos exemplos convexos e não convexos, analisando o comportamento do algoritmo para exemplos não convexos, em vista de que o algoritmo foi desenvolvido para problemas MINLP convexos. Também é analisada a solução de problemas dentro do item aplicações.

4.1 - Exemplos Convexos

- *Exemplo 1*

O exemplo é proposto por Floudas(1995) e tem convexidade em “ x ” e linearidade em “ y ”. O MINLP é modelado como:

$$z = \min_{x,y} y_1 + y_2 + y_3 + 5x^2$$

sujeito a

$$3x - y_1 - y_2 \leq 0$$

$$-x + 0.1y_2 + 0.25y_3 \leq 0$$

$$y_1 + y_2 + y_3 \geq 2$$

$$y_1 + y_2 + 2(y_3 - 1) \geq 0$$

$$0.2 \leq x \leq 1 \quad y \in \{0,1\}^3$$
(4.1)

A solução ótima de (4.1) é encontrada para $z = 2.2$ com $x = 0.2$ e $y = (1,1,0)$. Caso seja arbitrado o ponto inicial para as variáveis binárias $(1,1,0)$, o algoritmo termina com apenas uma iteração, ou seja é resolvido um NLPs e LPs.

A tabela 4.1 mostra o comportamento do algoritmo em relação ao número de NLPs e LPs para alguns valores iniciais de variáveis binárias “y”.

y	NLPs	LPs
(1,0,0)	3	7
(1,1,1)	2	4
(1,0,1)	2	6
(0,0,0)	2	4
(0,1,0)	3	5
(0,1,1)	3	5
(0,0,1)	3	8

Tabela 4.1: Número de LPs e NLPs

- Exemplo 2

Seja o problema MINLP dado por Floudas(1995) modelado como:

$$\begin{aligned}
 z &= \min_{x,y} -y + 2x_1 + x_2 \\
 \text{sujeito a} & \\
 x_1 - 2e^{-x_2} &= 0 \\
 -x_1 + x_2 + y &\leq 0 \\
 0.5 \leq x_1, x_2 &\leq 1.4 \quad y \in \{0,1\}
 \end{aligned} \tag{4.2}$$

Fazendo $x = 2e^{-x_2}$ em (4.2), tem-se:

$$\begin{aligned}
 z &= \min_{x,y} -y + 4e^{-x} + x \\
 \text{sujeito a} & \\
 -2e^{-x} + x + y &\leq 0 \\
 0.5 \leq x &\leq 1.4 \quad y \in \{0,1\}
 \end{aligned} \tag{4.3}$$

O algoritmo encontra a solução ótima do problema (4.3) para $x = 0.853$ e $y = 0$, sendo $z = 2.5578$. Quando o valor inicial arbitrado para “y” é zero, então o algoritmo resolve um NLP e três LPs, independente do valor de “x” arbitrado. Quando o valor inicial arbitrado é $y = 1$, o algoritmo encontra a solução ótima após resolver dois NLPs e quatro LPs, independente do valor inicial de x , o que pode ser analisado através da figura 4.1. O subproblema NLP inicial é inviável, logo o limite superior $zu = \infty$ e o primeiro nó resulta em um limite inferior $zl = 2.2105$ para a solução ótima do MINLP. Este nó produz uma solução inteira $y = 0$ com limite inferior, $zl = 2.43$. Conseqüentemente, um subproblema NLP é resolvido, produzindo $zu = 2.5578$. Os nós viáveis que estão abaixo do limite superior conservam-se abertos, logo o nó “2” conserva-se aberto e não foi ramificado, então ele é modificado por adição de novas restrições aproximadas (nó “3” e nó “4”), os quais são podados pois estes limites inferiores são maiores que o limite superior atual. Logo a solução do MINLP é $z = 2.5578$.

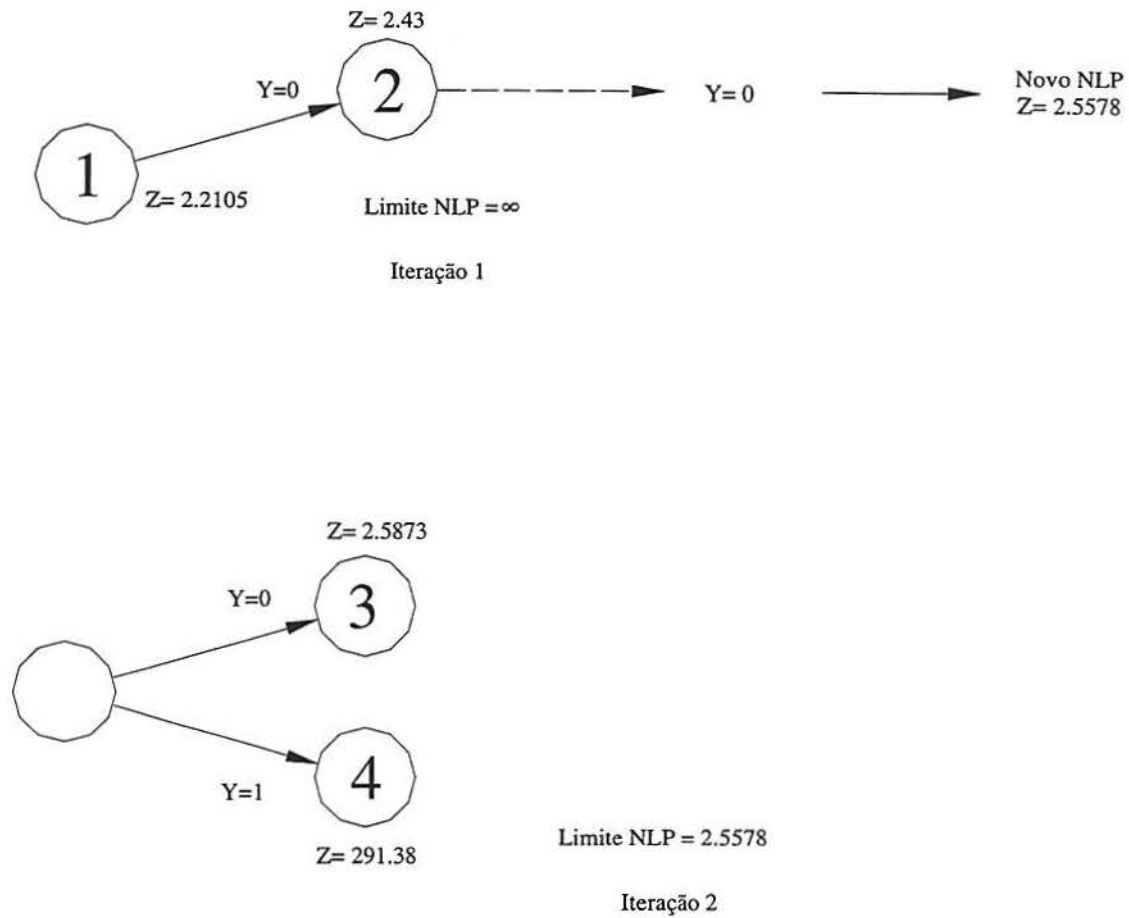


Figura 4.1: Árvore “branch” e “bound” do problema mestre MILP

Para este exemplo quando o valor inicial arbitrado para a variável binária é $y = 1$, são examinados quatro nós, resolvendo-se dois subproblemas NLPs, mas devido a simplicidade do caso, observa-se que não há necessidade de aplicar o algoritmo proposto, pois existem apenas duas possibilidades de soluções inteiras, ou seja, $y = 0$ ou $y = 1$, logo para o problema ser resolvido, seria suficiente a resolução de dois subproblemas NLPs.

- Exemplo 3

Considerando o problema proposto por Kocis e Grossmann(1987) definindo a melhor configuração para os processos dados conforme figura 4.2.

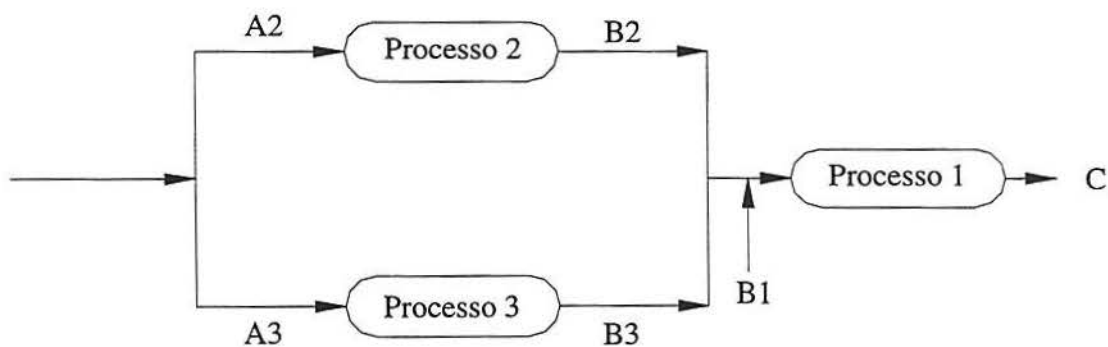


Figura 4.2: Configuração dos processos

A formulação do modelo MINLP, o qual é um problema convexo é dado por:

$$z = \min - [11C - 7B1 - B2 - 1.2B3 + 1.8(A2 + A3) - 3.5y_1 - y_2 - 1.5y_3]$$

sujeito a (4.4)

$$C = 0.9B$$

$$B2 = \log(1 + A2)$$

$$B3 = 1.2\log(1 + A3)$$

$$B = B1 + B2 + B3$$

$$C \leq y_1$$

$$B2 \leq 10y_2$$

$$B3 \leq 10y_3$$

$$y_2 + y_3 \leq 1$$

$$y_1, y_2, y_3 \in \{0, 1\}$$

$$C, B1, B2, B3, A2, A3 \geq 0$$

As variáveis y_1 , y_2 e y_3 definem a existência ou não dos processos 1, 2 e 3, respectivamente.

Para implementação, as restrições de igualdade do problema (4.4) foram eliminadas, ou seja, foram feitas as transformações a seguir.

Sendo $C = 0.9B$ e $B = B1 + B2 + B3$, então $C = 0.9(B1 + B2 + B3)$, mas

$B2 = \log(1+A2)$ e $B3 = 1.2 \log(1+A3)$, então $C = 0.9(B1 + \log(1+A2) + 1.2\log(1+A3))$, logo temos como variáveis do sistema $A2$, $A3$ e $B1$.

Fazendo $A2 = x_1$, $A3 = x_2$ e $B1 = x_3$, o problema (4.4) pode ser escrito da seguinte forma:

$$z = \min_{x,y} -2.9x_3 - 8.9\log(1+x_1) - 10.44\log(1+x_2) + 1.8x_1 + 1.8x_2 + 3.5y_1 + y_2 + 1.5y_3$$

sujeito a (4.5)

$$-y_1 + 0.9\log(1+x_1) + 1.08\log(1+x_2) + 0.9x_3 \leq 0$$

$$-10y_2 + \log(1+x_1) \leq 0$$

$$-10y_3 + 1.2\log(1+x_2) \leq 0$$

$$y_2 + y_3 - 1 \leq 0$$

Neste exemplo, usando os valores iniciais para as variáveis binárias $y_0 = (0,1,0)$ e para variáveis contínuas $x_0 = (0,0,1)$, uma busca é então conduzida, avaliando os nós “1” e “2”, de acordo com a figura 4.3. O NLP inicial resulta em um limite superior, $z_u = 1.0$ e o primeiro nó resulta em um limite inferior, $z_l = -4.333$ para a solução ótima do MINLP.

O nó “2” produz a solução inteira $y = (1,0,1)$ com um limite inferior de z igual a -3 , ou seja $z_l = -3$ (figura 4.3). Para esta configuração, um segundo subproblema NLP é resolvido o qual produz um limite superior para z , $z_u = -1.923$.

O limite superior, para podar nós na árvore, é determinado através das soluções dos subproblemas NLPs. Então os nós viáveis que estão abaixo deste limite são conservados abertos. Com a resolução do segundo subproblema NLP, novas linearizações são adicionadas para os nós abertos, estendendo a representação linear da região viável. O nó “2” não foi ramificado, então ele é modificado por adição de novas restrições aproximadas (ver nó “3” na figura 4.3). Este nó pode ser podado pois a solução é $z_l = -1.092$, sendo este limite inferior maior que o limite superior atual. Retrocedendo, o nó “1” já foi ramificado, então ele não é modificado, logo um novo nó é criado, ou seja, o nó “4”. Uma nova solução inteira é encontrada na seqüência,

$y = (1, 1, 0)$ e o subproblema NLP correspondente é resolvido dando um limite superior mais alto, $z_u = -1.72$. Consequentemente, $z_u = -1.923$ é que tem o melhor limite superior.

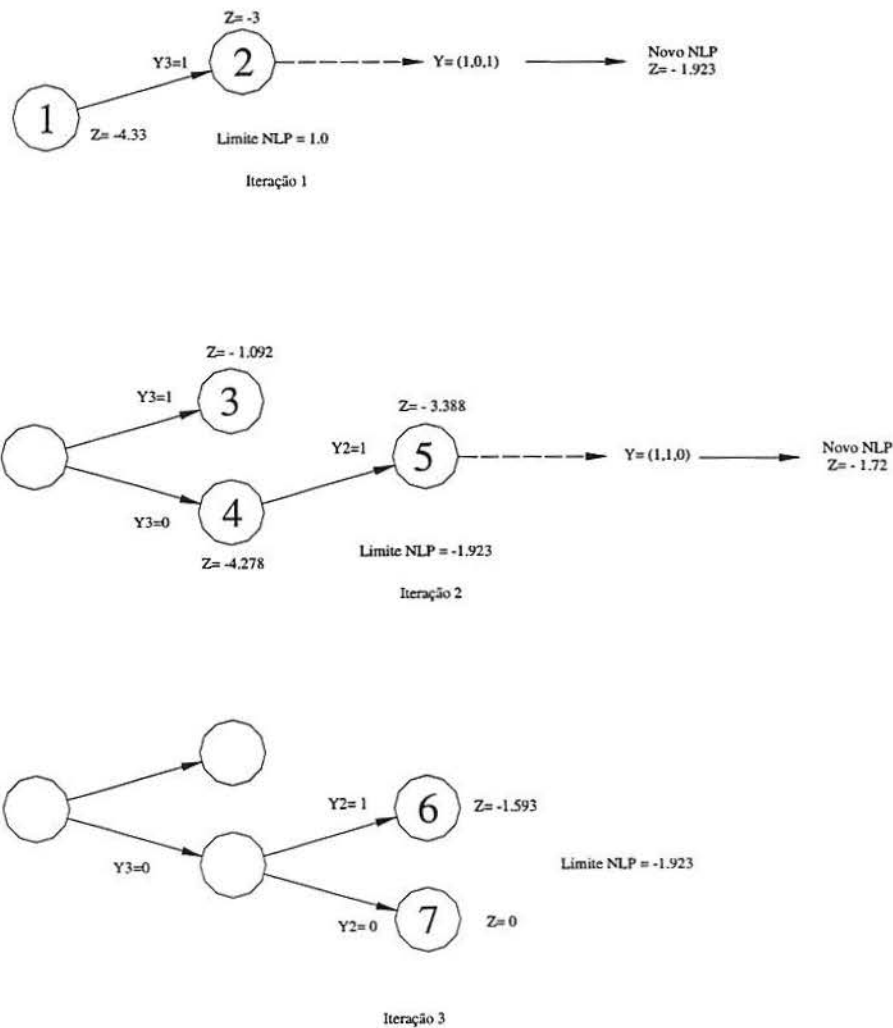


Figura 4.3: Árvore “branch” e “bound” do problema mestre MILP

Novamente, novas aproximações externas são adicionadas para nós abertos “6” e “7”, mas as soluções dos respectivos LPs excedem o limite superior corrente (ver figura 4.3). Consequentemente a busca para este ponto pode ser encerrada para confirmar que $z = -1.923$ é a solução ótima do MINLP, sendo $x = (0, 1.5242, 0)$ e $y = (1, 0, 1)$. Note que com o método proposto, a busca termina após examinar 7 nós para o nível “branch” e “bound”, onde foram resolvidos 3 subproblemas NLP.

A tabela 4.2 mostra os resultados do número de LPs para cada NLP resolvido na implementação dos algoritmos GBD padrão, GBD modificado proposto por Quesada e Grossmann(1992) e o proposto neste trabalho.

NLP	Número de LPs		
	GBD padrão	GBD modificado	Proposto
1	5	2	2
2	3	3	3
3	5	2	2

Tabela 4.2: Número de LPs e NLPs

Ainda dentro deste exemplo, fez-se um teste para valores iniciais de variáveis binárias e contínuas, $y_0 = (1,0,1)$ e $x_0 = (0,0,1)$, respectivamente. A solução ótima é encontrada pelo algoritmo proposto, terminando a busca após examinar 5 nós para o nível "branch" e "bound", sendo resolvido 2 NLP. A tabela 4.3 mostra a comparação deste método com GBD padrão e GBD modificado.

NLP	Número de LPs		
	GBD padrão	GBD modificado	Proposto
1	4	4	4
2	1	1	1
3	1	1	-
4	1	2	-
5	2	-	-

Tabela 4.3: Número de LPs e NLPs

O algoritmo proposto tem um bom desempenho para exemplos convexos e quando comparado com outros algoritmos, tem no mínimo desempenho semelhante.

4.2 - Exemplos Não Convexos

Apesar do algoritmo ser desenvolvido para exemplos convexos, sentiu-se a necessidade de testá-lo para exemplos não convexos e analisar suas soluções e estudar a existência de soluções não globais e o porque destas soluções.

- Exemplo 1

Considerando o problema proposto por Kocis e Grossmann(1988), modelado como:

$$\begin{aligned}
 z &= \min_{x,y} 2x + y \\
 \text{sujeito a} & \\
 1.25 - x^2 - y &\leq 0 \\
 x + y &\leq 1.6 \\
 0 \leq x \leq 1.6 \quad y &\in \{0,1\}
 \end{aligned} \tag{4.6}$$

onde a restrição de desigualdade não linear contém um termo não convexo para a variável contínua “x”.

O ótimo global do problema é localizado para $x = 0.5$ e $y = 1$, onde o valor da função objetivo é $z = 2$, sendo que também existe uma solução sub-ótima para $x = 1.118$ e $y = 0$, onde $z = 2.236$. A região viável do problema relaxado ($0 \leq y \leq 1$) e curvas de níveis da função objetivo quando $z = 2$ e $z = 2.236$ são mostradas na figura 4.4.

Na aplicação do algoritmo para este problema, quando o ponto inicial selecionado é $y = 1$, a solução ótima é encontrada, mas quando é selecionado $y = 0$ para ponto inicial, obtém-se a solução sub-ótima do problema.

A explicação do ocorrido está no fato de que quando assume-se $y = 0$, um subproblema NLP encontra a solução $x_1 = 1.118$. Para formular o problema mestre MILP, é feita uma linearização para este ponto, conforme figura 4.5, resultando no MILP dado por:

$$z = \min \alpha$$

sujeito a

$$\alpha \geq 2x + y$$

$$-2.236x - y \leq -2.5$$

$$x + y \leq 1.6$$

$$0 \leq x \leq 1.6 \quad y \in \{0, 1\}$$
(4.7)

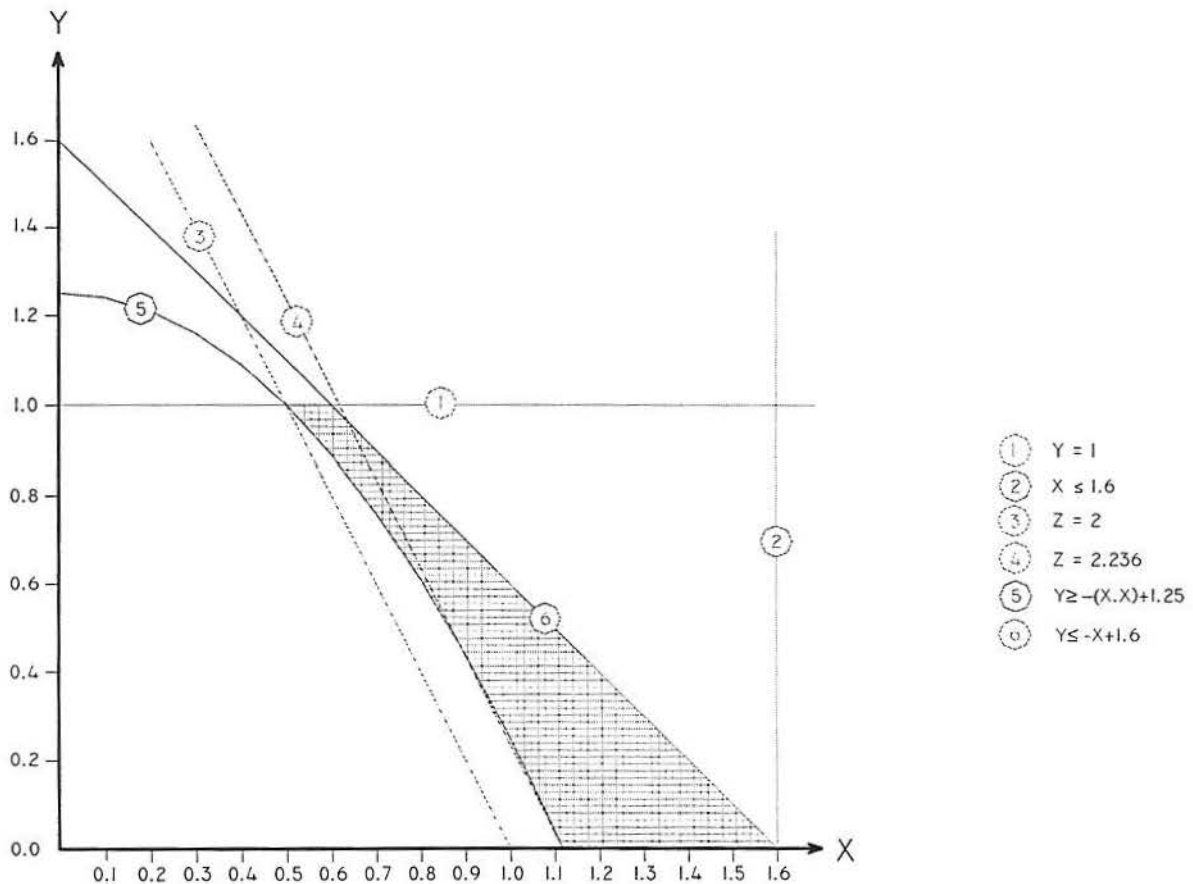


Figura 4.4: Região viável e curvas de níveis da função objetivo

Esta linearização faz com que a região viável seja reduzida e a solução ótima $x = 0.5$ e $y = 1$, onde $z = 2$ se encontre fora da região viável, o que pode ser confirmado na representação da figura 4.6.

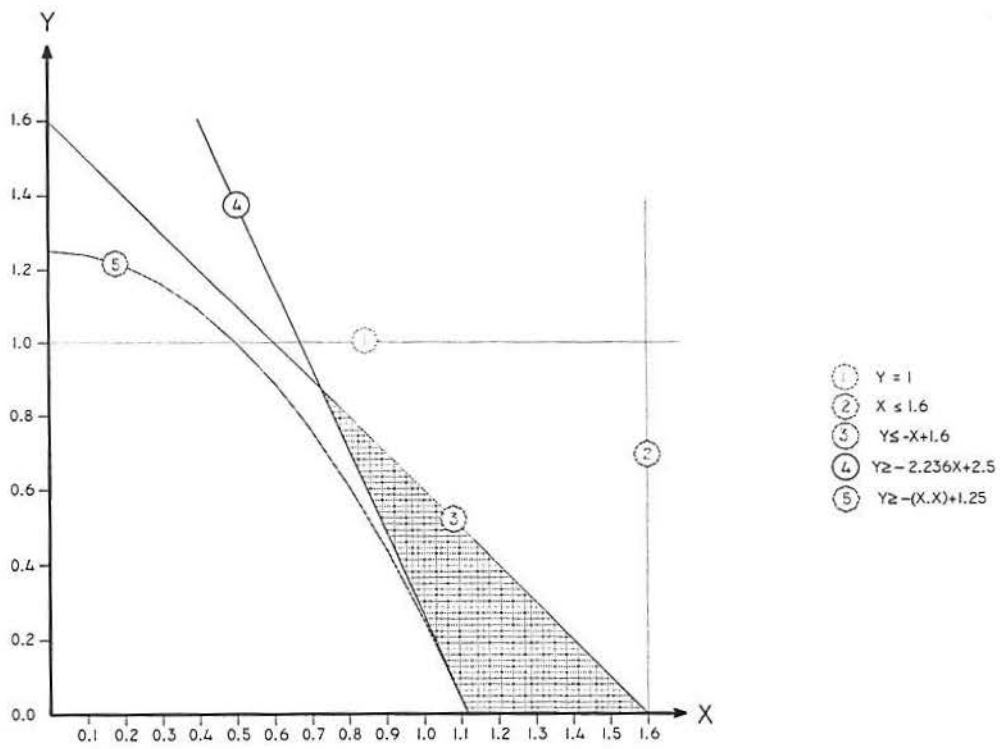


Figura 4.5: Linearização

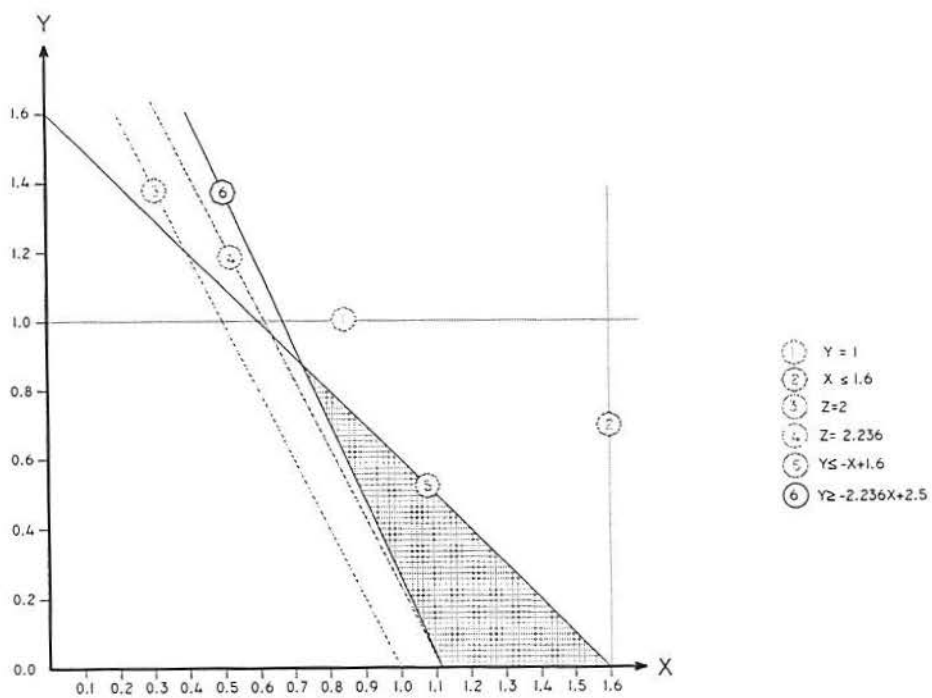


Figura 4.6: Região viável reduzida

- Exemplo 2

O exemplo do problema MINLP proposto por Viswanathan e Grossmann(1990) é modelado como:

$$z = \min -0.7y + 5(x - 0.5)^2 + 0.8$$

sujeito a

(4.8)

$$-e^{(x-0.2)} + 1.1y + 1 \leq 0$$

$$x - 1.2y - 0.2 \leq 0$$

$$0 \leq x \leq 1 \quad y \in \{0, 1\}$$

Como visto na figura 4.7, este é um problema não convexo, pois a primeira inequação é não convexa. Para $y = 0$, a solução do NLP é $z = 1.25$ para $x = 0.2$, enquanto que para $y = 1$, a solução do NLP é $z = 1.076$ para $x = 0.942$. Consequentemente, a solução ótima global do MINLP ocorre quando $y = 1$.

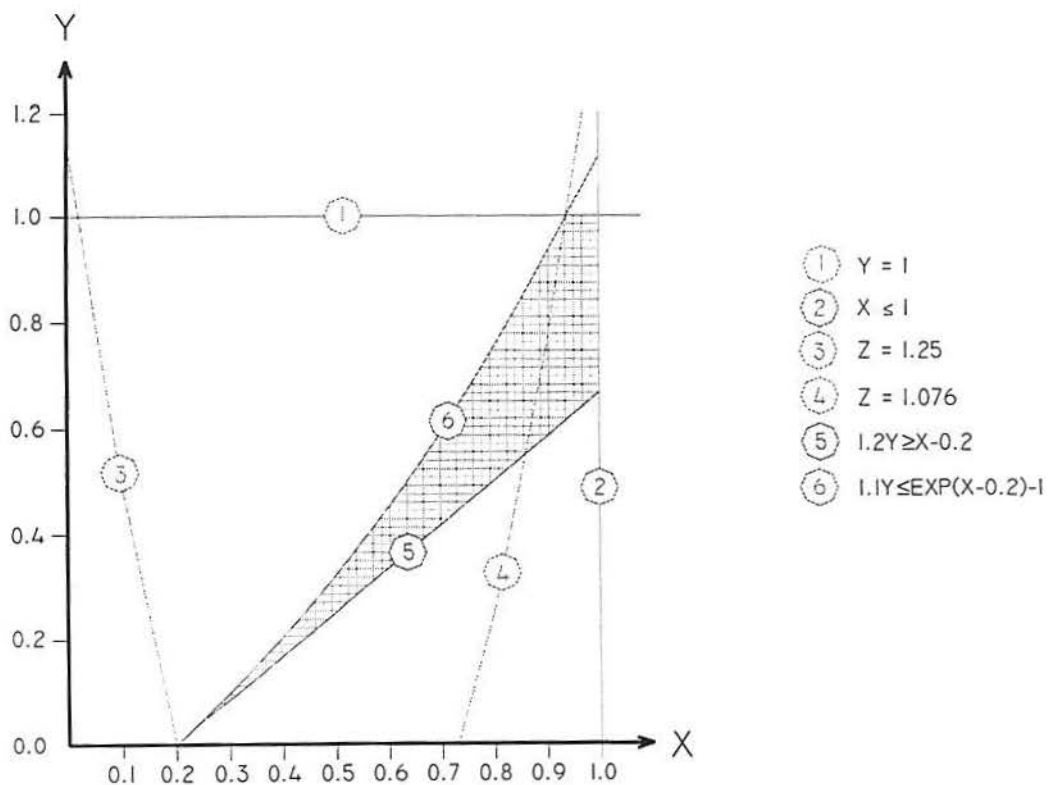


Figura 4.7: Região viável e curvas de níveis da função objetivo

Aplicado o algoritmo proposto neste trabalho e tendo como ponto inicial $y = 0$, então a linearização para o ponto $x = 0.2$ pode ser representada no seguinte problema mestre MILP

$$\begin{aligned}
 & \min \alpha \\
 & \text{sujeito a} \\
 & \alpha \geq -0.7y - 3x + 1.85 \\
 & -x + 1.1y + 0.2 \leq 0 \\
 & x - 1.2y - 0.2 \leq 0 \\
 & 0 \leq x \leq 1 \quad y \in \{0, 1\}
 \end{aligned} \tag{4.9}$$

Este MILP não tem solução viável quando $y = 1$, pois a segunda inequação em (4.9) produz $x \geq 1.3$. Então a solução global ótima se encontra fora da região viável, o que pode ser confirmado através da figura 4.8, mostrando a redução desta região, devido a linearização.

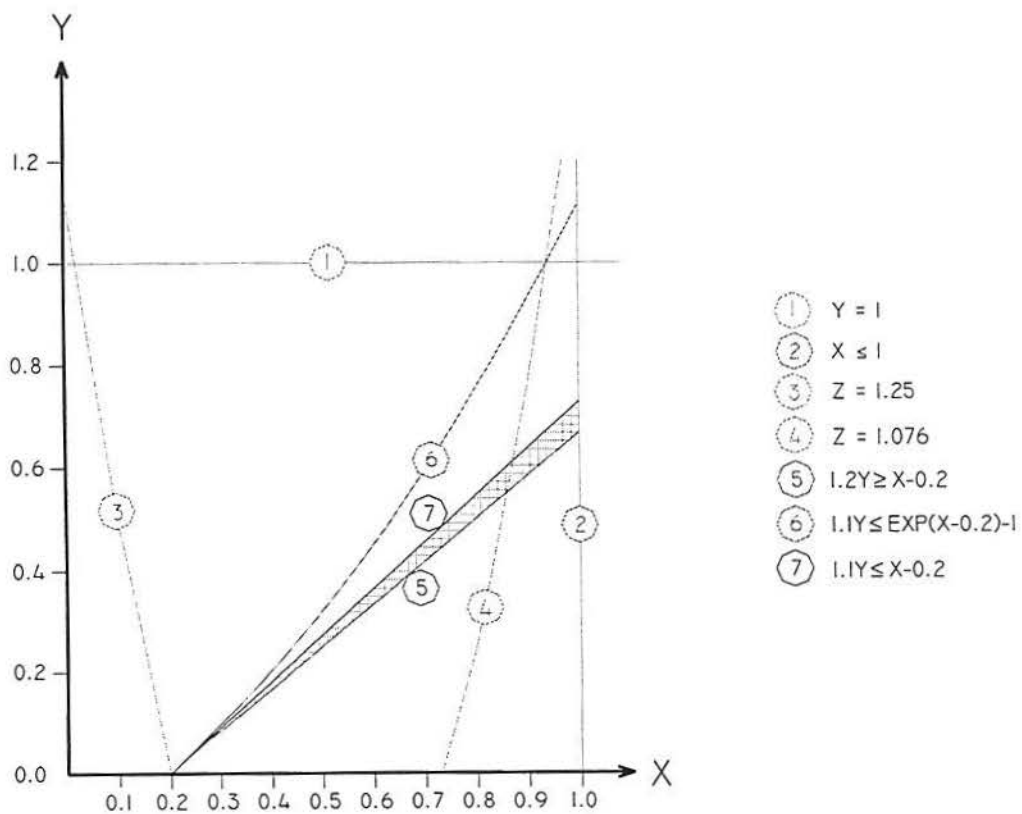


Figura 4.8: Região viável reduzida

- Exemplo 3

A figura 4.9 representa uma superestrutura para um problema de seleção entre dois reatores candidatos a minimizar o custo da produção de um produto desejado, proposto no trabalho de Kocis e Grossmann(1989).

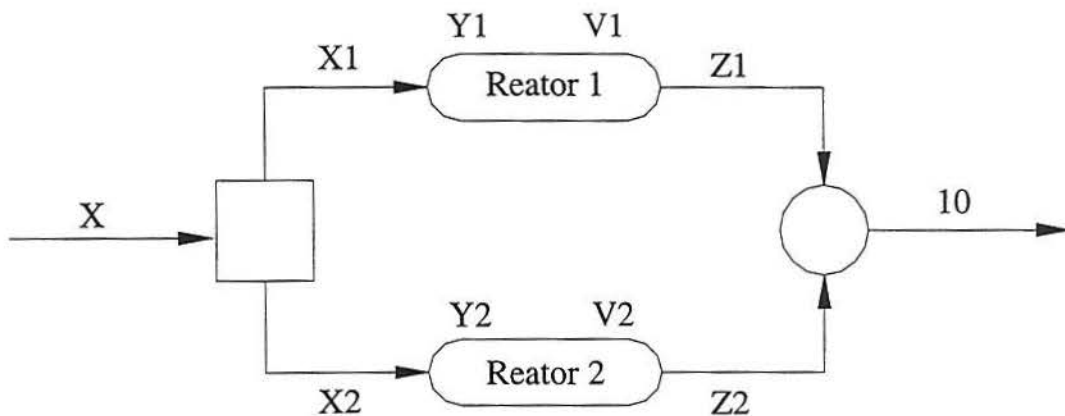


Figura 4.9: Superestrutura

A formulação do MINLP deste problema é dado como:

$$\text{custo} = \min 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5x$$

sujeito a

(4.10)

$$z_1 = 0.9[1 - \exp(-0.5v_1)]x_1$$

$$z_2 = 0.8[1 - \exp(-0.4v_2)]x_2$$

$$x + x_2 - x = 0$$

$$z + z_2 = 10$$

$$v \leq 10y$$

$$v_2 \leq 10y_2$$

$$x \leq 20y$$

$$x_2 \leq 20y_2$$

$$y + y_2 = 1$$

$$x_1, x_2, z_1, z_2, v_1, v_2 \geq 0 \quad y_1, y_2 \in \{0, 1\}^2$$

As variáveis binárias y_1 e y_2 significam a existência ou não dos reatores “1” e “2”, respectivamente. Na função objetivo, os valores 7.5 e 5.5 representam a capacidade dos reatores “1” e “2”, respectivamente; v_1 representa o volume do reator 1, v_2 , o volume do reator 2 e x é o preço da matéria-prima. As duas equações não lineares são relações de entrada e saída para os reatores que definem o fluxo de saída z_1 e z_2 em termos do fluxo de entrada x_1 e x_2 e os volumes dos reatores. A matéria-prima x é dividida entre os fluxos de entrada x_1 e x_2 dos reatores; uma demanda total de 10 unidades necessitam ser encontradas para os fluxos de saída z_1 e z_2 . As próximas quatro inequações são restrições lógicas que asseguram que se um dos reatores não existe (por exemplo $y_1 = 0$), então o volume correspondente e o fluxo da alimentação são nulos. A última restrição requer que apenas um dos reatores “1” e “2” sejam selecionados.

Para efeito de implementação, as restrições de igualdade foram transformadas em restrições de desigualdade, então o MINLP é modelado como:

$$\begin{aligned} \text{custo} &= \min 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5(x_1 + x_2) \\ \text{sujeito a} & & (4.11) \\ z_1 &= 0.9[1 - \exp(-0.5v_1)]x_1 \\ z_2 &= 0.8[1 - \exp(-0.4v_2)]x_2 \\ z_1 + z_2 &\geq 10 \\ z_1 + z_2 &\leq 10 \\ v_1 &\leq 10y_1 \\ v_2 &\leq 10y_2 \\ x_1 &\leq 20y_1 \\ x_2 &\leq 20y_2 \\ y_1 + y_2 &\geq 1 \\ y_1 + y_2 &\leq 1 \\ x_1, x_2, z_1, z_2, v_1, v_2 &\geq 0 \quad y_1, y_2 \in \{0, 1\}^2 \end{aligned}$$

UFMG - SISTEMA DE BIBLIOTECAS
BIBLIOTECA SETORIAL DE MATEMÁTICA
SEÇÃO DE ATENDIMENTO AO PÚBLICO
Av. Antônio Carlos 6627 - Pampulha - Belo Horizonte - Minas Gerais - Brasil

Para valores iniciais de $y = (1, 0)$, o algoritmo resolve um subproblema NLP e três problemas LP, encontrando a solução ótima de $\text{custo} = 99.24$ para $x =$

(13.428,0) e $v = (3.514,0)$, ao passo que se for arbitrado valores iniciais de $y = (0,1)$, então o algoritmo resolve um subproblema NLP e um problema LP, encontrando uma solução sub-ótima de $custo = 107.376$, onde $x = (0,15)$ e $v = (0,4.479)$.

Quando o problema MINLP possui restrições e/ou função objetivo não convexas, o algoritmo proposto pode encontrar soluções sub-ótimas, devido as linearizações que reduzem a região viável deixando a solução ótima fora desta região. Logo as condições implementadas no algoritmo não são suficientes para a resolução de problemas não convexas.

4.3 - Aplicações

O desenvolvimento de um modelo MINLP para síntese de rede de trocadores de calor (HENS) é descrito no trabalho de Daichendt e Grossmann(1994), referindo-se a este modelo como SYNHEAT.

A superestrutura SYNHEAT original requer que o número de estágios seja igual ao número máximo de correntes frias e quentes. A superestrutura original para duas correntes de processos frios e dois quentes bem como as utilidades quentes e frias são mostrados na figura 4.10.

A formulação do modelo SYNHEAT MINLP é mostrado a seguir

Função Objetivo:

$$\min c, \quad \text{onde } c = A_1 + A_2 + A_3 + A_4 + A_5$$

$$A_1 = \sum_{i \in HP} CCU \, qcu_i + \sum_{j \in CP} CHU \, qhu_j$$

$$A_2 = \sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} CF_{ij} \, y_{ijk} + \sum_{i \in HP} CF_{i,CU} \, ycu_i + \sum_{j \in CP} CF_{HU,j} \, yhu_j$$

$$A_3 = \sum_{i \in HP} \sum_{j \in CP} \sum_{k \in ST} C_{ij} \left\{ \left(\frac{q_{ijk}}{U_{ij}} \right) \left[0.5(dt_{ijk})(dt_{ijk+1})(dt_{ijk} + dt_{ijk+1}) \right]^{1/3} \right\}^B$$

$$A_4 = \sum_{i \in HP} C_{i,CU} \left\{ \left(\frac{qcu_i}{U_{i,CU}} \right) [0.5(dtcu_i)(TOUT_i - TIN_{CU})(dtcu_i + TOUT_i - TIN_{CU})]^{1/3} \right\}^B$$

$$A_5 = \sum_{j \in CP} C_{HU,j} \left\{ \left(\frac{qhu_j}{U_{HU,j}} \right) [0.5(dthu_j)(TIN_{HU} - TOUT_j)(dthu_j + TIN_{HU} - TOUT_j)]^{1/3} \right\}^B$$

$$\text{onde, } U_{ij} = \left[\frac{1}{h_i} + \frac{1}{h_j} \right]^{-1}; \quad U_{i,CU} = \left[\frac{1}{h_i} + \frac{1}{h_{CU}} \right]^{-1}; \quad U_{HU,j} = \left[\frac{1}{h_{HU}} + \frac{1}{h_j} \right]^{-1}$$

Balanco de energia para cada estágio:

$$(t_{i,k}^h - t_{i,k+1}^h)F_i = \sum_{j \in CP} q_{ijk}, \quad k \in ST, \quad i \in HP$$

$$(t_{j,k}^c - t_{j,k+1}^c)F_j = \sum_{i \in HP} q_{ijk}, \quad k \in ST, \quad j \in CP$$

Taxas de calor das utilidades:

$$(t_{i,K+1}^h - TOUT_i)F_i = qcu_i, \quad i \in HP$$

$$(TOUT_j - t_{j,1}^c)F_j = qhu_j, \quad j \in CP$$

Designações das temperaturas de entrada:

$$TIN_i = t_{i,1}^h, \quad i \in HP$$

$$TIN_j = t_{j,K+1}^c, \quad j \in CP$$

Decréscimo na temperatura com o aumento de k:

$$t_{i,k}^h \geq t_{i,k+1}^h, \quad k \in ST, \quad i \in HP$$

$$t_{j,k}^c \geq t_{j,k+1}^c, \quad k \in ST, \quad j \in CP$$

Limitações das temperaturas de saída:

$$TOUT_i \leq t_{i,K+1}^h, \quad i \in HP$$

$$TOUT_j \geq t_{j,1}^c, \quad j \in CP$$

Possibilidade de transferência de calor somente se unidade for selecionada:

$$q_{ijk} \leq \Omega y_{ijk}, \quad i \in HP, j \in CP, k \in ST$$

$$qcu_i \leq \Omega ycu_i, \quad i \in HP$$

$$qhu_j \leq \Omega yhu_j, \quad j \in CP$$

$$y_{ijk}, ycu_i, yhu_j = 0, 1$$

Cálculo das temperaturas de aproximação das correntes se a unidade for selecionada:

$$dt_{ijk} \leq t_{i,k}^h - t_{j,k}^c + \Gamma(1 - y_{ijk}), \quad i \in HP, j \in CP, k \in ST$$

$$dt_{ijk+1} \leq t_{i,k+1}^h - t_{j,k+1}^c + \Gamma(1 - y_{ijk}), \quad i \in HP, j \in CP, k \in ST$$

Cálculo das temperaturas de aproximação das utilidades se a unidade for selecionada:

$$dtcu_i \leq t_{i,K+1} - TOUT_{CU} + \Gamma(1 - ycu_i), \quad i \in HP$$

$$dthu_j \leq TOUT_{HU} - t_{j,1}^c + \Gamma(1 - yhu_j), \quad j \in CP$$

Forçar EMAT:

$$dt_{ijk} \geq EMAT$$

Índices:

i = processo quente ou corrente de utilidade quente.

j = processo frio ou corrente de utilidade fria.

k = índice para estágios $1, 2, \dots, K$ e localização da temperatura $1, 2, \dots,$

$K+1$.

Conjuntos:

$HP = \{ i / i \text{ é uma corrente de processo quente} \}$

$CP = \{ j / j \text{ é uma corrente de processo frio} \}$

HU = utilidades quentes.

CU = utilidades frias.

$ST = \{ k / k \text{ é um estágio da superestrutura } k = 1, 2, \dots, K \}$

Parâmetros:

TIN = temperatura de entrada da corrente.

$TOUT$ = temperatura de saída da corrente.

F = capacidade de calor da corrente.

h = coeficiente de transferência de calor individual da corrente.

U = coeficiente de transferência de calor total.

CCU = custo por unidade de utilidade fria.

CHU = custo por unidade de utilidade quente.

CF = custo fixo para os trocadores de calor.

C = coeficiente de custo da área.

B = expoente para o custo da área.

K = número total de estágios.

Ω = limite superior para troca de calor.

Γ = limite superior para diferenças de temperatura.

$EMAT$ = temperatura de aproximação mínima de troca.

Variáveis contínuas, positivas:

c = custo anual da rede de trocadores de calor.

dt_{ijk} = temperatura de aproximação para (i, j) na localização k .

$dtcu_i$ = temperatura de aproximação para (i, CU) .

$dthu_j$ = temperatura de aproximação para (HU, j) .

q_{ijk} = troca de calor para (i, j) no estágio k .

qcu_i = troca de calor para (i, CU) .

qhu_j = troca de calor para (HU, j) .

$t_{i,k}^h$ = temperatura da corrente quente “ i ” no fim do estágio k .

$t_{j,k}^c$ = temperatura da corrente fria “ j ” no fim do estágio k .

Variáveis binárias:

y_{ijk} = existência de unidade entre as correntes (i, j) no estágio k .

ycu_i = existência de unidade entre as correntes (i, CU) .

yhu_j = existência de unidade entre as correntes (HU, j) .

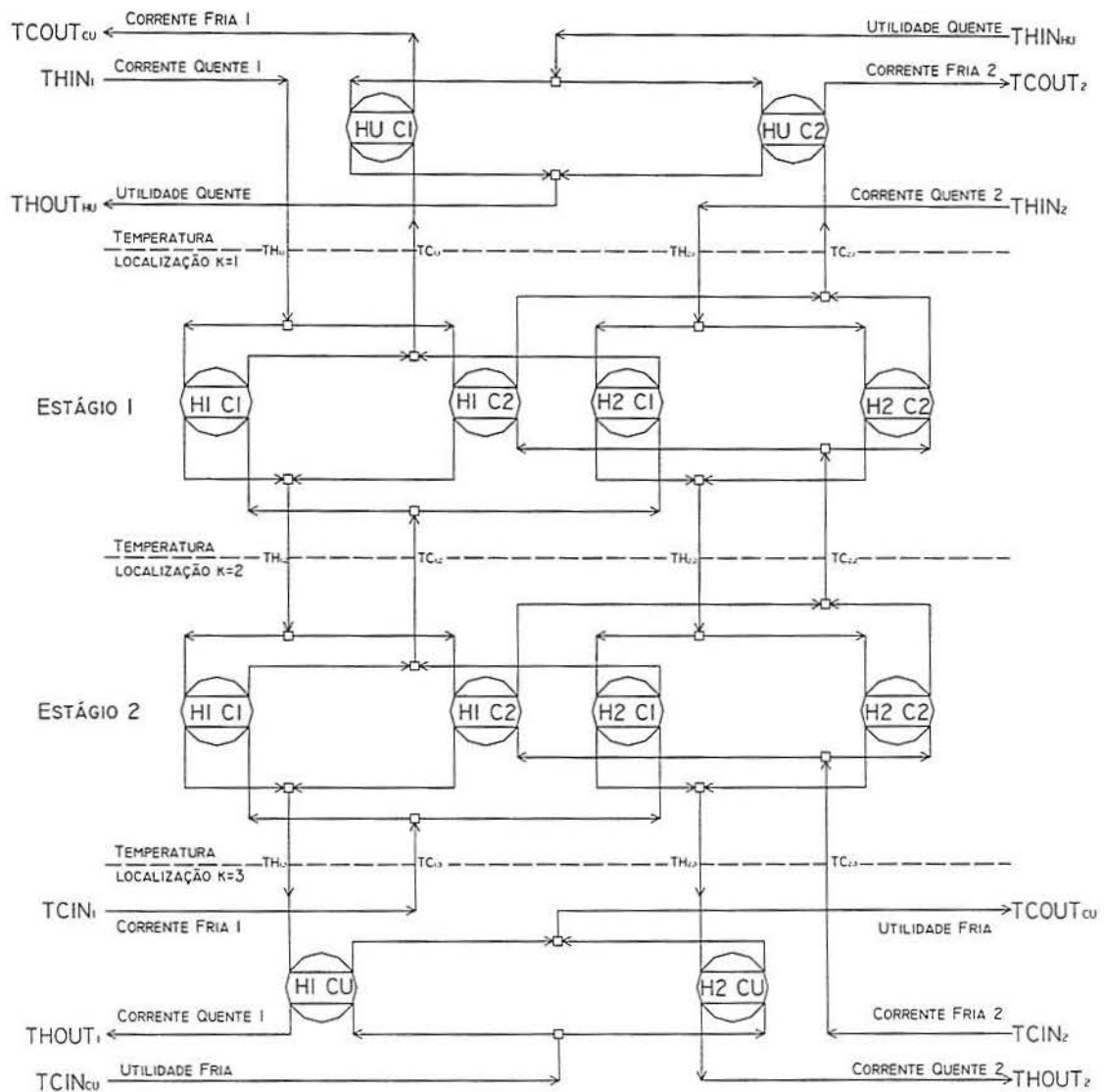


Figura 4.10: Superestrutura SYNHEAT original para problema com duas correntes quentes e duas frias

A figura 4.11 mostra uma superestrutura reduzida de alternativas. A redução foi realizada através de um procedimento de varredura preliminar desenvolvido no trabalho de Daichendt e Grossmann(1994). Este procedimento fez com que as variáveis binárias do problema fossem reduzidas de 135 para apenas 35.

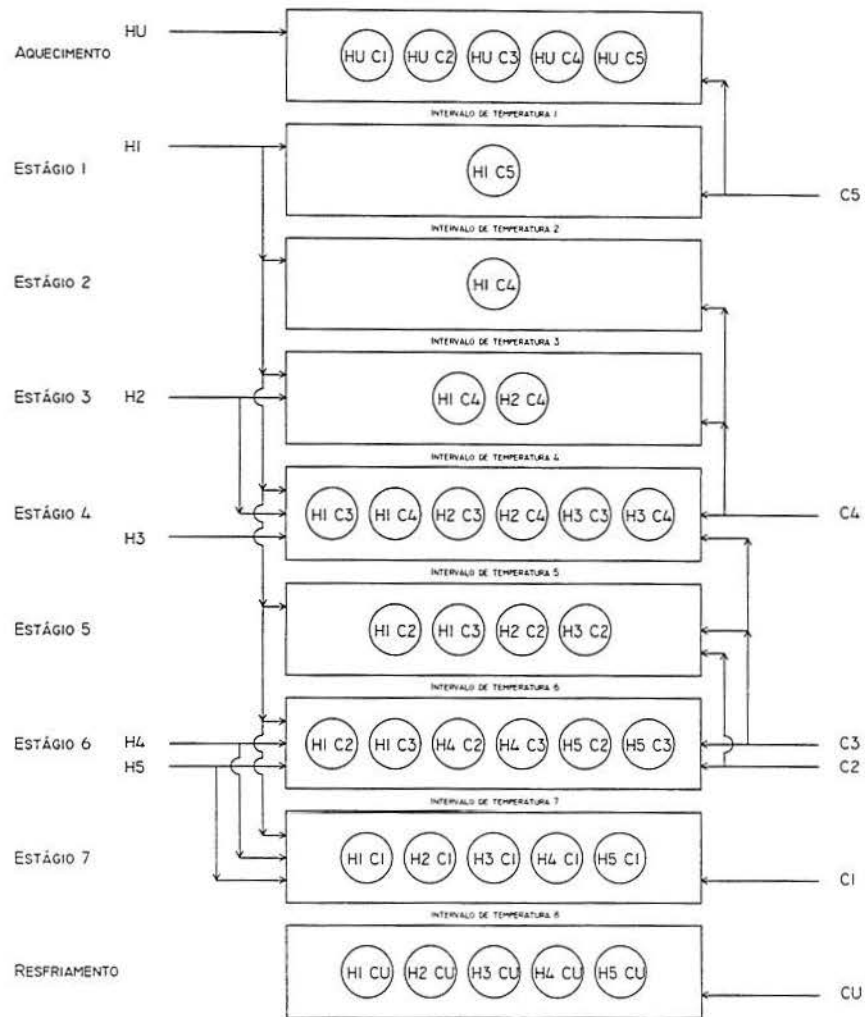


Figura 4.11: Superestrutura reduzida para cinco correntes frias e cinco correntes quentes

O seguinte exemplo baseia-se no procedimento de varredura preliminar. A superestrutura modificada é mostrada na figura 4.12 e a tabela 4.4 apresenta os dados para a implementação. A função objetivo e as restrições foram desenvolvidas conforme o modelo SYNHEAT descrito. Foi adotado $h = 1.6 \text{ kW/m}^2\text{K}$ para todas as correntes e custo de troca ($\$/\text{ano}$) = $1000 + 560(\text{área})^{0.6}$.

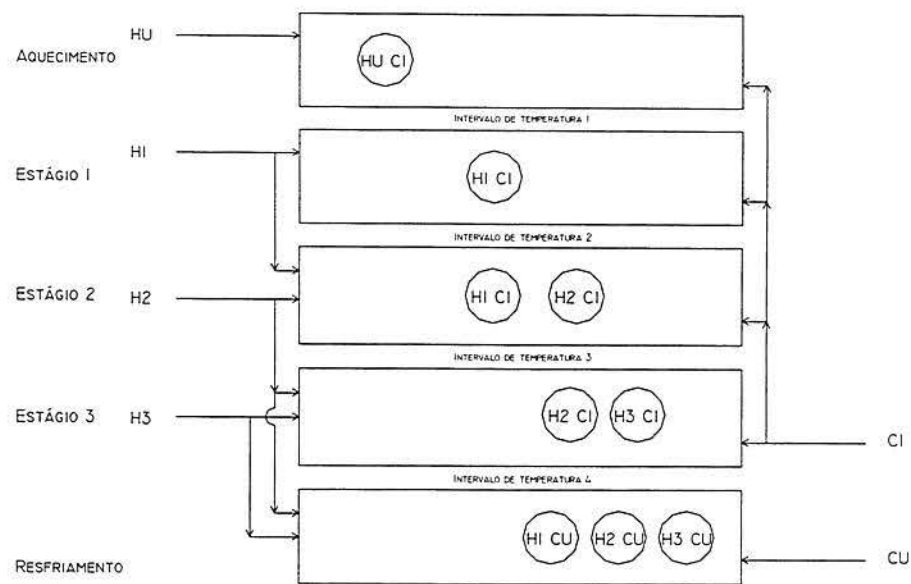


Figura 4.12: Superestrutura reduzida para três correntes quentes e uma corrente fria

<i>corrente</i>	T_{in} (K)	T_{out} (K)	$F \cdot C_p$ (kW/K)	<i>Custo</i> (\$/kW-ano)
H1	500	420	6	-
H2	430	330	6	-
H3	420	340	7	-
C1	340	500	10	-
HU	600	600	-	80
CU	300	320	-	20

Tabela 4.4: Dados para três correntes quentes e uma corrente fria

Este exemplo não possui convexidade, devido a presença de vários mínimos locais, ou seja o subproblema NLP pode exibir mais do que uma solução local. A seguir apresenta-se resultados obtidos pelo algoritmo proposto, ao ser arbitrado valores iniciais para variáveis binárias y_0 . A tabela 4.5 possui estes valores para três testes que foram implementados.

<i>Variáveis binárias</i>	<i>Teste 1</i>	<i>Teste 2</i>	<i>Teste 3</i>
y_{111}	0	1	1
y_{112}	1	1	1
y_{212}	0	0	1
y_{213}	1	1	1
y_{313}	1	1	1
y_{hu_1}	1	1	0
y_{cu_1}	1	1	1
y_{cu_2}	1	1	1
y_{cu_3}	1	1	1

Tabela 4.5: Valores arbitrários para variáveis binárias

O algoritmo mostrou-se eficiente ao resolver o exemplo aplicado a rede de trocadores de calor, onde os resultados obtidos para o custo anual da rede de trocadores de calor, comportamento em relação ao número de NLPs e LPs, temperaturas e variáveis binárias encontram-se, respectivamente, nas tabelas 4.6, 4.7, 4.8 e 4.9.

<i>custo anual</i>	<i>Teste 1</i>	<i>Teste 2</i>	<i>Teste 3</i>
c	6.26	6.37	6.15

Tabela 4.6: Resultados para custo anual

<i>NLPs</i>	<i>Número de Teste 1</i>	<i>de Teste 2</i>	<i>LPs Teste 3</i>
1^0	10	10	12
2^0	3	4	1
3^0	6	5	3
4^0	-	-	16

Tabela 4.7: Número de NLPs e LPs

<i>Temperaturas</i>	<i>Teste 1</i>	<i>Teste 2</i>	<i>Teste 3</i>
dt ₁₁₁	9.26	12.36	35.20
dt ₁₁₂	35.19	35.17	3.20
dt ₂₁₂	9.33	10.02	1.00
dt ₁₁₃	3.19	3.17	0.001
dt ₂₁₃	13.19	13.17	13.20
dt ₃₁₃	3.19	3.17	3.20
dt ₂₁₄	20.76	13.89	20.76
dt ₃₁₄	29.63	35.48	29.63
dthu ₁	135.19	135.17	135.20
dteu ₁	17.62	16.94	300.00
dteu ₂	40.76	33.89	40.76
dteu ₃	49.63	55.48	49.63
th ₁₂	500.00	500.00	420.00
th ₁₃	420.00	420.00	420.00
th ₁₄	420.00	420.00	420.00
th ₂₂	430.00	430.00	430.00
th ₂₃	430.00	430.00	430.00
th ₂₄	360.76	353.89	360.76
th ₃₃	420.00	420.00	420.00
th ₃₄	369.63	375.47	369.63

Tabela 4.8: Resultados para temperaturas

<i>Variáveis binárias</i>	<i>Teste 1</i>	<i>Teste 2</i>	<i>Teste 3</i>
y ₁₁₁	0	1	1
y ₁₁₂	1	1	0
y ₂₁₂	0	0	0
y ₂₁₃	1	1	1
y ₃₁₃	1	1	1
y _{hu1}	1	1	1
y _{cu1}	1	1	0
y _{cu2}	1	1	1
y _{cu3}	1	1	1

Tabela 4.9: Resultados para variáveis binárias

5 - CONCLUSÕES E SUGESTÕES

5.1 - Conclusões

Este trabalho apresentou um método “*branch*” e “*bound*” para problemas MINLP convexos que é baseado na solução de problemas LPs e subproblemas NLPs. Como foi visto, este método evita a solução seqüencial de subproblemas NLPs e problemas mestres MILP que é exigido na implementação dos algoritmos GBD e AO padrão. Os resultados dos problemas testes convexos mostraram que o algoritmo é eficiente, reduzindo o número de nós que necessitam ser examinados para o problema mestre MILP quando comparado com GBD padrão e, no mínimo, obtendo resultados iguais quando comparado com GBD modificado.

Existem duas razões possíveis onde a presença de não convexidade pode causar dificuldades. Primeiro, o subproblema NLP pode exibir mais do que uma solução local. Segundo, mesmo se o subproblema não convexo tem solução única, o

problema mestre MILP pode não fornecer limites inferiores rigorosos, e como consequência, pode cortar a solução ótima da região viável.

Analisando os resultados para problemas testes não convexos, o algoritmo é eficiente, mas não o suficiente para encontrar soluções globais ótimas, ou seja há necessidade de novas condições a serem levadas em conta na implementação para que soluções ótimas não se encontrem fora da região viável e não haja convergência para soluções sub-ótimas.

Também fez-se a análise do comportamento do algoritmo frente a um problema aplicado a rede de trocadores de calor, obtendo resultados satisfatórios. O algoritmo é sensível à condição inicial, isto é, dependendo da condição inicial arbitrada para variáveis binárias, o algoritmo encontrou soluções diferentes. A explicação está no fato do problema ser não convexo devido a presença de mínimos locais. Este problema pode ser resolvido com o uso de um esquema de inicialização aleatória sobre o espaço viável.

5.2 - Sugestões de Trabalhos

O estudo da não convexidade para problemas MINLP, tratando da segunda dificuldade como mencionada anteriormente, é sugerido como continuação deste trabalho, implementando condições adicionais, sugerido por Kocis e Grossmann(1988).

A sugestão neste trabalho seria resolver problemas MINLP não convexos em duas fases, aproveitando o algoritmo desenvolvido. Na fase I, o algoritmo proposto seria aplicado em sua forma original, porém com previsão para a identificação de funções não convexas que podem cortar soluções ótimas. Se estas funções não são detectadas, o algoritmo continua, mas com o acréscimo dos testes de convexidade, à medida que as linearizações são feitas. Caso contrário, procede-se para a fase II, onde as linearizações são sistematicamente modificadas em um novo problema mestre com a introdução de testes de convexidade à medida que os MILPs e NLPs são definidos. A busca termina quando melhorias não são encontradas nos subproblemas NLPs. Os

passos principais na fase II para lidar com problemas MINLP não convexos são mostrados na figura 5.1.

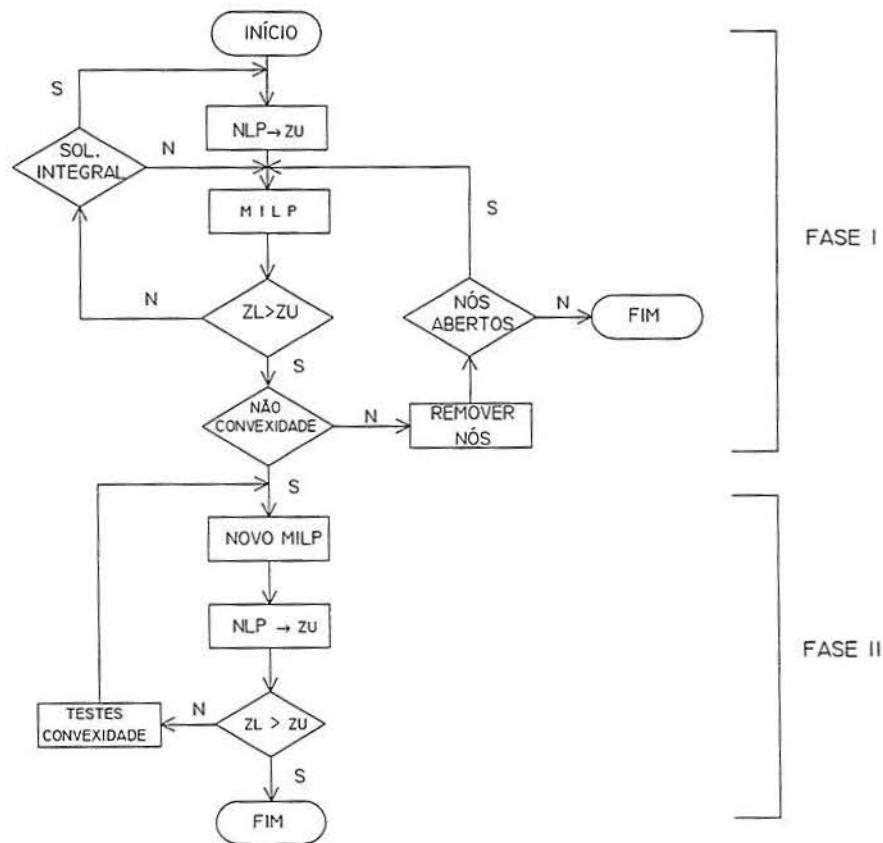


Figura 5.1: Passos na fase II

As condições adicionais para identificação de funções não convexas são tratadas no trabalho de Kocis e Grossmann(1988), através de testes locais e globais. A seguir este assunto será tratado de forma detalhada, usando como referência o MINLP modelado como:

$$z = \min_{x,y} c^T y + f(x)$$

sujeito a

(5.1)

$$h(x) = 0$$

$$By + Cx \leq d$$

$$x \in X = \{x / x \in R^n, x^l \leq x \leq x^u\}$$

$$y \in Y = \{y / y \in \{0,1\}^m\}$$

(5.1) Seja o problema mestre MILP na iteração “K”, associado ao MINLP

$$z_i^K = \min_{x,y,\mu} c^T y + \mu$$

sujeito a (5.2)

$$\left. \begin{aligned} \nabla f(x^k)^T x - \mu &\leq \nabla f(x^k)^T (x^k) - f(x^k) \\ T^k \nabla h(x^k)^T x &\leq T^k \nabla h(x^k)^T (x^k) \\ \nabla g(x^k)^T x &\leq \nabla g(x^k)^T (x^k) - g(x^k) \end{aligned} \right\} \quad k=1,2,\dots,K$$

$$By + Cx \leq d$$

$$x \in X \quad y \in \{0,1\}^m \quad \mu \in \mathbb{R}$$

onde z_i^K é o limite inferior produzido para a iteração “K”, T^k é uma matriz $r \times r$ para o relaxamento das aproximações lineares das “r” equações não lineares por “r” inequações. Os elementos da diagonal da matriz T^k são dados por :

$$t_{ii}^k = \begin{cases} -1 & \text{se } \lambda_i^k < 0 \\ 1 & \text{se } \lambda_i^k > 0 \\ 0 & \text{se } \lambda_i^k = 0 \end{cases} \quad I = 1, 2, \dots, r$$

onde λ_i^k estão associados aos multiplicadores de Lagrange ótimos para as equações não lineares $h_i(x) = 0, \quad i = 1, 2, \dots, r$ no subproblema NLP^k .

5.2.1 - Teste Local

O objetivo do teste local é obter informação para analisar condições de convexidade local das restrições não lineares devido suas linearizações. Primeiro considere as equações não lineares $h(x) = 0$ de (5.1) e seja h_i^{LIN} a linearização da equação h_i em torno da solução ótima do subproblema NLP^k, x^k .

$$h_i^{LIN}(x) = h_i(x^k) + \nabla h_i(x^k)^T (x - x^k)$$

Como o ponto x^k necessita satisfazer $h_i(x^k) = 0$, então

$$h_i^{LIN}(x) = \nabla h_i(x^k) (x - x^k)$$

As condições de quase-convexidade para $h_i(x)$ requer que a seguinte relação seja satisfeita se $h(x) \leq 0$:

$$\nabla h(x^k)^T (x - x^k) \leq 0 \quad (5.3)$$

Logo para a função $h_i(x)$ ser quase-convexa, a condição (5.3) necessita manter-se para todo x . No teste local, tenta-se encontrar um ponto $(\bar{x})^k$ próximo de x_k , o qual viola esta condição. Portanto, existem um número infinito de pontos candidatos à considerar. Uma escolha lógica é um ponto $(\bar{x})^k$, o qual produz um decréscimo na função objetivo. Tal ponto pode ser encontrado resolvendo um subproblema NLP relaxado, o LNLP_k, que é idêntico ao NLP_k, exceto que os limites inferiores e superiores de x_i^k e y_i^k não nulos são dados por :

$$\begin{aligned} x_i^k(1 - \varepsilon) &\leq x_i \leq x_i^k(1 + \varepsilon) \\ y_i^k(1 - \varepsilon) &\leq y_i \leq y_i^k(1 + \varepsilon) \end{aligned}$$

e x_i^k e y_i^k nulos dados por:

$$-\varepsilon \leq x \leq \varepsilon$$

$$-\varepsilon \leq y_i \leq \varepsilon$$

onde ε é um número pequeno (por exemplo 0.05).

Estes limites asseguram que o ponto solução (\bar{x}) permanecerá próximo de x^k . É claro que, para a solução do LNLP^k, $h_i((\bar{x})^k) = 0$. Referindo-se a condição de quase-convexidade (5.3), o teste local reduz-se a verificar se $\nabla h_i(x^k) \left((\bar{x})^k - x^k \right) \leq 0$, o qual é equivalente testar a viabilidade da linearização para (\bar{x}) . Em termos das linearizações do problema mestre (5.2), o teste local de quase-convexidade nas

equações não lineares $h(x) = 0$ e inequações ativas $g_A(x) = 0$ reduz-se então para verificar se :

$$T^k \nabla h(x^k)^T (\bar{x})^k \leq T^k \nabla h(x^k)^T (x^k) \tag{5.4}$$

$$\nabla g_A(x^k)^T (\bar{x})^k \leq \nabla g_A(x^k)^T (x^k) - g_A(x^k)$$

As desigualdades violadas em (5.4) correspondem às linearizações que falharam o teste local.

Agora, considerando as restrições de desigualdade (inequações inativas) $g_I(x) < 0$ e o termo da função objetivo não linear $f(x)$. Para garantir convergência para a solução ótima, as restrições de desigualdade inativas e a função objetivo precisam ser convexas, ou seja as seguintes condições devem ser satisfeitas:

$$g(x) \geq g(x^k) + \nabla g(x^k)^T (x - x^k) \tag{5.5}$$

O lado direito da condição (5.5) é a linearização de $g_I(x)$ para o ponto x^k , então (5.5) pode ser escrito como:

$$g_I(x) \geq g_I^{LIN}(x) \tag{5.6}$$

O teste local para as linearizações no problema mestre (5.2) de $g_I(x) < 0$ e $f(x)$ reduz-se a partir de (5.6) em verificar as seguintes desigualdades para os pontos $(\bar{x})^k$.

$$g_I((\bar{x})^k) \geq \nabla g_I(x^k)^T (\bar{x})^k - [\nabla g_I(x^k)^T (x^k) - g_I(x^k)] \tag{5.7}$$

$$f((\bar{x})^k) \geq \nabla f(x^k)^T (\bar{x})^k - [\nabla f(x^k)^T (x^k) - f(x^k)]$$

5.2.2 - Teste Global

O teste local objetiva verificar as condições de quase-convexidade e convexidade próximo o ponto para o qual a aproximação linear é derivada. Entretanto, este procedimento não determina se a linearização prevê uma aproximação externa

válida para outros pontos no espaço viável do MINLP. O teste global prevê esta informação e é similar ao teste local, exceto que eles são aplicados para as linearizações da iteração “ l ” para os pontos x^k , $k = 1, 2, \dots, K$, $k \neq l$. Assim, a partir de (5.4), as condições que precisam ser verificadas para as equações e inequações ativas são:

$$T^l \nabla h(x^l)^T(x^k) \leq T^l \nabla h(x^l)^T(x^l) \tag{5.8}$$

$$\nabla g_A(x^l)^T(x^k) \leq \nabla g_A(x^l)^T(x^l) - g_A(x^l) \quad l, k = 1, 2, \dots, K \quad l \neq k$$

e a partir de (5.7), para desigualdades não lineares inativas e a função objetivo, as condições a serem testadas são:

$$g_l(x^k) \geq \nabla g_l(x^l)^T(x^k) - [\nabla g(x^l)^T(x^l) - g(x^l)] \tag{5.9}$$

$$f(x^k) \geq \nabla f(x^l)^T(x^k) - [\nabla f(x^l)^T(x^l) - f(x^l)] \quad l, k = 1, 2, \dots, K \quad l \neq k$$

Finalmente, as linearizações encontradas que falham as condições (5.8) e (5.9) do teste global são modificadas para satisfazer todos os pontos x^k . Isto pode ser feito determinando para cada linearização a violação máxima para o ponto x^m (a partir dos pontos x^k , $k = 1, 2, \dots, K$), ou seja:

$$r^l = \nabla h(x^l)^T x^m$$

$$s_A^l = \nabla g_A(x^l)^T x^m \tag{5.10}$$

$$s^l = \nabla g(x^l)^T x^m - g(x^k)$$

$$w_0^l = \nabla f(x^l)^T x^m - f(x^m)$$

onde r^l, s_A^l, s^l, w_0^l são os novos valores dos coeficientes que determinam aproximações externas válidas para os pontos x^k , $k = 1, 2, \dots, K$.

5.2.3 - Problema Mestre na Fase II

A formulação do problema mestre para a fase II é similar ao problema mestre (5.2), exceto para a adição de um limite superior para a função objetivo original e o tratamento das linearizações que falharam para os testes local e/ou global.

Se as condições para o teste local (5.4) e (5.7) falharam, uma variável de folga não negativa ξ_i^k é introduzida para cada linearização “ i ” da iteração “ k ”, que falhou no teste local. Assim, as linearizações modificadas no problema mestre são dadas por:

$$\begin{aligned} \nabla f_i(x^k)^T x - \mu - \left| \nabla f_i(x^k)^T(x^k) - f_i(x^k) \right| \xi_i^k &\leq \nabla f_i(x^k)^T(x^k) - f_i(x^k) \\ T^k \nabla h_i(x^k)^T x - \left| T^k \nabla h_i(x^k)^T(x^k) \right| \xi_i^k &\leq T^k \nabla h_i(x^k)^T(x^k) \\ \nabla g_i(x^k)^T x - \left| \nabla g_i(x^k)^T(x^k) - g_i(x^k) \right| \xi_i^k &\leq \nabla g_i(x^k)^T(x^k) - g_i(x^k) \\ k &= 1, 2, \dots, K \end{aligned} \quad (5.11)$$

Substituindo (5.10) em (5.8) e (5.9) para as linearizações que não satisfazem estas inequações. E introduzindo as variáveis de folga ξ_i^k em cada linearização “ i ” da iteração “ k ”, como em (5.11), tem-se as linearizações modificadas no problema mestre, onde os testes globais falharam, ou seja:

$$\begin{aligned} \nabla f_i(x^k)^T x - \mu - \left| w_{io}^k \right| \xi_i^k &\leq w_{io}^k \\ T^k \nabla h_i(x^k)^T x - \left| T^k r_i^k \right| \xi_i^k &\leq T^k r_i^k \\ \nabla g_i(x^k)^T x - \left| s_i^k \right| \xi_i^k &\leq s_i^k \\ k &= 1, 2, \dots, K \end{aligned} \quad (5.12)$$

A partir de (5.11) e (5.12) tem-se o novo problema mestre MILP definido para a fase II. Maiores detalhes à respeito dos testes locais e globais podem ser encontrados no trabalho de Kocis e Grossmann (1988).

Outra sugestão de trabalho é o desenvolvimento de algoritmos MINLP paralelos, isto é, algoritmos que explorem a potencialidade de processamento concorrente deste tipo de problema .

REFERÊNCIAS BIBLIOGRÁFICAS

- BENDERS, J.F. Partitioning procedures for solving mixed-variables programming problems, *Numer. Math.*, **4**, 238-252, 1962.
- BROYDEN, C. G. The convergence of a class of double-rank minimization algorithms 2: the new algorithm, *Journal Institute of Math and its Appl.*, **6**, 222-231, 1970.
- COOPER, L. e STEINBERG, D. *Introduction to Methods of Optimization*, W. B. Saunders Company, Philadelphia, 1970.
- DAICHENDT, M. M. e GROSSMANN, I. E. Preliminary screening procedure for the MINLP synthesis of process systems-II. Heat exchanger networks, *Computers Chem. Engng.*, **8**, 679-709, 1994.

- DANTZIG, G. B. *Maximization of a linear function of variables subject to linear inequalities*. Activity analysis of production and allocation, editores: Tj. C. Koopmans, 339-347, Wiley, New York, 1951.
- DURAN, M. A. e GROSSMANN, I. E. An outer approximation algorithm for a class of mixed-integer nonlinear programs, *Math. Progr.*, **36**, 307-339, 1986a.
- DURAN, M. A. e GROSSMANN, I. E. A mixed-integer nonlinear programming algorithm for process systems synthesis, *AIChE Journal*, **32**(4), 592-606, 1986b.
- FLETCHER, R. A new approach to variable metric algorithm, *The Computer Journal*, **13**(3), 317-322, 1970.
- FLETCHER, R. e LEYFFER, S. Solving mixed integer nonlinear programs by outer approximation, *Math. Progr.*, **66**(3), 327-349, 1994.
- FLOUDAS, C. A. *Nonlinear and Mixed-Integer Optimization*, Oxford University, New York, 1995.
- FLOUDAS, C. A. e VISWESWARAN, V. A global optimization algorithm (GOP) for certain classes of nonconvex NLPs-I. Theory. *Computers Chem. Engng.*, **14**(12), 1397-1417, 1990a.
- FLOUDAS, C. A. e VISWESWARAN, V. A global optimization algorithm (GOP) for certain classes of nonconvex NLPs-II. Application of theory and test problems, *Computers Chem. Engng.*, **14**(12), 1419-1434, 1990b.
- GEOFFRION, A. M. Generalized Benders decomposition, *Journal Optimization Theory Aplic.*, **10**(4), 237-260, 1972.

- GOLDFARB, D. A family of variable metric methods derived by variational means, *Mathematics of Computation*, **24**, 23-26, 1970.
- GROSSMANN, I. E. e KRAVANJA, Z. Mixed-integer nonlinear programming techniques for process systems engineering, *Computers Chem. Engng.*, **19**, 189-204, 1995.
- HIMMELBLAU, D.M. *Applied Nonlinear Programming*, McGraw-Hill, New York, 1972.
- HOLMBERG, K. On the convergence of the cross decomposition, *Math. Progr.*, **47**(2), 269-296, 1990.
- KOCIS, G. R. e GROSSMANN, I. E. Relaxation strategy for the structural optimization of process flowsheets, *Ind. Engng. Chem. Res.*, **26**(9), 1869-1880, 1987.
- KOCIS, G. R. e GROSSMANN, I. E. Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis, *Ind. Engng. Chem. Res.*, **27**, 1407-1421, 1988
- KOCIS, G. R. e GROSSMANN, I. E. A modelling and decomposition strategy for the MINLP optimization of process flowsheets, *Computers Chem. Engng.*, **13**(7), 797-819, 1989.
- KRAVANJA, Z. e GROSSMANN, I. E. New developments and capabilities in PROSYN – an automated topology and parameter process synthesizer, *Computers Chem. Engng.*, **18**(11,12), 1097-1114, 1994.
- KRAVANJA, Z. e GROSSMANN, I. E. Multilevel-hierarchical MINLP synthesis of process flowsheets, *Computers Chem. Engng.*, **21**, S421-S426, 1997.

- LAND, A. H. e DOIG, A. G. An automatic method for solving discrete programming problems, *Econometrica*, **28**, 497,1960.
- MATTHEWS, R. How do you work out the answers to questions that would take a desktop computer 15 milion years to solve?, *New Scientist*, pp 40, 1995.
- MINOUX, M. *Mathematical Programming. Theory and Algoritms*, John Wiley and Sons, França,1986.
- NOVAK, Z., KRAVANJA, Z. e GROSSMANN, I.E. Simultaneous synthesis of distillation sequences in overall process schemes using an improved MINLP approach, *Computers Chem. Engng.*, **20**(12), 1425-1440, 1996.
- PAULES, G. E. e FLOUDAS, C. A. Stochastic programming in process synthesis: a two-stage model with MINLP recourse for multiperiod heat-integrated distillation sequences, *Computers Chem. Engng.*, **16**(3), 189-210, 1992.
- QUESADA, I. e GROSSMANN, I. E. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems, *Computers Chem. Engng.*, **16**(10/11), 937-947, 1992.
- QUESADA, I. e GROSSMANN, I. E. Global optimization algorithm for heat exchanger networks, *Ind. Eng. Chem. Res.*, **32**, 487-499, 1993.
- RAMAN, R. e GROSSMANN, I. E. Relation between MILP modelling and logical inference for chemical process synthesis. *Computers Chem. Engng.*, **15**, 73-84, 1991.
- RAMAN, R. e GROSSMANN, I. E. Integration of logic and heuristic knowledge in MINLP optimization for process synthesis, *Computers Chem Engng.*, **16**(3), 155-171, 1992.

- RAMAN, R. e GROSSMANN, I. E. Symbolic integration of logic in mixed-integer linear programming techniques for process synthesis, *Computers Chem. Engng*, **17**(9), 909-927, 1993.
- RAMAN, R. e GROSSMANN, I. E. Modelling and computational techniques for logic based integer programming, *Computers Chem. Engng.*, **18**(7), 563-578, 1994.
- SECCHI, A. R. *Técnicas de Otimização. Notas de Aula*, Universidade Federal do Grande do Sul, Porto Alegre, 1993.
- SECCHI, A. R. e BISCAIA JR., E. C. Processamento Paralelo: otimização de processos químicos, Anais do XII CNMAC, São José do Rio Preto, São Paulo, 46-48, 1989.
- SECCHI, A. R. e PERLINGEIRO, C. A. Otimização: busca aleatória adaptativa, Anais do XII CNMAC, São José do Rio Preto, São Paulo, 49-52, 1989.
- SHANNO, D. F. Conditioning of quasi-Newton methods for function minimization, *Mathematics of Computation*, **24**, 641-656, 1970.
- SMITH, E. M. B. On the optimal design of continuous process, PhD thesis, University of London, 1996.
- SMITH, E. M. B e PANTELIDES, C. C. Global optimisation of nonconvex MINLPs, *Computers Chem. Engng.*, **21**, S791-S796, 1997.
- TURKAY, M. e GROSSMANN, I. E. Logic-based algorithms for the optimal synthesis of process networks, *Computers Chem. Engng.*, **20**(8), 959-978, 1996.

- VAN ROY, T. J. Cross decomposition for mixed integer programming, *Math. Progr.*, **25**(1), 46-63, 1983.
- VECCHIETTI, A. e GROSSMANN, I. E. LOGMIP: a disjunctive 0-1 nonlinear optimizer for process systems models, *Computers Chem. Engng.*, **21**, S427-S432, 1997.
- VISWANATHAN, J. e GROSSMANN, I. E. A combined penalty function and outer approximation method for MINLP optimization, *Computers Chem. Engng.*, **14**(7), 769-782, 1990.
- VISWANATHAN, J. e GROSSMANN, I. E. Optimal feed locations and number of trays for distillation columns with multiple feeds, *Ind. Eng. Chem. Res.*, **32**, 2942-2949, 1993
- YEE, T. F. e GROSSMANN, I. E. Simultaneous optimization models for heat integration-II. Heat exchanger network synthesis, *Computers Chem. Engng.*, **14**(10), 1165-1184, 1990.
- ZAMORA, J. M. e GROSSMANN, I. E. Global optimization of MINLP problems, INFORMS Spring Meeting, ME29-3, Washington, 1996.
- ZAMORA, J. M. e GROSSMANN, I. E. A comprehensive global optimization approach for the synthesis of heat exchanger networks with no stream splits, *Computers Chem. Engng.*, **21**, S65-S70, 1997.

ANEXOS

A rotina MINLP.m estrutura o algoritmo “branch” e “bound” espacial.

```

function [wo,St,x0,y0] =
minlp(NLP,GRAD,n,m,rx,fig,x0,y0,xlb,xub,M,epsR,epsI,epsC,epsZ)
~
%   * * * Solution of MINLP * * *
~
%   Z = min(y,x) F(x,y) = C'y + f(x)
~
%   s.t. G(x,y) = By + g(x) <= 0
~
%       x in X in R^n
~
%       y in Y = {0,1}^m
%
% usage: [wo,St,x0,y0] =
minlp(NLP,GRAD,n,m,rx,fig,x0,y0,xlb,xub,M,epsR,epsI,epsC,epsZ)
%
% wo   : best solution found [x,y,u,Z]
%       where u is the maximum constraint violation
% St   : matrix = (number of constraints, number of LPs) for each NLP
% NLP  : M file describing the NLP: [F,G] = nlp(x,y)
%       with the first line: if size(x,2) > n, y = x(n+1:n+m); end
% GRAD : M file for gradients: [Fx,Gx,C,B] = grad(x,y)
%       with the first line: if size(x,2) > n, y = x(n+1:n+m); end
%       and last lines: if size(x,2) > n, Fx = [Fx C]; Gx = [Gx B]; end
%       Gx = Gx';
% rx   : = 1 : relaxed initial NLP. = 0 : unrelaxed initial NLP (default)
% fig  : = 1 : plot constraints surfaces. = 0 : no plots (default).
% x0   : initial guess for x (default = zeros(1,n)). Returns first solution.
% y0   : initial guess for y (default = round(rand(1,m)))
% xlb  : lower bound for x (default = zeros(1,n))
% xub  : upper bound for x (default = inf * ones(1,n))
% M > 0 : constraint violation penalization (default = 1e3)
% epsR : real variable tolerance (default = 1e-4)

```

```
% epsI : integer variable tolerance (default = 1e-4)
% epsC : constraint violation tolerance (default = 1e-6)
% epsZ : objective function tolerance (default = 1e-4)
%

% authors: Argimiro R. Secchi and Elaine C. Pereira
% Chemical Engineering Department
% Applied Math Department
% Universidade Federal do Rio Grande do Sul, Brasil
% arge@enq.ufrgs.br and dmtecp@super.furg.br
%

% version 1.0: 15/mar/1999

if nargin < 4
    disp('minimal usage: minlp(NLP,GRAD,n,m)');
    return;
end
if nargin < 5
    rx = 0;
end
if nargin < 6
    fig = 0;
end
if nargin < 7
    x0 = zeros(1,n);
end
if nargin < 8
    y0 = round(rand(1,m));
end
if nargin < 9
    xlb = zeros(1,n);
end
if nargin < 10
    xub = inf * ones(1,n);
end
if nargin < 11
    M = 1e3;
end
if nargin < 12
    epsR = 1e-4;
end
if nargin < 13
    epsI = 1e-4;
end
if nargin < 14
    epsC = 1e-6;
end
if nargin < 15
    epsZ = 1e-4;
end
```

```
if isempty(rx)
    rx = 0;
end
if isempty(fig)
    fig = 0;
end
if isempty(x0)
    x0 = zeros(1,n);
end
if isempty(y0)
    y0 = round(rand(1,m));
end
if isempty(xlb)
    xlb = zeros(1,n);
end
if isempty(xub)
    xub = inf * ones(1,n);
end
if isempty(M) | M <= 0
    M = 1e3;
end
if isempty(epsR)
    epsR = 1e-4;
end
if isempty(epsI)
    epsI = 1e-4;
end
if isempty(epsC)
    epsC = 1e-6;
end
if isempty(epsZ)
    epsZ = 1e-4;
end

y1b = zeros (1,m);
y1u = ones(1,m);

% * * * Solution of initial NLP * * *
%     Z = min(x) C'y0 + f(x) + Mu
%     s.t. By0 + g(x) <= u
%         x in X
%         y0 fixed
%         u in R+

if rx      % NLP with relaxed integrality
    x0=[x0 y0];
    w1b=[x1b y1b];
    w1u=[x1u y1u];
else
```



```

wlb=xlb;
wub=xub;
end

op = foptions;
% op(1) = 1;      % 1 - display some results; 0 - nothing
op(2) = epsR;
op(3) = epsZ;
op(4) = epsC;
op(13) = 0;
% op(14) = 150; % max function evaluation

% constr returns the best solution, even violating the constraints
% then u does not need be computed explicitly.

[x0,op,lambda] = constr(NLP,x0,op,wlb,wub,GRAD,y0);

if rx
    y0 = round(x0(n+1:n+m));
    dym = max(abs(x0(n+1:n+m)-y0));
    x0 = x0(1:n);
    if dym > epsI
        [x0,op,lambda] = constr(NLP,x0,op,xlb,xub,GRAD,y0);
    end
end

% getting the values of f, g, and their gradients
[f_nlp,g_nlp] = eval([NLP '(x0,y0)']);
u = max(g_nlp);      % u > 0 means constraint violation
[gf_nlp,gg_nlp,C,B] = eval([GRAD '(x0,y0)']);
f = f_nlp - C * y0';
g = g_nlp' - B * y0';
gf = gf_nlp;
gg = gg_nlp';

if u > epsC          % Z upper bound
    zu = inf;
else
    zu = f_nlp;
end

r = size(B,1);      % number of constraints
St = [r+1,0];      % matrix of NLPs: number of constraints x LPs

wo = [x0,y0,u,zu];  % best solution at each iteration

A = [gg,B,-ones(r,1),zeros(r,1); gf,C,M,-1];
b = [gg*x0'- g; gf*x0'-f];

wlb = [xlb,ylb,0,-inf];

```

```

wub = [xub,yub,inf,inf];

if fig, planos(n,A,b,y0,u,xlb,xub,1); end

Tr = [0, 1, 0, 0, 1:m]; % binary tree [parent, child, mj, nfix, fix, ~fix]
W = wo; % solution of the LPs
no = 1; % number of nodes in the tree

[flag,wo,A,b,W,Tr,no,St] =
lp_nlp(NLP,GRAD,n,m,fig,wo,A,b,M,wlb,wub,W,Tr,1,no,epsR,epsI,epsC,epsZ,St);

while no
  if flag % non integer solution
    w = W(no,:); % create two child problems deleting the parent
    j = Tr(no,3);
    pai = Tr(no,2);
    nfx = Tr(no,4);
    fx = [Tr(no,5:4+nfx),j];
    vr = Tr(no,5+nfx:4+m);
    vr = vr(find(vr-j));
    Tr(no,:) = [pai,pai+1,0,nfx+1,fx,vr];
    no = no + 1;
    W(no,:) = w;
    Tr(no,:) = Tr(no-1,:);
    Tr(no,2) = pai+2;
    j = j + n;
    yj = round(w(j));
    W(no-1,j) = yj;
    W(no,j) = 1 - yj;
  end

  % round trip for version 4.2
  % version 5.2: if no > 1 & Tr(no,1) == Tr(no-1,1)

  if no > 1
    xx = Tr(no-1,1);
  else
    xx = -1;
  end

  if Tr(no,1) == xx % last two child problems with same parent
    old_no = no;
    [flag,wo,A,b,W,Tr,no,St] =
lp_nlp(NLP,GRAD,n,m,fig,wo,A,b,M,wlb,wub,W,Tr,no,no,epsR,epsI,epsC,epsZ,St);
    if flag
      k = old_no - 1;
      [flag,wo,A,b,W,Tr,no,St] =
lp_nlp(NLP,GRAD,n,m,fig,wo,A,b,M,wlb,wub,W,Tr,k,no,epsR,epsI,epsC,epsZ,St);
      if flag & old_no == no
        if W(old_no,n+m+2) > W(k,n+m+2)

```

```
        W([k old_no],:) = W([old_no k],:); % switch last two nodes
        Tr([k old_no],:) = Tr([old_no k],:);
    end
end
end
else
    [flag,wo,A,b,W,Tr,no,St] =
lp_nlp(NLP,GRAD,n,m,fig,wo,A,b,M,wlb,wub,W,Tr,no,no,epsR,epsI,epsC,epsZ,St);
end
end
```

A subrotina LP_NLP.m é chamada para resolver os problemas de LP, com condições de integralidade relaxadas, e os problemas de NLP quando soluções inteiras são obtidas nos LPS.

```
function [flag,wo,A,b,W,Tr,no,St] =
lp_nlp(NLP,GRAD,n,m,fig,wo,A,b,M,wlb,wub,W,Tr,k,no,epsR,epsI,epsC,epsZ,St)
```

```
% authors: Argimiro R. Secchi and Elaine C. Pereira
%   Chemical Engineering Department
%   Applied Math Department
%   Universidade Federal do Rio Grande do Sul, Brasil
%   arge@enq.ufrgs.br and dmtecp@super.furg.br
%
% version 1.0: 15/mar/1999
```

```
% * * * Solution of the k-th LP * * *
%  $z_l = \min(x,y,u,\alpha)$  alpha
% s.t.  $\alpha \geq C'y + f(x_j) + gf(x_j)'(x-x_j) + \text{Mu}$ 
%  $By + g(x_j) + gg(x_j)'(x-x_j) \leq u$ 
%  $x$  in  $X$ 
%  $y$  in  $Y$ 
%  $\alpha$  in  $R^+$ 
```

```
nm = n+m;
zu = wo(nm+2);
w = W(k,:);
w(nm+1) = 0; % u = 0 (initial guess for LP)
if w(nm+2) == inf
    w(nm+2) = 0; % alpha = 0 (initial guess for LP)
end
```

```
nfx = Tr(k,4);
nlps = size(St,1);
```

```
if nfx
    fx = n+Tr(k,5:4+nfx); % yj fixed in w
    vr = [1:n,n+Tr(k,5+nfx:4+m),nm+1,nm+2]; % w - yj fixed
    br = b - A(:,fx)*w(fx)';
    h = [zeros(nm-nfx+1,1); 1];
    wr = lp(h,A(:,vr),br,wlb(vr),wub(vr),w(vr))';
    w(vr) = wr;
else
    h = [zeros(nm+1,1); 1];
    w = lp(h,A,b,wlb,wub,w)';
end
```

```
St(nlps,2) = St(nlps,2) + 1; % LPs counter
```

```

%if w(nm+1) > epsC    % constraint violation
% w(nm+2) = inf;
%end

W(k,:) = w;
zl = w(nm+2);    % lower bound for Z

if zl+epsZ >= zu    % deleting k-th node of the tree
    flag = 1;
    W = W([1:k-1, k+1:no],:);
    Tr = Tr([1:k-1, k+1:no],:); % delete all P(i,k) such that z(i,k) >= zu
    no = no - 1;
    return;
end

x0 = w(1:n);
y = w(n+1:n+m);
[dym,j] = max(abs(y-round(y)));

if dym < epsI
    flag = 0;    % integer solution of last LP
    y = round(y);
    Tr(no,3) = 0;

% * * * Solution of the NLP * * *
% Z = min(x) C'y + f(x) + Mu
% s.t. By + g(x) <= u
%    x in X
%    y fixed
%    u in R+

op = foptions;
op(1) = 1;    % 1 - display some results; 0 - nothing
op(2) = epsR;
op(3) = epsZ;
op(4) = epsC;
op(13) = 0;
op(14) = 150; % max function evaluation

% constr returns the best solution, even violating the constraints
% then u does not need be computed explicitly.

[x,op,lambda] = constr(NLP,x0,op,wlb(1:n),wub(1:n),GRAD,y);

% getting the values of f, g, and their gradients
[f_nlp,g_nlp] = eval([NLP '(x,y)']);
u = max(g_nlp);
[gf_nlp,gg_nlp,C,B] = eval([GRAD '(x,y)']);
f = f_nlp - C * y';

```

```

g = g_nlp' - B * y';
gf = gf_nlp;
gg = gg_nlp';

if u > epsC
    znl = inf;
    lambda=-lambda(1:size(B,1));
    A = [A; zeros(1,n),C+lambda*B,M-sum(lambda),-1]; % adding Bender's cut
    b = [b; -f-lambda*g];
else
    znl = f_nlp;
    A = [A; gf,C,M,-1]; % adding OA
    b = [b; gf*x'-f];
end

if fig, planos(n,A,b,y,u,wlb(1:n),wub(1:n),1); end

if znl < zu
    zu = znl;
    wo = [x,y,u,zu];
end

nlps = nlps + 1; % NLPs counter
St(nlps,1:2) = [size(A,1), 0];

% round trip for version 4.2
% version 5.2: while no > 0 & W(no,nm+2) >= zu

if no > 0
    xx = W(no,nm+2);
else
    xx = -inf;
end
while no > 0 & xx >= zu
    no = no - 1;
    W = W(1:no,:);
    Tr = Tr(1:no,:); % delete all P(i,k) such that z(i,k) >= zu
    if no > 0, xx = W(no,nm+2); end
end
else
    Tr(k,3) = j; % less integer yj
    flag = 1; % non-integer solution
end

```