

## DETERMINAÇÃO DO MÍNIMO GLOBAL EM PROBLEMAS DE SÍNTESE DE ESTRUTURAS

L. M. C. Simões

Professor Auxiliar, Departamento de Engenharia Civil, Universidade de Coimbra (Portugal)

### Resumo

Neste artigo são descritas duas estratégias que permitem a obtenção do óptimo global de estruturas calculadas para um consumo mínimo de material. São apresentados, pela primeira vez, em problemas de optimização de estruturas, exemplos de comportamento não convexo deste programa bilinear. Indicam-se duas versões da primeira metodologia, do tipo "branch and bound" (B & B). A segunda, que corresponde a um desenvolvimento original, é do tipo "cutting plane" (plano de corte), deduzida a partir de uma generalização do algoritmo de Benders, que é resolúvel por códigos para programas lineares mistos. Se a escolha das secções se limitar a um conjunto discreto de alternativas, verifica-se uma convergência mais rápida para o mínimo global, num número finito de passos.

### 1. Introdução

No 25º ano a seguir ao trabalho publicado por Schmidt na segunda Conferência em Métodos de Computação Electrónica [16], pode dizer-se que as técnicas de programação matemática tiveram um lugar privilegiado no desenvolvimento de novas metodologias para o cálculo estrutural auxiliado por computadores. Entre as aplicações mais correntes, refere-se a determinação do peso mínimo de uma estrutura de topologia constante. A literatura específica é tão numerosa que se poderá remeter o leitor para a recolha bibliográfica [19].

Os métodos que têm sido apresentados mais frequentemente para solucionar esses problemas podem ser catalogados em duas categorias: critério de optimalidade e programação matemática. O primeiro é baseado na minimização do Lagrangeano e o outro decorre da aplicação de métodos de optimização não-linear.

De um modo geral, aceita-se que qualquer das alternativas possui falhas devidas à natureza não-linear e não-convexa do problema. Não é possível garantir que o mínimo local, fornecido por métodos que assentam num comportamento convexo, é também o mínimo global.

A determinação do volume mínimo de uma treliça que se comporta de uma forma elástica é um exemplo desses problemas. Contudo, como na literatura não existem referências relativas ao comportamento não-convexo deste tipo de estruturas, este trabalho foi iniciado com a apresentação de situações em que se verifica a não convexidade do domínio, em síntese de estruturas.

Foram escolhidas duas das estratégias mais apropriadas à optimização não-convexa. O método de B&B, que permite a convergência para o mínimo global, pode ser aplicado à solução de problemas com variáveis separáveis (por exemplo, as funções de uma variável e o produto destas). Este algoritmo é competitivo, dado que em cada nó se resolve um programa linear. São resolvidos problemas por esta metodologia e apresentados alguns resultados da experiência computacional do autor.

Em alternativa, propõe-se uma generalização do algoritmo de Benders em que são utilizadas funções duais em lugar de variáveis duais. Deduz-se um programa "master" que é equivalente ao programa matemático do peso mínimo. O "master" é um programa linear misto (variáveis reais e 0-1), que se pode resolver de um modo eficiente. O desenvolvimento apresentado é original, sendo complementado com um exemplo de aplicação.

## 2. Exemplos de comportamentos não-convexos em programas bilineares de optimização de estruturas

Os exemplos a seguir indicados servem para ilustrar alguns dos problemas que surgem, quando se pretende efectuar a síntese de estruturas que se comportam de um modo elástico.

### 2.1. Cálculo de uma grelha

A grelha representada na figura 1 é constituída por vigas "sandwich" ortogonais e suporta um carregamento perpendicular ao seu plano.

Supondo que a geometria da estrutura é conhecida, incluindo o número de vigas, comprimento dos vãos e condições de apoio, as restrições vão estar relacionadas com as tensões por flexão no nó B (provocadas pelos momentos  $m_1$  e  $m_3$ ) e tensões por flexão nas secções críticas entre os nós AB e BC (provocadas pelos momentos  $m_2$  e  $m_4$ , respectivamente).

Pretende-se calcular a área das secções  $y_1$  e  $y_2$ , de modo a minimizar o volume da grelha, isto é:

$$\min l_1 y_1 + l_2 y_2 \quad (1)$$

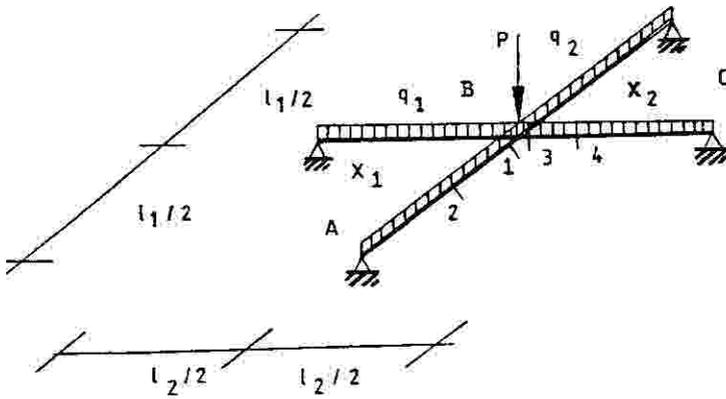


Fig. 1 - Grelha.

onde  $l_1$  e  $l_2$  são os comprimentos das vigas que se cruzam perpendicularmente. Variando simplesmente  $y_1$  e  $y_2$ , de modo a representar o domínio admissível da grelha, indicam-se a seguir duas situações em que o comportamento é não convexo.

1º CASO

Supõe-se a actuar em toda a grelha, uma carga uniformemente distribuída  $q_1 = 0.8 = q_2$ , sendo a carga pontual  $P$  nula.

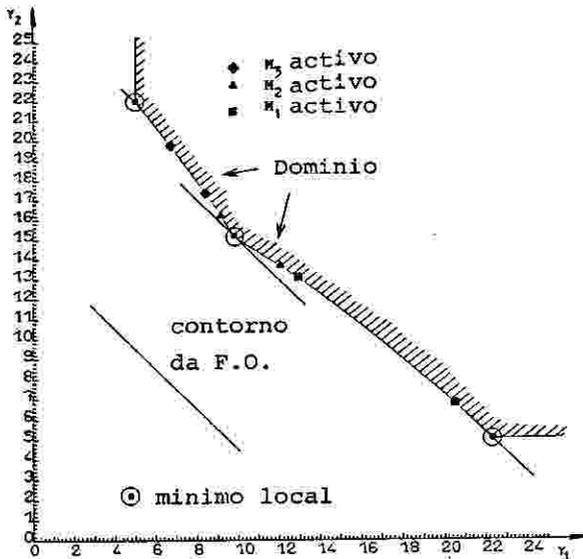


Fig. 2 - Domínio da grelha para o 1º caso.

Os comprimentos das vigas são  $l_1 = 30$  e  $l_2 = 31$ . Considera-se que quer os momentos estáticos  $w_j$ , quer os momentos de inércia  $I_j$  são directamente proporcionais à área das secções, ou seja:

$$w_j = 0.5 y_j \quad ; \quad I_j = 0.3 y_j \quad ; \quad j = 1,2 \quad (2)$$

O problema daí resultante possui os três óptimos locais indicados na figura 2.

**2º CASO**

Considera-se outra situação de carregamento a que corresponde uma carga uniformemente distribuída  $q_1 = 1.2 = q_2$  e uma carga concentrada  $P = 25.0$

Os comprimentos das vigas passam a ser  $l_1 = 21$  e  $l_2 = 24$ . As novas relações que ligam as características geométricas das secções são:

$$w_j = 5. y_j \quad ; \quad I_j = 25. y_j \quad (3)$$

Nestas condições seriam apenas obtidos os dois mínimos locais representados na figura 3.

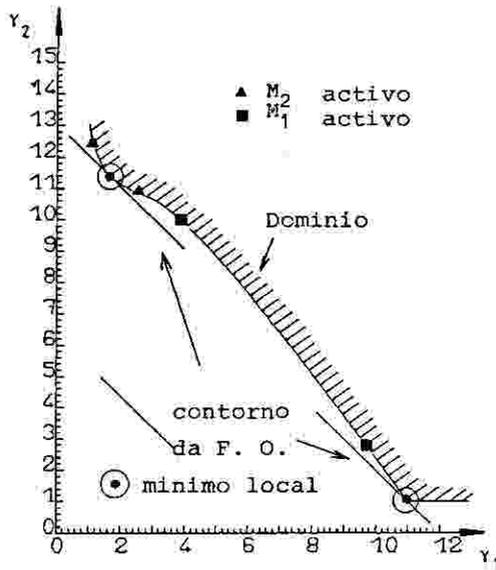


Fig. 3 - Domínio da grelha para o 2º caso.

## 2.2. Minimização do peso de uma treliça com dez barras

O segundo exemplo diz respeito à minimização da treliça hiperstática da figura 4 que está sujeita à acção de duas cargas verticais. Limitando as tensões admissíveis a  $s_j = +2.5$ , os deslocamentos máximos  $d_j = +3.5$  e para um vector das cargas nodais  $\lambda$  igual a:

$$\lambda^T = [0, 0, 0, -10, 0, 0, 0, -10,]$$

Para manter a topologia da estrutura, impõe-se um limite inferior na área das barras  $a_j$  de 0.1

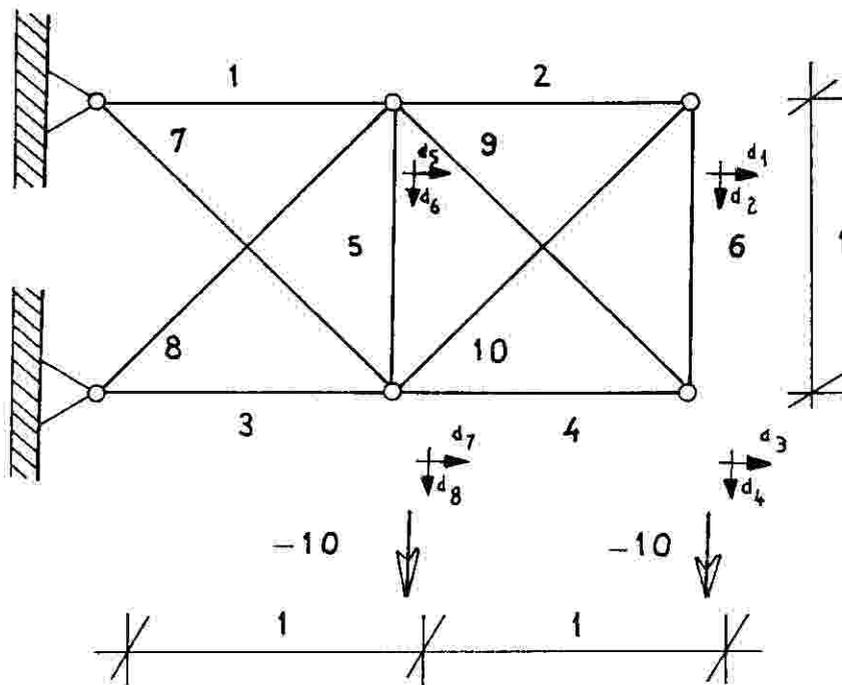


Fig. 4 - Treliça de dez barras.

Obtêm-se duas soluções óptimas a seguir indicadas:

OF = 219.93

a	48.7	0.1	35.6	24.1	0.1	1.2	9.4	34.3	34.1	0.1
s	0.4	-0.3	-0.6	-0.4	2.2	0.0	2.9	-0.8	0.8	0.7
d	0.2	-3.5	-1.0	-3.5	0.4	-1.3	-0.6	-3.5		

OF = 223.34

a	48.7	0.1	38.1	23.3	0.1	0.1	13.7	33.1	33.0	0.1
s	0.4	0.0	-0.5	-0.4	1.3	0.0	2.0	-0.9	0.9	0.0
d	0.4	-3.5	-1.0	-3.5	0.4	-1.3	-0.5	-2.6		

Qualquer destes pontos é uma solução das equações de Kuhn-Tucker e corresponde a um conjunto específico de multiplicadores de Lagrange. Ambos minimizam localmente o volume da estrutura. Qualquer direcção que ligue estas soluções possuirá, pelo menos, um ponto fora do domínio do problema, ou então um volume superior ao mínimo local.

### 3. Métodos de "branch and bound"

Em face da existência de mínimos locais nos problemas de optimização com restrições bilineares, é necessário desenvolver métodos de programação matemática adaptados a esse tipo de comportamento não-convexo. A estratégia B & B converge para o mínimo global dado que o domínio inicial, não-convexo, é substituído por uma sequência de domínios convexos que se intersectam (utilizam-se para isso funções convexas, que subestimam as funções não-convexas do problema inicial). Como se sabe, se um programa matemático tiver uma função objectivo convexa e um domínio convexo, qualquer solução local é também a global.

Os dois ingredientes principais desta metodologia são uma árvore combinatória (cujos nós vão sendo definidos) e limites superiores e inferiores associados a cada nó da árvore. Nestas condições é possível eliminar um grande número de soluções possíveis sem ser necessário calculá-las.

Tratando-se de um algoritmo de enumeração implícita, a sua eficiência pode ser bastante melhorada se se partir de uma boa solução inicial, pertencente ao domínio do problema (que fornece um limite superior adequado).

Pode-se eliminar uma solução parcial em cada uma das situações seguintes:

1. Chega-se a uma solução que pertence ao domínio do problema inicial;
2. O subprograma em que se utilizam subestimativas, não tem solução;
3. Não é possível encontrar uma solução nesse domínio com um peso inferior ao da estrutura utilizada como limite superior.

Se uma solução parcial for eliminada, isto significa que todas as continuações desse nó foram enumeradas de um modo implícito (e por esse motivo não é necessário enumerá-las explicitamente). O algoritmo termina quando o último nó for eliminado; o óptimo global corresponde à estrutura que dá o menor limite superior.

Tem de se assegurar que nenhum dos nós é estudado mais de uma vez, ou omitido no cálculo, sendo necessário varrer a árvore combinatória.

### 3.1. Transformação de funções factoráveis em subestimativas convexas

As expressões bilineares do tipo:

$$p x + q y + k x y \quad (4)$$

são casos particulares do tipo de programas que envolvem funções factoráveis.

Como a subestimativa convexa de uma expressão linear é ela própria, é necessário determinar a envolvente dos termos em que surgem produtos de variáveis. Considere-se a função  $f(x,y) = x y$  definida entre os limites:

$$a \leq x \leq b \quad ; \quad c \leq y \leq d \quad (5)$$

Nos semi-espacos em que as variáveis são todas não-negativas, os valores da função nos cantos do domínio coincidem com a subestimativa. Como são suficientes três pontos para definir um plano em 3-D, a função é subestimada por um de dois planos que se intersectam, segundo uma recta determinada a partir dos cantos que correspondem a valores intermédios da função bilinear. Toma-se para subestimativa convexa a coordenada  $z$  no plano mais próximo da expressão factorável:

$$z_1 = c x + a y - a c \quad (6)$$

$$z_2 = d x + b y - b d \quad (7)$$

$$z = \max \{ z_1, z_2 \} \leq f(x,y) = x y \quad (8)$$

De um modo semelhante para sobrestimativas côncavas, consideram-se dois planos que se intersectam segundo uma recta determinada a partir dos cantos a que correspondem os valores mais elevado e mais baixo da função bilinear. Tira-se a coordenada no plano mais próximo da função inicial:

$$z_1^0 = d x + a y - a d \quad (9)$$

$$z_2^0 = c x + b y - b c \quad (10)$$

$$z^0 = \min \{ z_1^0, z_2^0 \} \geq f(x,y) = x y \quad (11)$$

Note-se que têm que adaptar-se estas aproximações quando se consideram os outros quadrantes. De realçar que a função subestimativa  $z$  (e sobrestimativa  $z^0$ ) não é diferenciável em todos os pontos do seu domínio. É contudo possível ultrapassar esse obstáculo através da construção de um

programa linear (PL) equivalente. A maneira mais simples de o fazer consiste em adicionar algumas restrições e variáveis. O programa matemático:

$$\min c^T x \quad (12)$$

$$\text{sa } Ax \geq b \quad (13)$$

$$f^T x + \max \{ g_1^T x, g_2^T x \} \leq h \quad (14)$$

$$p^T x + \min \{ q_1^T x, q_2^T x \} \geq r \quad (15)$$

é pois equivalente ao PL :

$$\min c^T x \quad (16)$$

$$\text{sa } Dx \geq e \quad (17)$$

$$f^T x + u \leq h \quad (18)$$

$$p^T x + v \geq r \quad (19)$$

$$u \geq g_1^T x \quad u \geq g_2^T x \quad (20)$$

$$v \leq q_1^T x \quad v \leq q_2^T x \quad (21)$$

Esta formulação não é a única possível. Recomenda-se a introdução de mais variáveis quando aparecer o mesmo termo não-diferenciável em várias restrições.

Se as relações em que entram termos bilineares forem igualdades, como é o caso das equações de equilíbrio no método dos deslocamentos, essas equações devem ser substituídas pelo par de inequações equivalentes. Isto conduz a um grande aumento do número de variáveis e restrições. Pode-se contudo demonstrar que nos problemas de optimização de estruturas uma dessas inequações é sempre não-activa, pelo que não é necessário considerá-la, o que reduz de um modo significativo a dimensão de cada PL.

### 3.2. Sequência de fora para dentro

É apresentado a seguir o algoritmo de Soland [18] para restrições separáveis, que foi adaptado pelo autor às funções subestimativas convexas acima descritas.

Na primeira iteração, considera-se o programa linear ( $P_p$ ), cujas restrições são as subestimativas convexas que substituem as restrições do programa bilinear. A solução  $x^p$  de ( $P_p$ ) é um limite inferior do óptimo que se pretende determinar.

$$\min c^T x \quad (22)$$

$$\text{sa } Ax \geq b \quad (23)$$

$$l_p \leq x \leq L_p \quad (24)$$

Se  $x^p$  não pertencer ao domínio do problema bilinear inicial, tem de se restringir o domínio do problema, de modo a ser obtida uma solução que se encontre no domínio do programa bilinear.  $(P_p)$  é pois substituído por um conjunto de subproblemas definidos de entre os limites do problema inicial, dado que existe uma solução óptima  $x^*$  para pelo menos um dos problemas  $j \in WP$ . Calculando as soluções de todos os subproblemas daí resultantes, seja:

$$x^s = \min_{j \in WP} c^T x^j \tag{25}$$

Se  $x^s$  for uma solução que não pertence ao domínio do programa bilinear, vai-se substituir um dos problemas correspondendo a um subdomínio de  $WP$  por um conjunto de subproblemas. Então, faz-se  $p = p + 1$ . O problema  $s$  é substituído pelo conjunto  $WP$ , de modo que  $WP = (WP^{p-1} - \{s\}) \cup WS$  contenha uma solução óptima do problema inicial para pelo menos um dos programas  $WP$ .

Para cada  $j \in WP$ , ou  $x^j$  não pertence ao domínio do programa bilinear, ou  $c^T x^j > c^T x^*$ . Isto significa que se realizou algum progresso para a solução final.

A árvore combinatoria vai ter um dos seus nós identificado pelo subproblema  $j$  e os problemas que substituem  $j$  no conjunto  $WP$  estão nas extremidades dos ramos que partem para baixo desse nó. Em qualquer situação intermédia, no decurso da resolução do problema, pode-se identificar o conjunto  $WP$  como sendo o conjunto de nós que constituem folhas da árvore.

Cada nó depende do limite superior  $v$ , determinado pela solução até aí encontrada de mínimo volume. Qualquer folha cujo limite inferior seja estritamente inferior a  $v$  está activa. Se isso não acontecer, corta-se esse nó, o qual nunca mais é considerado nas iterações seguintes do algoritmo. Prossegue-se esta metodologia até que cada folha seja cortada.

Além disso, é necessário definir (de um modo heurístico) uma regra de refinamento, que permita dividir os limites das variáveis que vão originar os novos subproblemas: Escolhe-se o índice  $i$  da variável que maximiza a diferença entre o termo bilinear e a subestimativa convexa na restrição que mais violaria as restrições do programa bilinear usando  $x^s$ . O intervalo  $[l_i, L_i]$  é pois subdividido nos intervalos  $[l_i, x_i]$  e  $[x_i, L_i]$ .

Deste modo, logo que se escolhe um nó para ramificar, a partição do seu domínio só depende da solução do subproblema onde surgem as subestimativas convexas, não estando relacionado com outras partições ao mesmo nível da árvore. Este procedimento corresponde a uma forma mais fraca do teorema de convergência, dado que não é obrigatório completar as partições dos intervalos no mesmo nível da árvore, reduzindo assim o número total de subproblemas que vai ser necessário resolver.

3.3. Aplicação

O problema matemático seguinte possui restrições bilineares e é semelhante à determinação do volume mínimo de treliças (as variáveis  $x$  e  $y$  representam áreas das secções e tensões nas barras, respectivamente):

$$\begin{aligned} \min \quad & x_1 + x_2 + x_3 \\ \text{sa} \quad & x_1 x_4 + x_3 x_6 = 0 \\ & 3 x_1 x_4 + 1.2 x_2 x_5 - x_3 x_6 = 10 \\ & 5 x_4 + x_5 + x_6 \leq 2.5 \\ & .1 \leq x_1 \leq 5. ; .1 \leq x_2 \leq 5. ; .1 \leq x_3 \leq 5. \\ & 0. \leq x_4 \leq 2.5 ; 0. \leq x_5 \leq 2.5 ; -2.5 \leq x_6 \leq 0. \end{aligned}$$

Este problema possui três mínimos locais:

OF	x					
3.53	0.1	3.33	0.1	0.0	2.5	0.0
3.60	0.5	3.0	0.1	0.5	2.5	-2.5
3.60	2.5	0.1	1.0	1.0	0.0	-2.5

Para representar este problema não convexo graficamente, não é possível ir além de duas equações a três incógnitas sem se ser colocado dentro de um subespaço do problema. A figura 5 representa uma perspectiva onde são imprimidos todos os valores da função objectivo abaixo de um dado nível.

Substituindo as funções factoráveis pelas suas subestimativas convexas (dado que a determinação das envolventes das funções lineares é trivial), obtém-se (na formulação mais extensa) um PL com 10 variáveis e 15 inequações. Qualquer dos nós da árvore combinatoria é definido pelos limites nas variáveis  $x_1$  a  $x_6$ . Na árvore combinatoria da figura 6, estão representados os resultados da estratégia B & B, utilizando uma regra de refinamento fraca e uma ramificação do tipo alargado (escolher em primeiro lugar o limite inferior mais baixo).

3.4. Experiência de computação

Foram utilizadas outras subestimativas convexas [17], que conduziram por vezes a problemas cuja dimensão é menor. Contudo, o número de subproblemas que seria necessário resolver cresce de uma forma significativa, em virtude dessas aproximações serem mais grosseiras.

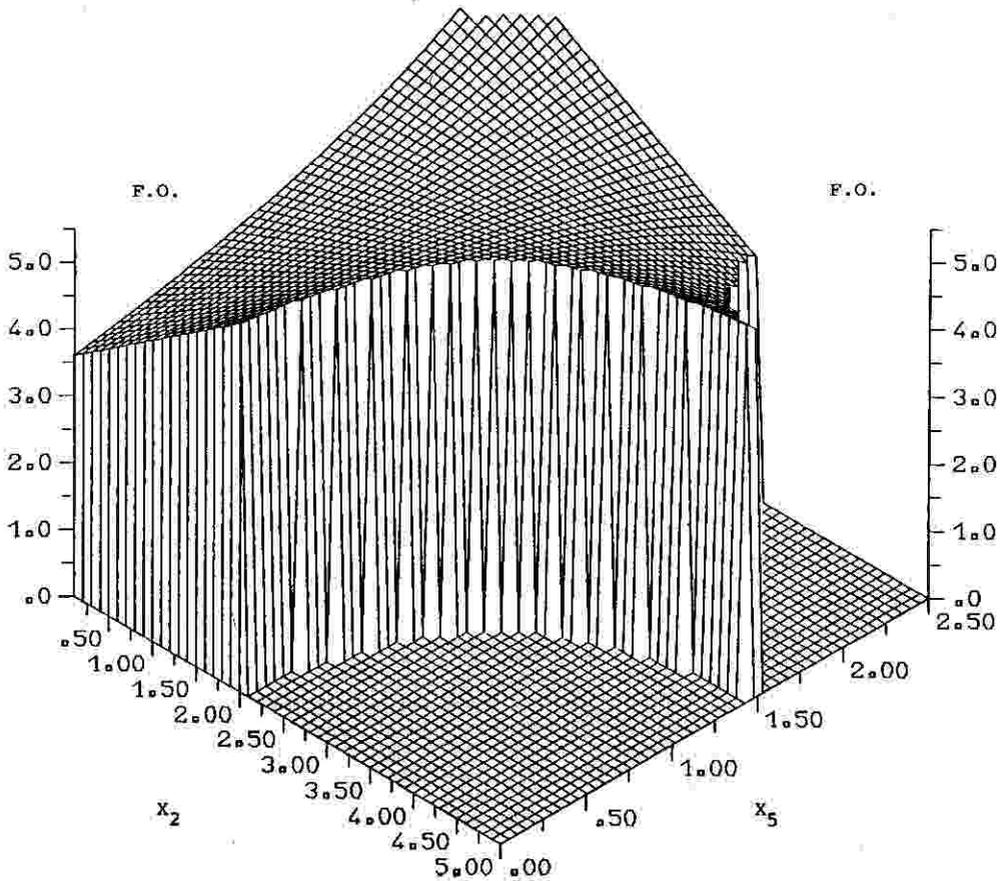


Fig. 5 - Gráfico para diversos valores de  $x_2$  e  $x_5$  mantendo  $x_6$  constante.

Nesse trabalho foram igualmente abordados outros tipos de exploração da árvore combinatória. Utilizou-se uma estratégia alternativa, que consistia em escolher sempre o nó direito do nível inferior até se chegar a um nó que seria eliminado; voltava-se atrás na árvore e prosseguia-se, utilizando a mesma regra até serem percorridos os nós remanescentes. Esta estratégia (oposta à que consiste na escolha do nó associado à subestimativa cuja função objectivo é mais baixa), embora reduzisse o espaço de armazenagem necessário, iria aumentar o número de programas lineares que seria necessário resolver entre 30 e 35%.

A regra de ramificação que é seguida para subdividir o intervalo da variável, cuja aproximação mais se afasta do valor real, tem uma influência mais marcante. Se o intervalo for dividido ao meio, em lugar de se utilizar a informação do programa que utiliza subestimativas convexas, o número de problemas necessário para obter o mínimo global crescerá entre 70 e 80%.

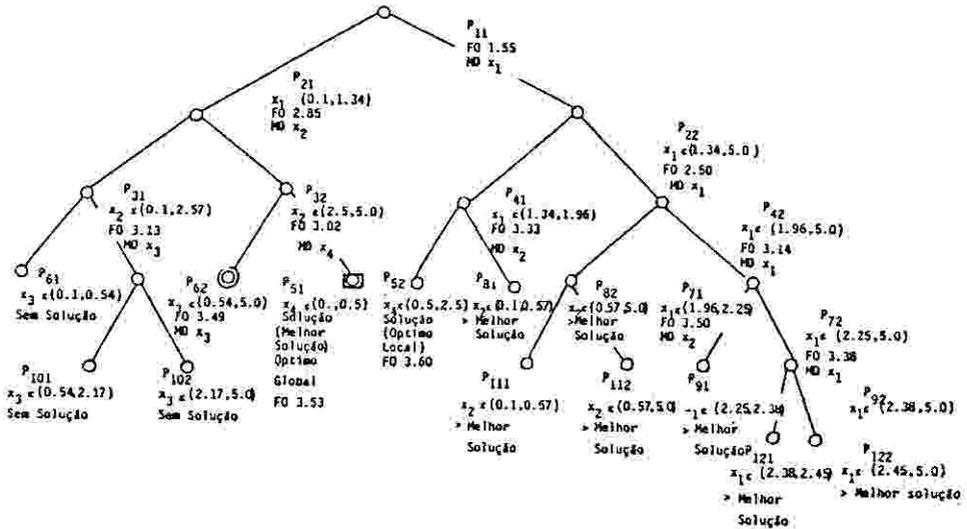


Fig. 6 - Arvore combinatória correspondente à sequência de fora para dentro.

### 3.5. Sequência de dentro para fora

Um algoritmo alternativo de B & B que foi inicialmente desenvolvido por Reeves [14] para o tipo de programação em que, quer a função objectivo, quer as restrições presentes, são quadráticas pode ser igualmente aplicável a problemas nos quais as restrições e/ou a função objectivo são redutíveis a formas factoráveis. Cada iteração vai ser efectuada sobre um subintervalo do domínio inicial, dado pelos limites nos valores das variáveis.

Basicamente este algoritmo é constituído por três passos:

No primeiro passo de cada iteração determina-se um ponto base a partir se efectuam as operações de "branch and bound". Os óptimos locais de programas convexos que determinam o volume mínimo são valores que asseguram à partida melhores resultados, embora seja possível iniciar o algoritmo em pontos que não sejam mínimos locais (ou mesmo pertencentes ao domínio do problema com restrições bilineares). Quando se utiliza este tipo de algoritmo como verificação do

resultado obtido pela aplicação de métodos convexos, o mínimo local encontrado é um ponto de partida ideal.

Logo que seja especificado um ponto de partida, no segundo passo elimina-se o intervalo que o rodeia. Se o ponto  $x^V$ , onde V representa o número de iterações do algoritmo, pertencer ao domínio do programa bilinear, elimina-se o intervalo para o qual  $x^V$  é a solução global do programa de otimização em que se utilizam as subestimativas convexas. Este segundo passo consiste em três operações:

1. Divide-se o intervalo considerado em subintervalos que rodeiam o ponto base.
2. Define-se uma região em cada subintervalo para a qual o ponto de partida é também o mínimo global do problema inicial.
3. Obtém-se um intervalo de eliminação a partir da união das regiões eliminadas dos subintervalos individuais.

No caso de se partir de um valor de  $x^V$ , que não pertença ao domínio do programa inicial, define-se um intervalo para o qual o PL obtido utilizando subestimativas convexas (LCE) não possui qualquer solução; esse intervalo é igualmente eliminado.

No último passo do algoritmo, procede-se ao B & B. As regiões que ainda não foram eliminadas são divididas em subintervalos e para cada um destes resolve-se um LCE. Os limites inferiores daí resultantes, são comparados com o valor do limite superior mais baixo v que pertence ao domínio do programa bilinear e são eliminados, caso o igualem ou excedam. Se ainda restar algum subconjunto do domínio inicial do problema inicia-se nova iteração, tomando um ponto base situado dentro do intervalo não eliminado.

Na figura 7 estão representadas as duas estratégias B & B descritas.

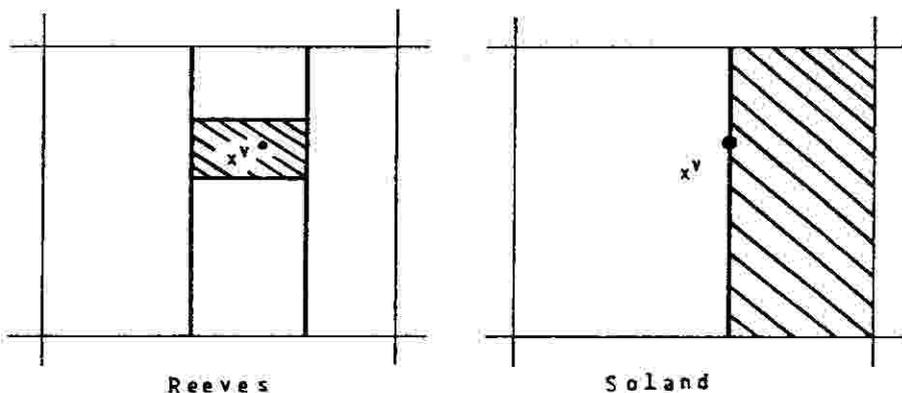


Fig. 7 - Estratégias de Reeves e Soland.

O critério de paragem, que permite a verificação do mínimo global para valores limites fixados, requer que a solução  $x^V$  dada pelo LCE nesse intervalo esteja também no domínio do programa

bilinear. Isto implica, atendendo às subestimativas convexas donde se partiu, que o valor óptimo das variáveis de decisão se encontra num ponto extremo.

### 3.6. Aplicação

Nesta secção aplica-se o método de Reeves como verificação do volume mínimo de uma treliça de três barras, obtido por rotinas de programação convexa. Considera-se a treliça de três barras representada na figura 8 e que está sujeita a duas condições alternativas de carregamento. Supõe-se que as variáveis de decisão (áreas das secções), podem variar de uma forma contínua.

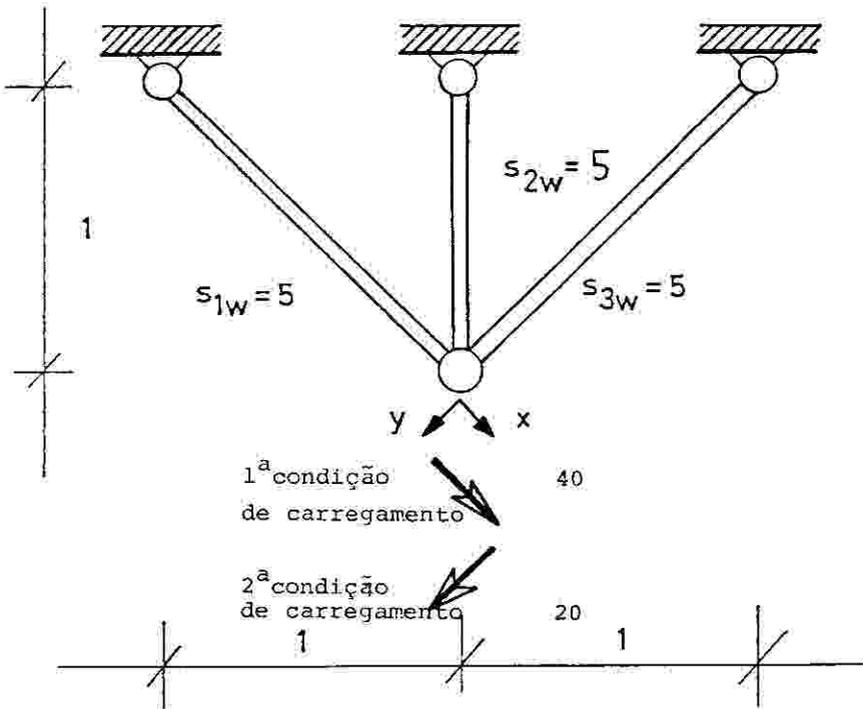


Fig. 8 - Treliça de três barras sujeita a duas condições alternativas de carregamento.

Utilizando o método dos deslocamentos, pode escrever-se o programa:

$$\begin{aligned} \min \quad & \sqrt{2} a_1 + a_2 + \sqrt{2} a_3 \\ \text{sa} \quad & (\sqrt{2}/2) a_1 d_1^1 + 1/2 a_2 (d_1^1 + d_2^1) = 40. \\ & 1/2 a_2 (d_1^1 + d_2^1) + (\sqrt{2}/2) a_3 d_2^1 = 0. \\ & (\sqrt{2}/2) a_1 d_1^2 + 1/2 a_2 (d_1^2 + d_2^2) = 0. \\ & 1/2 a_2 (d_1^2 + d_2^2) + (\sqrt{2}/2) a_3 d_2^2 = 20. \\ & 0 \leq (\sqrt{2}/2) d_1^1 \leq 5 \quad ; \quad -5 \leq (\sqrt{2}/2) d_1^2 \leq 0 \\ & 0 \leq (\sqrt{2}/2) (d_1^1 + d_2^1) \leq 5 \quad ; \quad 0 \leq (\sqrt{2}/2) (d_1^2 + d_2^2) \leq 5 \\ & -5 \leq (\sqrt{2}/2) d_2^1 \leq 0 \quad ; \quad 0 \leq (\sqrt{2}/2) d_2^2 \leq 5 \\ & 1. \leq a_1 \leq 11. \quad ; \quad 1. \leq a_2 \leq 4. \quad ; \quad 1. \leq a_3 \leq 5 \end{aligned}$$

Depois de aproximar os termos não-convexos com as envolventes convexas dadas para funções factoráveis e estabelecendo para um dos limites do domínio os valores óptimos das variáveis de decisão no LCE daí resultante, tem-se para  $V = 1$ :

### 1º PASSO

O valor óptimo vai ser

$$\text{OF} = 15.969 \quad ; \quad a_1 = 7.024 \quad ; \quad a_2 = 2.138 \quad ; \quad a_3 = 2.756$$

### 2º PASSO

Em primeiro lugar, tenta-se eliminar todo o domínio de  $a_j$ , i.e.:  $[a_{j1}, a_{ju}]$  para tantos valores de  $j$  quantos seja possível. A intenção deste procedimento é saber se é ou não possível reduzir as operações de B & B do 3º PASSO, de modo a minimizar o número total de LCE. Limita-se a violação máxima das restrições pertencentes ao programa bilinear e eliminam-se os pontos que pertençam a este com um volume superior ao mínimo local. O intervalo de eliminação é obtido depois de se resolverem 8 LCE duas vezes.

O intervalo de eliminação será:

$$e_l^1 = [5.69 \ 1.85 \ 2.22] \quad ; \quad e_u^1 = [7.97 \ 2.54 \ 3.14]$$

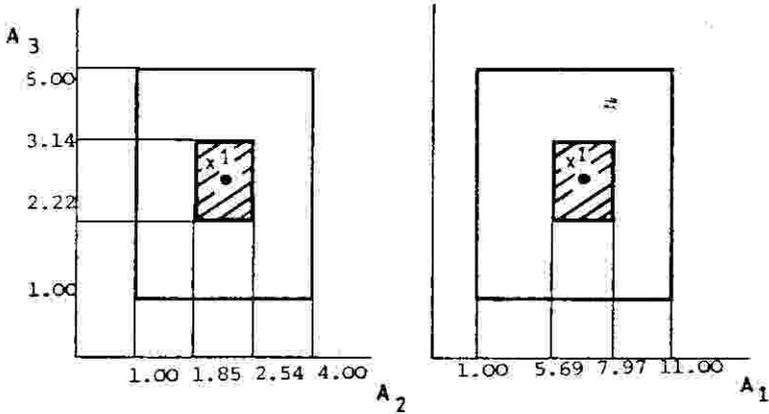


Fig. 9 - Intervalo de eliminação no 2º passo,  $V = 1$ .

3º PASSO

Em seguida toma-se a região remanescente depois de ser removido o intervalo de eliminação e efectua-se a sua partição em  $3 \times 2 = 6$  regiões. Podem ser eliminados os subintervalos 1,2,5 e 6 (1 e 5, porque o LCE não tem solução nos domínios respectivos; 2 e 6, porque as soluções dos LCE excedem o valor do mínimo local).

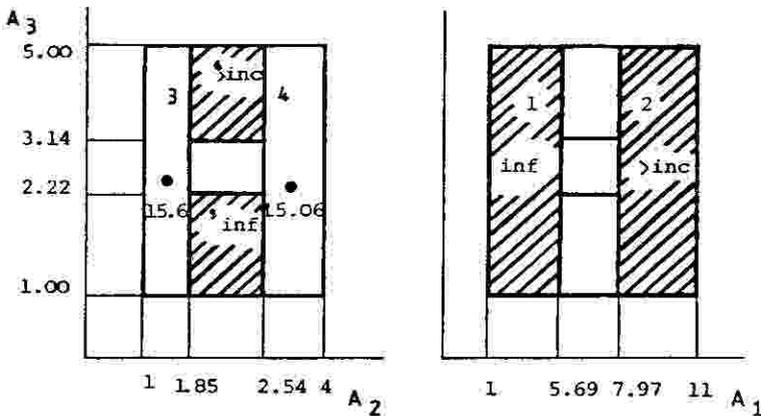


Fig. 10 - Intervalo de eliminação no 3º passo,  $V = 1$ .

Na iteração seguinte do algoritmo,  $V = 2$ , parte-se para o intervalo 4 a que corresponde o limite inferior mais baixo. Como o ponto base não pertence ao domínio do problema, o procedimento é simplificado. Não é necessário proceder a qualquer subdivisão e só se considera um intervalo. O algoritmo prosseguiria, terminando após 8 ciclos que compreenderiam 78 LCE.

### 3.7. Discussão

Muitas vezes em problemas de optimização de treliças, o óptimo que é encontrado por rotinas de programação convexa é o óptimo global. A sequência de dentro para fora serve como processo de verificação, em que se estabelece se esse mínimo é global ou se é possível encontrar um valor melhor. Além disso, tomar como ponto de partida um mínimo local aumenta a possibilidade de os problemas que utilizam subestimativas convexas não terem solução, ou de a solução obtida exceder o valor do mínimo local.

Do ponto de vista teórico, o número de subintervalos de eliminação definidos à volta do ponto base (no 2º Passo do algoritmo) cresce exponencialmente com o número de variáveis, o que constitui uma grande desvantagem desta metodologia. Um factor que tende a diminuir a importância deste argumento é que à medida que aumentam as dimensões do problema, aumenta também a possibilidade de não ser necessário explorar grande parte destes subintervalos. Outro aspecto negativo deste algoritmo, tem a ver com a determinação do intervalo a eliminar, que rodeia o ponto base. Este intervalo é obtido por tentativas, o que significa que os subproblemas (cujo número cresce, em potencial, exponencialmente) têm de ser resolvidos mais do que uma vez.

## 4. Método da Decomposição de Recursos

O algoritmo da decomposição de recursos de Benders tem sido até agora utilizado para resolver problemas convexas e parcialmente convexas (eg: [7]) que envolvam dois tipos de variáveis. Tem uma interpretação em termos económicos, vendo os multiplicadores de Lagrange como preços sombra de uma estratégia de decomposição que não é totalmente descentralizada. O órgão central produz as decisões finais atribuindo pesos óptimos às propostas de cada um dos subsistemas, tendo sempre em atenção as ofertas anteriores.

Se o conjunto de variáveis  $y$  na minimização com restrições bilineares for fixado, o problema nas variáveis  $x$  daí resultante é bastante simples. Embora o problema inicial seja não-convexo considerando simultaneamente as variáveis  $x$  e  $y$ , mantendo  $y$  constante ter-se-á um problema linear em termos de  $x$  para resolver. A ideia fundamental, que permite ao problema (P) :

$$\min c^T y \quad (26)$$

$$\text{sa } g_i^T x + x^T H_i y \geq b_i \quad ; \quad i = 1, \dots, m \quad (27)$$

$$0 \leq y \leq y_{\max} \quad \text{ie: } y \in Y \quad (28)$$

$$x_l \leq x \leq x_u \quad \text{ie: } x \in X \quad (29)$$

ser visto no espaço das variáveis  $x$  é o conceito de projecção:

$$\min v(x) \tag{30}$$

$$\text{sa } x \in X \cap V \tag{31}$$

onde

$$v(x) = \text{infimum } c^T y \tag{32}$$

$$\text{sa } g_i^T x + x^T H_i y - b_i \geq 0 \tag{32}$$

$$0 \leq y \leq y_{\max} \tag{33}$$

e se tem:

$$V = \{x: g_i^T x + x^T H_i y - b_i \geq 0 \text{ para } y \in Y\} \tag{34}$$

Note-se que  $v(x)$  é o óptimo de (P), mantendo  $x$  fixo. Determinar  $v(x)$  é consequentemente muito mais fácil do que resolver o programa com restrições bilineares.

Considere-se  $(P(x))$  o problema de optimização:

$$\min c^T y \tag{35}$$

$$\text{sa } g_i^T x + x^T H_i y - b_i \geq 0 \quad ; \quad i = 1, \dots, m \tag{36}$$

$$0 \leq y \leq y_{\max} \tag{37}$$

O conjunto  $V$  consiste nos valores de  $x$  para os quais  $(P(x))$  pertence ao domínio do problema e a intersecção  $X \cap V$  é a projecção do domínio do problema (P) no espaço dos  $X$ . Como a função  $v$  e o conjunto  $V$  só são conhecidos de um modo implícito através das suas definições, torna-se difícil utilizar o conceito de projecção do problema como forma de o resolver.

Para ultrapassar esta dificuldade utiliza-se um método de plano de corte que permita construir aproximações para  $v$  e  $V$ . A ideia base consiste na utilização da dualidade existente na programação linear aplicada a  $v$  e  $V$ , depois de projectar o problema inicial.

Resolve-se o problema "master" daí resultante, através de um processo de relaxação, o qual vai gerar aproximações a  $v$  e  $V$  convergindo o processo para o óptimo global. Assim, determinam-se os multiplicadores de Lagrange para  $(P(x))$  correspondentes a diversos valores de  $x$  e que vão sendo obtidos sucessivamente, adicionando-se tantos planos de corte ao programa "master" relaxado quantos os necessários até satisfazer o critério de convergencia.

#### 4.1. Formulação do problema "master"

O programa "master" (PM) que é equivalente a (P) é deduzido à custa de três teoremas.

(A) Projectar (P) sobre  $x$  daí resultando  $(P(\underline{x}))$ .

(B) Invocar a representação dual de  $V$  em termos do conjunto de regiões que o contem.

(C) Invocar a representação dual de  $v$  em termos do ínfimo do conjunto de funções que o dominam.

Manipulando estes resultados obtém-se o PM a seguir indicado:

$$\min \eta \quad (38)$$

$$\text{sã} \quad \eta - \sum_{i=1,m} u_i^s (b_i - g_i^T x) + \sum_{j=1,n} [-c_j + \sum_{i=1,m} u_i^s (x^T H_i)_j]^+ y_{j\max} \geq 0 \quad (39)$$

$$- \sum_{i=1,m} u_i^c (b_i - g_i^T x) + \sum_{j=1,n} [\sum_{i=1,m} u_i^c (x^T H_i)_j]^+ y_{j\max} \geq 0 \quad (40)$$

$$\sum_{i=1,m} u_i^c = 1 \quad ; \quad u_i^s \geq 0 \quad ; \quad u_i^c \geq 0 \quad (41)$$

onde  $u_i^c$  e  $u_i^s$  são multiplicadores de Lagrange correspondentes às funções de corte e apoio, respectivamente.

O PM tem apenas interesse teórico, já que possui um número enorme de restrições. No entanto, pode ser resolvido através de uma sequência de subproblemas: em cada iteração é resolvida uma versão relaxada do PM, que contém apenas algumas das restrições dos tipos (39) e (40).

Testa-se a solução  $(\underline{\eta}, \underline{x})$  a fim de se verificar se pertence ou não ao domínio do problema, sendo para isso resolvido o subproblema  $(P(\underline{x}))$  ou o seu dual. Acrescentam-se então novos cortes ou funções de suporte, até que se chegue a uma solução final que satisfaça o critério de convergência. Quer as funções de apoio, quer as de corte definem uma região linear côncava, consistindo cada programa "master" relaxado (PMR) da minimização sobre uma região linear côncava (ie: programação não-convexa), do tipo representado na figura 11. Os termos  $[f^T x + e]^+$  que representam disjunções, quer nas funções de apoio quer nas funções de corte, podem ser linearizados através das variáveis binárias  $\delta$ , de modo a transformar cada PMR num programa linear misto (com variáveis reais e 0-1). Tomando para limite inferior e superior da expressão dentro de parêntesis  $L$  e  $U$ , respectivamente,

$$L \leq f^T x + e \leq U \quad (42)$$

$$0 \leq \delta \leq 1 \quad (43)$$

e introduzindo a variável  $r$ , tem-se que  $[f^T x + e]^+$  é equivalente a:

$$r \leq \delta U \quad ; \quad r \leq (\delta - 1)L + (f^T x + e) \tag{44}$$

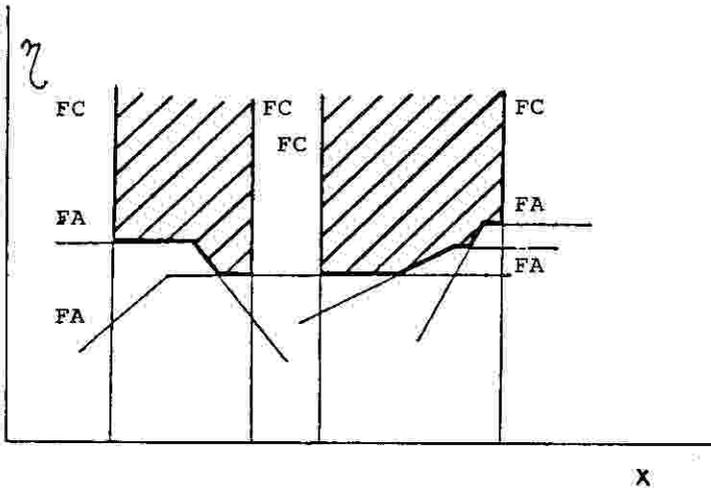


Fig. 11 - Interpretação física do domínio do programa "master".

Em alternativa a esta operação, que transforma o PMR num programa linear misto, pode reduzir-se este a um programa de complementaridade linear. Para isso introduzem-se duas variáveis  $r$  e  $q$  e a condição de complementaridade  $r^T q = 0$ , correspondentes a termo do tipo  $[f^T x + e]^+$ . Estas novas variáveis estão relacionadas do seguinte modo:

$$q = r - f^T x - e \tag{45}$$

Por este processo, reduz-se o número de restrições e evita-se a utilização de variáveis binárias. Para aplicar a extensão do método de Benders à optimização de treliças, é necessário que cada variável  $y_j$  esteja situada no semi-espço não-negativo, o que obriga a que  $x$  represente o vector das tensões.

Contudo, numa treliça, uma barra pode estar sujeita a compressões ou tracções, mas não a esses dois esforços em simultâneo. Restringe-se a variação das tensões em cada membro ao semi-espço que é não-positivo, se o membro estiver comprimido e positivo, se estiver traccionado. A expressão:

$$[u^T H x_j - c]_j^+ y_{jmax} \quad ; \quad 0 \leq y_j \leq y_{jmax} \tag{46}$$

pode ser transformada em:

$$[(u^T H x_j - c)_j (y_{j\min} + y_{j\max})]^+ ; y_{j\min} \leq y_j \leq y_{j\max} \quad (47)$$

onde  $x$  e  $y$  representam as variáveis de decisão e de estado, respectivamente, e se tem  $y_{j\min}$  ou  $y_{j\max}$  nulo.

#### 4.2. Algoritmo de convergência rápida para optimização de estruturas

Os esforços internos numa estrutura hiperstática são função das secções das peças. Uma treliça com comportamento elástico, possui como característica a invariância do vector das tensões/deslocamentos: se todas as áreas forem multiplicadas por um factor de escala  $\rho$  positivo e para que as forças dos membros permaneçam constantes, é necessário multiplicar as tensões nos membros por  $1/\rho$ ; dado que os deslocamentos nodais são uma combinação linear das tensões nos membros, seriam também multiplicados por  $1/\rho$ .

Tendo em atenção estas propriedades, é possível deduzir uma versão simplificada do algoritmo de decomposição de recursos. Fixando o conjunto  $x$  das áreas, é possível determinar de uma forma única o vector das tensões que lhe corresponderia por inversão das equações de equilíbrio. Se as tensões/deslocamentos forem todas inferiores aos seus valores máximos seria possível aumentá-las ( $0 < \rho < 1$ ), de modo a reduzir o volume da peça até que aquelas encontrem a fronteira do seu domínio. Caso contrário, isto é, se as tensões/deslocamentos forem superiores aos seus valores máximos (em valor absoluto), as variáveis de decisão teriam de ser multiplicadas por  $\rho > 1$  de modo a reduzir as tensões/deslocamentos resultantes serem admissíveis. Deste modo, o vector das áreas  $\rho x$  tem pelo menos um dos seus membros sujeito a uma tensão admissível extrema, ou um deslocamento máximo/mínimo.

Cada PMR vai fornecer um limite inferior, que em cada iteração se aproxima do resultado final. Por esse motivo, as variáveis de estado que correspondem às áreas obtidas pelos PMR estão fora do domínio do problema. Esse conjunto de áreas permite obter um limite inferior  $I^T x$  e a partir destes valores determinar um factor de escala  $\rho > 1$ . Isto acontece até que o valor de  $x$  obtido pelo PMR seja o óptimo global do problema.

Podem então calcular-se o raio dual associado ao vector  $x$ , multiplicando por  $\rho$  as variáveis duais que correspondem às equações bilineares de equilíbrio e mantendo constantes os multiplicadores ligados às restantes restrições lineares activas.

Destas considerações resulta o program (ARC):

$$\min \eta = \mathbf{1}^T \mathbf{x} \quad (48)$$

$$\text{sa} \quad -\sum_{i=1,m} u_i^e b_i + \sum_{j=1,n} \{ [\sum_{i=1,m} u_i^e (\mathbf{x}^T \mathbf{H}_i + f_i)_j (y_{j1} + y_{ju})] \}^+ \geq 0 \quad (49)$$

$$\mathbf{x} \in X \quad ; \quad e = 1, \dots, e_f \quad (50)$$

que se pode converter do mesmo modo, quer num programa linear misto (com variáveis reais e binárias), quer num programa de complementaridade linear.

#### 4.3. Aplicação ao cálculo de treliças cujas secções são escolhidas de entre um conjunto finito de peças

O algoritmo descrito no parágrafo anterior é facilmente convertível à resolução de problemas de síntese de estruturas em que as variáveis de decisão assumam valores discretos. Em cada iteração é resolvido um programa linear misto (com variáveis reais e binárias) (ARD).

A natureza de cada problema faz com que a solução de cada ARD multiplicada por  $\rho$  não constitua uma solução do problema inicial; por esse motivo não tem sentido desenvolver um critério de terminação que se baseie na aproximação dada pelo limite superior do problema. Deve-se contudo realçar que, pelo facto das secções das peças serem escolhidas de entre um conjunto finito de alternativas, se garante a convergência de ARD para o óptimo global num número finito de passos. Este programa é muito mais eficiente do que ARC, em que não é possível garantir que a convergência seja finita, em virtude de todas as variáveis relacionadas com o comportamento da estrutura serem contínuas.

A treliça de três barras que foi resolvida atrás, vai servir para mostrar como funciona este método de plano de corte. Se se considerar que os membros diferem entre si de um valor constante e o custo é proporcional à área das secções, torna-se apenas necessário introduzir a restrição que obriga a que  $a_1, a_2$  e  $a_3$  sejam variáveis inteiras.

Fazendo no programa "master" relaxado:

$$x_1 = a_1 ; x_2 = a_2 ; x_3 = a_3$$

representar as variáveis de decisão (áreas) e,

$$y_1 = d_1^1 ; y_2 = d_2^1 ; y_3 = d_1^2 ; y_4 = d_2^2$$

os deslocamentos correspondentes, tem-se:

## INICIALIZAÇÃO DO ALGORITMO:

Faz-se  $UB = +\infty$ ,  $LB = 3.828$  (que corresponde a  $a_1 = a_2 = a_3 = 1$ )

e  $\epsilon = 0.000$  (em virtude de se terminar com a solução exacta).

De  $d^k = K^{-1} \lambda^k$  ( $k = 1, 2$ ) tem-se:

$$d^1 = [40, -16.6]^T; d^2 = [-8.3, 20.]^T$$

Para determinar um conjunto de deslocamentos nodais pertencente ao domínio do problema, conclui-se que fazendo  $\rho = 5.657$  aumentaria as áreas até que o conjunto de tensões ficasse dentro dos limites permitidos no problema. Isto significa que o membro 1 teria uma tensão de tracção máxima quando se aplicava à estrutura a primeira condição de carregamento.

A restrição  $e_f = 1$  é construída, fazendo:

$$u^1 = [1, -0.414, 0, 0, \sqrt{2}, 0, 0, 0, 0, 0.]^T$$

$$x^T H_1 = [(\sqrt{2}/2x_1 + 1/2x_2), 1/2x_2, 0, 0.]$$

$$x^T H_2 = [1/2x_2, (\sqrt{2}/2x_3 + 1/2x_2), 0, 0.]$$

$$x^T H_3 = [0, 0, (\sqrt{2}/2x_1 + 1/2x_2), 1/2x_2]$$

$$x^T H_4 = [0, 0, 1/2x_2, (\sqrt{2}/2x_3 + 1/2x_2)]$$

$$f_5 = [-\sqrt{2}/2, 0, 0, 0.]$$

$$f_6 = [-\sqrt{2}/2, -\sqrt{2}/2, 0, 0.]$$

$$f_7 = [0, -\sqrt{2}/2, 0, 0.]$$

$$f_8 = [0, 0, -\sqrt{2}/2, 0.]$$

$$f_9 = [0, 0, -\sqrt{2}/2, -\sqrt{2}/2]$$

$$f_{10} = [0, 0, 0, -\sqrt{2}/2]$$

$$b^T = [40, 0, 0, 20, -5, -5, 5, 5, -5, -5.]$$

Chega-se à desigualdade que corta a solução inicial:

$$-40. + 5\sqrt{2} + [\sqrt{2}/2 x_1 + 1/2 x_2 - 0.414 \cdot 1/2 x_2 - \sqrt{2} \cdot \sqrt{2}/2]^+ 5\sqrt{2} \\ + [1/2 a_2 - 0.414 (\sqrt{2}/2 x_3 + 1/2 x_2)]^+ (-5\sqrt{2}) \geq 0$$

i.e.:

$$-32.929 + (5x_1 + 2.071x_2 - 7.071)^+ + (-2.071x_2 + 2.071x_3)^+ \geq 0$$

Linearizando a restrição, quer introduzindo variáveis binárias, quer por adição de variáveis reais e impondo a condição de complementaridade, a solução do programa "master" relaxado,  $\eta = LB$  é 13.728 :

$$x_1 = 8 \ ; \ x_2 = 1 \ ; \ x_3 = 1$$

Invertendo as relações de equilíbrio, obtém-se:

$$d^1 = [6.724 \ -2.784]^T \ ; \ d^2 = [-1.392 \ 17.148]^T$$

concluindo-se que  $\rho = 2.42$  , para mobilizar o limite superior do deslocamento  $d_2^2$  . O vector relativo aos multiplicadores de Lagrange para  $e_f = 2$  vai ser:

$$u = [0. \ 0. \ -0.07 \ 0.857 \ 0. \ 0. \ 0. \ 0. \ 0. \ 2.]^T$$

Sendo a nova restrição:

$$20.(-0.857) + 5\sqrt{2} + \{[-0.07(1/2x_2 + \sqrt{2}/2x_1) + 0.857 \cdot 1/2x_2] (-5\sqrt{2})\}^+ \\ + [-0.07 \cdot 1/2x_2 + 0.857(1/2 x_2 + \sqrt{2}/2 x_3) - 2\sqrt{2}] (-5\sqrt{2})^+ \geq 0$$

ou:

$$-10.069 + [0.35x_1 - 2.784x_2]^+ + [4.235x_3 - 2.784x_2 - 7.071]^+ \geq 0$$

O programa ARD tem por solução  $\eta = LB = 16.142$  e

$$x_1 = 7 \ ; \ x_2 = 2 \ ; \ x_3 = 3$$

o parâmetro  $\rho = 1.005$  , correspondente ao limite superior em  $d_1^1$  . Vai ser portanto acrescentada uma nova restrição ao ARD.

O algoritmo termina após a 4ª iteração (a que correspondem 4 planos de corte no ARD), para a qual  $\eta = LB = 16.728$  correspondendo a:

$$x_1 = 7 \ ; \ x_2 = 4 \ ; \ x_3 = 2$$

de modo que  $\rho = 0.996 < 1$  , donde  $UB = 16.728 = LB$  .

#### 4.4. Discussão

Para se resolver o problema que surge por relaxação do "master" em cada iteração, pode utilizar-se o método da decomposição de Benders convencional, que teria a vantagem de reduzir sensivelmente a metade o número de variáveis envolvidas em cada RMP. Além disso, por se resolver em cada iteração um programa interrelacionado com o RMP das iterações anteriores, é conveniente tirar partido dessas soluções. A introdução de novos cortes é feita introduzindo pequenas alterações no problema da iteração anterior e utilizando a capacidade de reoptimizar. Deste modo, reduz-se o tempo de computação em relação ao que seria necessário para resolver todos os RMP, independentemente uns dos outros.

#### 5. Conclusões

Muitas vezes, quando se resolve um problema de síntese de treliças com comportamento elástico linear e cujas secções podem variar de um modo contínuo, o resultado obtido utilizando um algoritmo de programação convexa coincide com o mínimo global. Por esse motivo as metodologias apresentadas neste artigo servem de técnicas para verificar os resultados obtidos. É possível ao utilizador controlar de um modo interactivo as aplicações destas técnicas que requerem mais tempo de computação (e espaço de memória) do que qualquer algoritmo que forneça mínimos locais.

A introdução de restrições que condicionam a escolha da secção a um conjunto discreto, o que é aliás a situação real no projecto, aumenta a competitividade destes métodos de programação não-convexa. Em qualquer destas estratégias é possível garantir a convergência para o mínimo global num número finito de iterações.

#### Referências

- [1] Bartholomew, P. (1979), "Dual Bound used for Monitoring Structural Optimization Problems", *Engineering Optimization*, **4**, 45-50.
- [2] Batt, J. R. and Gellin, S. (1985), "Rapid Reanalysis by the Force Method", *Comp. Meth. in Appl. Mech. and Engng.*, **53**, 105-117.
- [3] Belegundu, A. D. and Arora, J. S. (1985), "A Study of MP methods for Structural Optimization", *Int. J. Num. Meth. Engng.*, **21**, 1583-1623.
- [4] Cottle, R. W., Gianessi, F. and Lions, J. L. (1980), *Variational Inequalities and Complementarity Problems*, J. Wiley.
- [5] Fisher, M. (1981), "The Lagrangian Relaxation for solving IP Problems", *Management Science*, **20**, 1-18.

- [6] Fuchs, M. (1980), "Linearized Homogeneous constraints in structural design", *Int. J. Mech. Sci.*, 22, 33-40.
- [7] Geoffrion, A. M. and Graves, G. W. (1974), "Multicommodity Distribution Systems Design by Bender's Decomposition" *Management Science*, 20, 822-844.
- [8] Geoffrion, A. M. (1977), "How can Specialized Discrete and Convex Optimization Methods be married", *Annals of Discrete Mathematics*, 1, 205-220.
- [9] Imai, K. and Shoji, M. (1981), "Minimum Cost Design of Framed Structures by the Mini-max Dual Method", *Int. J. Num. Meth. Engng.*, 17, 213-229.
- [10] Kanarachos, A., Makris, P. and Koch, M. (1985), "Localization of Multi-constrained Optima and Avoidance of Local Optima in Structural Optimization Problems", *Comp. Meth. in Appl. Mech. and Engng.*, 51, 79-106.
- [11] Kaneko, I. and Maier, G. (1981), "Optimum Design of Plastic Structures under Displacement Constraints", *Comp. Meth. in Appl. Mech. and Engng.*, 27, 369-391.
- [12] Kovacs, L. B. (1980), "Combinatorial Methods of Discrete Programming", *Akademiai Kiadó, Budapest*.
- [13] McCormick, G.P. (1979), "Future directions in Mathematical Programming" in *NATO-ASI on Engineering Plasticity - University of Waterloo 1977*, Cohn, M. Z. and Maier, G. (Ed.), Pergamon Press.
- [14] Reeves, G. R. (1973), "Global Optimization in Nonconvex All Quadratic Programming", *Management Science* 17.
- [15] Saglam, M. R. (1981), "Computer Oriented Algorithms for Solving Problems with Discrete Programming Techniques" in *NATO-ASI on Optimization of Distributed Parameter Structures-Iowa City 1980*, Haug, E.J. and Cea, J. (Ed.), Sijthoff and Noordhoff.
- [16] Schmit, L.A. and Mallett, R. A. (1963), "Structural Synthesis and Design Parameter Hierarchy", *J. Struct. Div., Proc. ASCE*, 89, 269-299.
- [17] Simões, L.M.C. (1982), "Exhaustive Search for the Global Optimum of Least Volume Structures", Ph. D. Thesis, University of London.
- [18] Soland, R.M. (1971), "Algorithm for Separable Nonconvex Programming Problem: Nonconvex Constraints", *Management Science*, 17, 759-773.
- [19] Venkaya, V. B. (1978), "Structures Optimization: a Review and some Recommendations", *Int. J. Num. Meth. Engng.*, 13, 203-228.
- [20] Wolsey, L. A. (1980), "Heuristic Analysis, Linear Programming and Branch and Bound", *Math. Programming Study*, 13, 121-134.
- [21] Yates, D. F., Templeman, A. B. and Boffey, T. B. (1982), "The complexity of Procedures for Determining the Minimum Weight of Trusses with Discrete Members", *Int. J. Solids Struct.*, 18, 487-495.