SBA: Controle de Automação, Vol. 1, Nº 2, pp 154-161

OTIMIZAÇÃO NÃO DIFERENCIÁVEL NOS MÉTODOS DE DECOMPOSIÇÃO

Philippe Mahey
Deptº Eng. Elétrica
PUC/RJ

Resumo

A classe das funções-Max constitui um caso particular bem conhecido em otimização não diferenciável. Os métodos de decomposição em programação matemática geram funções deste tipo. Apresenta-se as principais propriedades dessas funções e os algoritmos de subgradientes para minimizá-las. Mostra-se como esses algoritmos podem ser implementados para acelerar a convergência dos métodos clássicos para decomposição de programas lineares.

Nonsmooth optimization for decomposition methods.

Abstract

The class of Max-functions is a well studied case in nonsmooth optimization. We present here general results for these funtions and for the subgradient methods. We show then how to use these techniques to accelerate the convergence of some classical decomposition methods in linear programming.

1. INTRODUÇÃO

A otimização não diferenciável atraiu a atenção dos pesquisadores em programação matemática quando Held e Karp em 1970 propuseram um algoritmo eficiente para solução do problema do caixeiro viajante. Este algoritmo usava explicitamente como direção de busca um subgradiente de uma função linear por partes, logo não sempre diferenciável. De fato, a não diferenciabilidade em foco era compensada pela subdiferenciabilidade das funções, ou seja os resultados teóricos da analise convexa podiam sustentar a construção de algoritmos. Deste ponto de vista, a escola de Kiev, liderada por Shor, Polyak , Ermolev e outros, já vinha desenvolvendo pesquisas teóricas e aplicadas desde os anos 60.

Os trabalhos de Clarke (1975) contribuiram a partir de 1975 para delimitar o domínio prático da otimização não diferenciável sem restringir ao caso exclusivamente convexo. Com efeito, todas as funções encontradas na prática são localmente Lipschitz, ou seja, funções $f:R^n \to R$ tais que, em qualquer subconjunto limitado S, satisfazem:

$$|f(x_1)-f(x_2)| \leq K \, \| x_1-x_2 \| \, , \; \forall x_1,x_2 \in S \tag{1}$$

Um velho teorema de Rademacher afirma que tais funções possuem um gradiente em quase toda parte. Isso levou Clarke a definir o gradiente generalizado de uma função localmente Lipschitz em x como:
a casca convexa de todos os limites dos gradientes de f calculados em sequências de pontos convergentes para x.

O gradiente generalizado é denotado $\partial f(x)$ e tem as seguintes propriedades:

i) É um conjunto convexo, compacto e não vazio. Logo, pode-se definir uma função suporte notada:

$$f^o(x;v) = \text{Max}\{v^T g, \; g \in \partial f(x)\} \tag{2}$$

Clarke a chama de derivada direcional generalizada.

ii) Se f é continuamente diferenciável em x então $\partial f(x)=\{\nabla f(x)\}$ o gradiente de f em x.

iii) Se f é convexa, então f é Lipschitz e $\partial f(x)$ é o subdiferencial de f em x, ou seja o conjunto de todos os subgradientes, os vetores $g \in R^n$ tais que:

$$f(y) \geq f(x) + g^T (y-x), \;\; \forall y$$

Na tentativa de buscar aplicações práticas, Womersley (1980) introduziu a subclasse das funções diferenciáveis por parte. Dentro dessa classe estão as chamadas funções Max , do tipo:

$$f(x) = \text{Max}\{h(x,y), \; y \in Y\} \tag{3}$$

onde Y é compacto e h é geralmente conjuntamente diferenciável em relação a x e y.

Pode-se verificar que uma função-Max é localmente Lipschitz e possui derivadas direcionais $f'(x;v)$ em todas as direções. Além disso $f'(x;v)=f^o(x;v)$ e f' é convexa em relação a v (propriedade de convexidade tangencial segundo Hiriart-Urruty, 1982).

Observamos que as funções Max contêm todas as funções convexas e s.c.i., as funções de penalidades não diferenciáveis ℓ_1 ou ℓ_∞ recentemente introduzidas em programação não linear restrita e todas as funções duais associadas a técnicas de relaxação lagrangeana.

Finalmente, pode-se observar também que a dificuldade de minimizar uma função Max é equivalente a resolver o seguinte problema de programação não linear restrita:

Minimizar v

$$v \geq h(x,y), \quad \forall y \in Y \tag{4}$$

O conjunto Y é muitas vezes simples ou discreto. A dificuldade de resolver (4) vem do grande número de restrições, o que exige o uso de métodos enumerativos ou de geração de linhas.

Nosso propósito neste artigo é, na impossibilidade de retratar todos os aspectos desta relativamente nova área de pesquisa, para os quais o leitor pode se referir às monografias de Lemaréchal (1982), Shor (1985) e Kiwiel (1985), enfatizar o uso das técnicas de subgradientes nos modelos de decomposição de programas lineares de grande porte.

2. OS ALGORITMOS DE SUBGRADIENTES

2.1. A direção de maior descida

Começamos com alguns resultados teóricos que mostrarão as dificuldades encontradas para desenhar algoritmos de descida para a minimização de uma função do tipo (3).

Nota-se Y_x o subconjunto não vazio de Y dos y que minimizam $h(x,y)$ para um dado x (suponha-se $x \in R^n$ e $y \in R^m$)

O primeiro resultado clássico é:

$$\partial f(x)=\text{conv } \{\nabla_x h(x,y),\ y \in Y_x\} \tag{5}$$

A direção de maior descida em x é oposta ao vetor de norma mínima em $\partial f(x)$. Com efeito, a derivada direcional de f em x na direção v é dada por:

$$f'(x;v) = \text{Max } \{g^T v; g \in \partial f(x)\}$$

Logo:

$$\text{Min } f'(x;v) = \underset{\|v\| \leq 1}{\text{Min}} \ \underset{g \in \partial f(x)}{\text{Max}} \ g^T v$$

$$= \underset{g \in \partial f(x)}{\text{Max}} \ \underset{\|v\| \leq 1}{\text{Min}} \ g^T v = \underset{g \in \partial f(x)}{\text{Max}} \ (-\|g\|) =$$

$$= -\underset{g \in \partial f(x)}{\text{Min}} \ \|g\| = -\|\tilde{g}\| \tag{6}$$

Observa-se ainda que qualquer direção de descida satisfaz $f'(x;v)<0$, logo satisfaz:

$$g^T v < 0, \ \forall g \in \partial f(x) \tag{7}$$

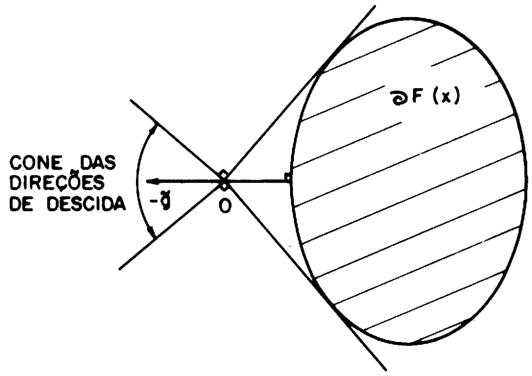

FIG. 1

Logo, uma condição necessária de otimalidade é que não existem direções de descida, ou seja:

$$\tilde{g} = 0, \text{ isto é } 0 \in \partial f(x) \tag{8}$$

Wolfe (1975) foi um dos primeiros a mostrar exemplos de funções-Max para as quais um algoritmo do tipo maior descida efetuando buscas unidimensionais exatas não converge. Além disso, mesmo nos casos favoráveis, o cálculo de $\tilde{g}$ pode ser muito caro nos pontos de não diferenciabilidade ("bicos") da função. Lemaréchal observa que, na prática, a função é sempre diferenciável em pontos possivelmente muito próximos de bicos. Isto significa que tem-se acesso a um subgradiente apenas e não ao conjunto $\partial f(x)$ inteiro. A abordagem de Lemaréchal (cf. (1978) por exemplo) é ampliar o subdiferencial para incluir os subgradientes da função nos pontos calculados anteriormente até uma certa memória (cadeia) gerando "feixes" de gradientes. Estes métodos geram algoritmos de descida, mas ainda não foram plenamente testados nos modelos de decomposição. Passaremos agora a descrever os métodos ditos de subgradientes. A característica principal destes últimos é que a função de descida associada não é a função objetiva, mas a função distância a solução ótima.

2.2. Convergência teórica

A seguir, f é uma função convexa própria no R^n. O algoritmo de subgradiente supõe que disponhamos de um subgradiente da função f no ponto iterado x_k, $g_k \in \partial f(x_k)$, e:

$$x_{k+1} = x_k - \alpha_k \, g_k / \|g_k\| \tag{9}$$

Na figura 2 são representadas as curvas de nível da função no $R^2 (x=[\xi_1\ \xi_2]')$:

$$f(\xi_1,\xi_2)= \max\{|\xi_1+3\xi_2|,|\xi_1-3\xi_2|\}$$

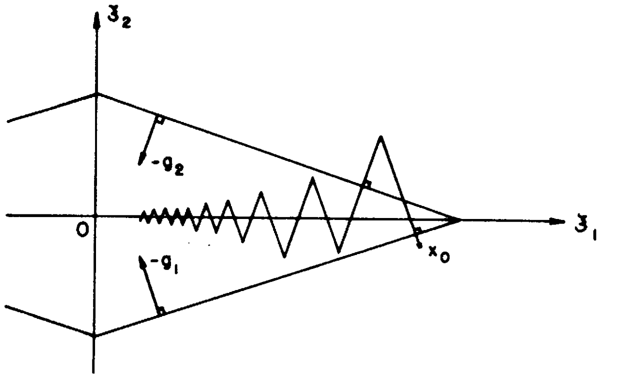

FIG. 2

Constata-se que no eixo $(0\xi_1)$ a função f não é diferenciável e que as direções opostas aos subgradientes extremos $-g_1$ e $-g_2$ (que são de fato as direções acessíveis na prática) não são direções de descida. Toda a dificuldade de implementação do algoritmo reside na escolha do passo α_k já que nenhuma busca unidimensional pode servir. Polyak (1969) mostrou que o passo α_k deve tender para 0, mas não muito rápido.

Teorema 1 (Polyak, 1969)

Supondo que o conjunto dos pontos minimizando f, X^*, é não vazio e limitado, e que:

$$\alpha_k \to 0 \quad \sum_{k=0}^{\infty} \alpha_k = +\infty \tag{10}$$

então para qualquer ponto inicial x_0, existe uma subsequência de $\{x_k\}$, x_{k+1} dado por (9), que converge para um $x^* \in X^*$.

Prova: Ver Polyak para os detalhes técni - cos. Observa-se apenas que como:

$$\|x_0 - x_k\| \leq \|x_0 - x_1\| +\ldots+ \|x_{k-1} - x_k\| =$$

$$= t_0 + t_1+\ldots+ t_{k-1}$$

a condição $\Sigma t_k=+\infty$ significa que pode-se escolher um ponto x_0 tão afastado de x^* quanto quiser.

A condição $t_k \to 0$ é necessária porque, no caso não diferenciável, a norma do gradiente não vai para zero em uma sequência con - vergente para x^*.

A condição (10), embora muito simples e de implementação imediata, sugere uma taxa de convergência muito lenta, aliás sublinear.

Teorema 2

Seja $d(x)= \frac{1}{2} \|x - x^*\|^2$ e f convexa e finita em $x \neq x^*$. Então qualquer direção $v=-g$, onde $g \in \partial f(x)$, é uma direção de descida para d.

Prova: Seja $g \in \partial f(x)$. Então

$$f(x^*) \geq f(x) + g^T(x^* - x)$$

Como $f(x^*)-f(x)<0$ e $x^*-x=-\nabla d(x)$, temos

$$0 > - g^T \nabla d(x)$$

Observa-se que o angulo θ entre v e $x^* - x$ (Fig.3) é sempre menor que 90^o e quando é próximo deste valor,

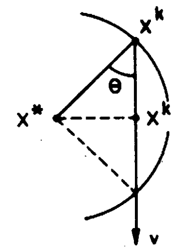

FIG. 3

o passo correspondente é menor. Seguindo esta idéia, Goffin (1977) define a condição da função f:

$$\gamma = \inf_{x\neq x^*} \{\min_{g\in\partial f(x)} \frac{g^T(x - x^*)}{\|g\|\ \|x^*-x\|}\} \tag{11}$$

ou seja, é o cosine do maior angulo θ en - contrado. Logo, para garantir a descida da função d, o passo α_k deve satisfazer:

$$\alpha_k < 2\gamma\ \|x^* - x_k\| \tag{12}$$

2.3. Convergência linear

Observamos que (12) depende do conhecimento de γ e da distância a solução ótima , parâmetros estes que podem geralmente ser aproximados por limites apenas muito grosseiros.

Shor propõe para o passo α_k uma série convergente:

$$\alpha_k = \alpha_0\ \rho^k \tag{13}$$

onde α_0 aproxima (12) no meio do arco:

$$\alpha_0 \sim \gamma\ \|x^* - x_0\| \tag{14}$$

e ρ aproxima $\frac{\|x^* - x_{k+1}\|}{\|x^* - x_k\|}$ ou seja:

$$\rho \sim \sqrt{1-\gamma^2} \tag{15}$$

Goffin (1977) observa que, quando $\gamma > \frac{\sqrt{2}}{2}$ $(\theta < \frac{\pi}{4})$, a taxa de convergência ρ

pode ser aproximada por $\frac{1}{2\gamma}$ no lugar de (15) acelerando a convergência. Infelizmente, a prática mostra que este caso é raro e por outro lado, ele implica que qualquer $v=-g$, e $g \in \partial f(x)$, é uma direção de descida para f. Em resumo, o algoritmo do subgradiente (9) - (13) é útil quando as direções de descida são difíceis de ser computadas ($\theta \geq \frac{\pi}{4}$) e gera uma taxa de convergência

$\rho \geq \sqrt{1-\frac{1}{2}} = 0.7$, ou seja ainda bastante lenta. Os testes com os modelos de decomposição em programação linear mostraram condições menores do que 0.5 levando a taxas de convergência entre 0.87 e 0.93.

Held, Wolfe e Crowder (1974) mostraram que, usando a convexidade, pode-se estimar a condição de f:

$$g \in \partial f(x) \Rightarrow f(x^*) \geq f(x) + g^T(x^*-x)$$

Logo:

$$\gamma \sim \frac{-f(x^*) + f(x)}{\|g\| \; \|x^* - x\|} \tag{16}$$

É então natural de escolher α_k usando:

$$\alpha_k = t_k \frac{f(x) - f(x^*)}{\|g_k\|} \tag{17}$$

e (12) implica que t_k satisfaz

$$0 < t_k < 2$$

Infelizmente, não se conhecendo $f(x^*)$, deve-se usar uma aproximação $\bar{f}$ que, em geral, é um limite inferior de $f(x^*)$ (se f é uma função do tipo dual, seria o valor de uma solução primal viável obtida anteriormente). Neste caso a convergência teórica pode ser problemática e Held, Wolfe e Crowder sugerem diminuir t_k para zero. Eles até propõem dividir t_k por 2 a cada 5 iterações, o que corresponde a uma taxa de convergência $\rho = (\frac{1}{2})^{1/5} = .87$, equivalente à série convergente de Shor quando $\gamma = .5$.

Goffin mostrou que o algoritmo de subgradiente (9)-(17) é equivalente ao método de relaxação para sistemas de equações lineares. A fórmula (17) é que foi a mais escolhida nas aplicações em otimização combinatória (ver, por exemplo, Mahey (1985)).

2.4. Caso restrito

Completamos a apresentação dos métodos de subgradientes introduzindo restrições. O problema é formulado como:

$$\text{Minimizar } f(x)$$

$$x \in \Omega \tag{18}$$

e Ω é geralmente um conjunto convexo simples como veremos nas aplicações aos modelos de decomposição.

A tendência natural é então projetar o iterado (9) sobre Ω a cada iteração. No entanto, o passo verdadeiro $\|x_{k+1}-x_k\|$ pode se tornar muito pequeno após a projeção o que leva a seguinte modificação: o problema (18) pode ser escrito:

$$\text{Minimizar } F(x)=f(x) + \chi_\Omega(x)$$

onde $\chi_\Omega(x)$ é a função indicadora de Ω ($\chi_\Omega(x)=0$ se $x \in \Omega$ e $=+\infty$ senão).

Tem-se:

$$\partial F(x) = \partial f(x) + N_\Omega(x) \tag{19}$$

onde $N_\Omega(x)$ é o cone das normais a Ω em x.

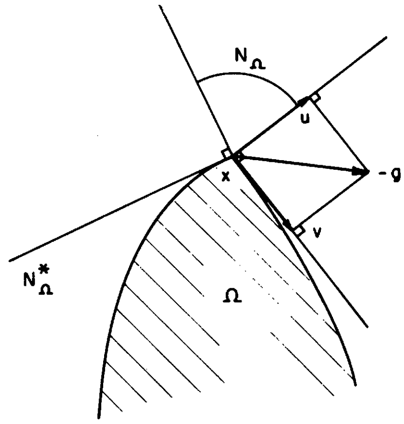

FIG. 4

Na fig 4, obtém-se:

$$\forall g \in \partial f(x), \quad -g=u+v$$

onde $u \in N_\Omega(x)$ e $v \in N^*_\Omega(x)$ (o cone polar de N_Ω ou cone das direções viáveis)

Logo $-v \in \partial F(x)$ e o algoritmo do subgradiente projetado deve ser:

$$x_{k+1} = x_k + \alpha_k \frac{v}{\|v\|} \tag{20}$$

onde

$$v = \mathrm{Proj}/_{N^*_\Omega}(-g) \text{ para um } g \in \partial f(x)$$

O algoritmo (20) significa que a escolha do passo, segundo uma estratégia do tipo (13) ou (17), deve ser feita na direção do subgradiente projetado e não na direção do subgradiente seguido da projeção do iterado.

3. APLICAÇÃO À DECOMPOSIÇÃO DE PROGRAMAS LINEARES

Consideremos agora um modelo linear com uma estrutura bloco-angular:

$$\text{Minimizar } \sum_{i=1}^{n} c_i^T x_i$$

sujeito a:

$$\sum_{i=1}^{n} A_i x_i \leq b_o$$

$$x_i \in S_i = \{x_i \in R^{n_i} | B_i x_i = b_i, x_i \geq 0\}$$

$$i=1,\dots,n \qquad (21)$$

onde: $x_i \in R^{n_i}$

$c_i \in R^{n_i}$

A_i $(m_o \times n_i)$

B_i $(m_i \times n_i)$

$b_o \in R^{m_o}$

$b_i \in R^{m_i}$

Hipóteses:

- Os poliedros S_i, $i=1,\dots,n$, são limitados
- A solução ótima de (21), notada $x^*=[x_1^* \dots x_n^*]^T$, é única e não degenerada. Notaremos p^* o vetor dos multiplicadores ótimos associados as m_o restrições de acoplamento ($p^* \in (R^{m_o})^+$).

3.1. Decomposição pelos preços

Esta técnica de decomposição consiste na separação do problema em n subproblemas cujos critérios são modificados por um vetor de "preços" associados às restrições de acoplamento (recursos comuns), vetor este ajustado iterativamente por um nível superior de coordenação até obter o equilíbrio entre a oferta e a demanda (destes recursos). Em programação matemática, é um método dual associado à relaxação lagrangeana das restrições de acoplamento.

O Lagrangeano é definido sobre $R^{n_1} \times R^{n_2} \times \dots \times R^{n_n} \times (R^{m_o})^+$

$$L(x,p) = \sum_{i=1}^{n} c_i^T x_i + p^T (\sum_{i=1}^{n} A_i x_i - b_o)$$

$$= \sum_{i=1}^{n} (c_i^T + p^T A_i) x_i - p^T b_o$$

Para um vetor de preços $p \geq 0$ dado, decompõe-se o problema nos n subproblemas:

$$h_i(p) = \min_{x_i \in S_i} (c_i^T + p^T A_i) x_i \qquad (22)$$

Seja $X_i(p)$ o conjunto das soluções ótimas de (22). O nível coordenador deve então arbitrar as soluções propostas pelos decisores locais resolvendo o problema dual:

$$\text{Minimizar} \quad h(p) = \sum_{i=1}^{n} h_i(p) - p^T b_o \qquad (22)$$

sujeito a $p \geq 0$

Verifica-se que -h é uma função-Max e aplicando (5) obtém-se:

$$\partial h(p) = \{g \in R^{m_o} | g = \sum_{i=1}^{n} A_i x_i - b_o, x_i \in X_i(p)\}$$

$$(24)$$

(23) é então um problema de otimização não diferencial restrito e pode-se resolvê-lo por um método de subgradiente do tipo (20). No entanto existem algoritmos que resolvem (21) em um número finito de passos. Tipicamente, é o algoritmo de Dantzig-Wolfe(1960) ou na versão dual equivalente, o algoritmo do "cutting-plane" de Kelley (1960). Este último resolve a versão transformada de (23) (cf.(4)):

$$v^t = \text{Max } v$$

$$v \leq h(p^\ell) + g_\ell^T (p-p^\ell), \ \ell=1,\dots,t \qquad (25)$$

$$p \geq 0$$

onde $g_\ell \in \partial h(p^\ell)$

As restrições são geradas iterativamente, ou seja p^{t+1} é a solução de (25) e gera uma nova restrição resolvendo os subproblemas (22) para $p=p^{t+1}$ para obter g_{t+1}.

Este algoritmo já foi bastante analizado e criticado (ver,por exemplo, Dirickx e Jennergren,1979) e as suas principais características são:

i) Convergência sublinear

ii) Monotonia da sequência $\{v^t\}$

iii)Não há monotonia da sequência$\{\|p^t - p^*\|\}$.

Essas propriedades chamam os comentários seguintes: por convergência sublinear entende-se a taxa de convergência de uma sequência gerada pelo algoritmo aplicado a uma função convexa geral. Se a convergência é finita como no caso presente, isso implica que o número de iterações pode se tornar muito elevado nas aplicações práticas que são de grande porte. As propriedades i) e ii) são complementares das propriedades observadas no algoritmo de subgradiente.

Na Tab.5 são representados quatro gráficos mostrando a convergência das sequências de custo e da distância ótima na comparação dos algoritmos do subgradiente (9-13) e Dantzig-Wolfe (25).

Obs: O modelo utilizado para a comparação é um modelo de planejamento da produção de uma oficina com vários ítens e vários meios de produção interligados sobre um horizonte fixo. O problema global (21) tem 336 variáveis e 216 restrições, das quais 48 são de acoplamento. O número de subproblemas é igual a 7. Cada iteração representa a resolução dos 7 subproblemas e também no caso de Dantzig-Wolfe a resolução do programa-mestre (25).

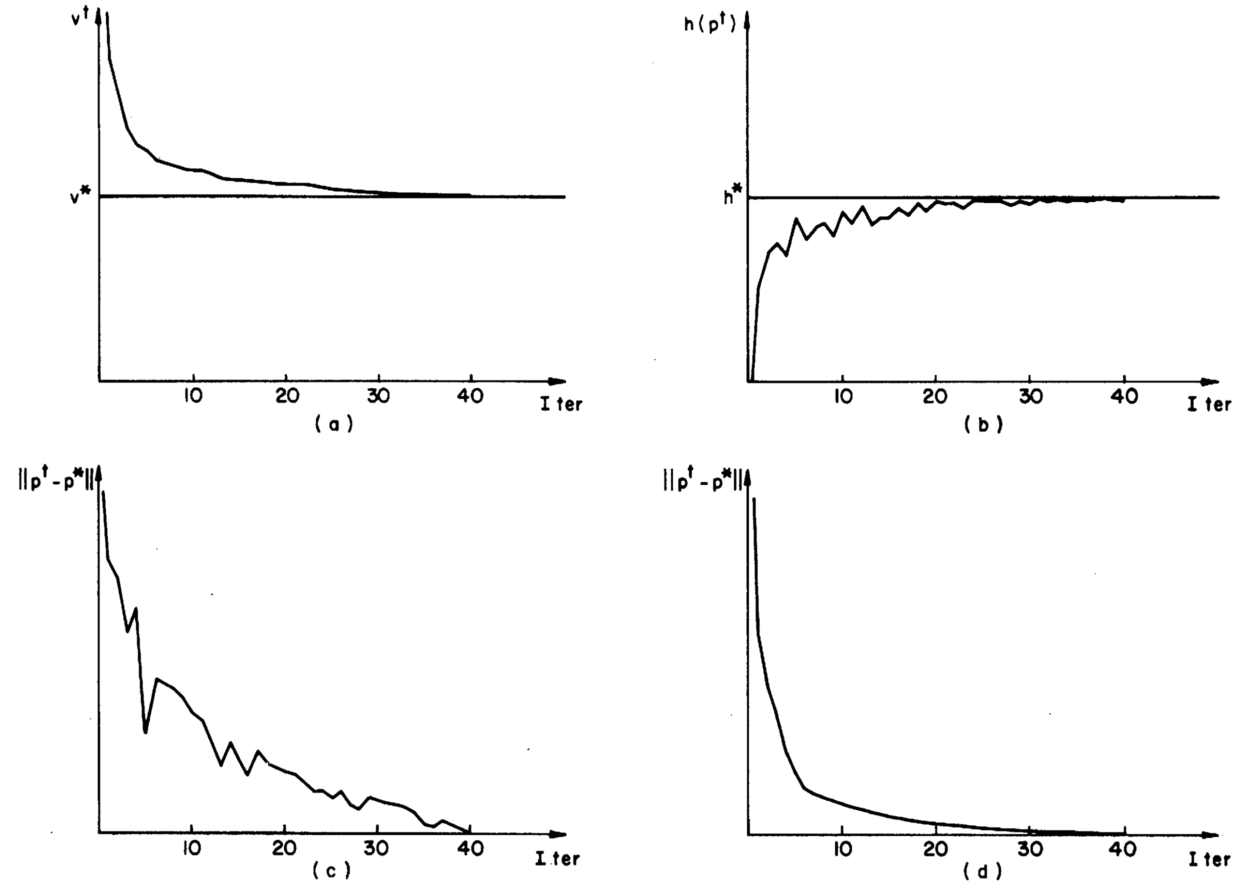

TABELA 5

VALOR ÓTIMO	(a) e (b)
DISTÂNCIA ÓTIMA	(c) e (d)
DANTZIG-WOLFE	(a) e (c)
SUBGRADIENTE	(b) e (d)

3.2. Um Algoritmo composto

As propriedades complementares dos algoritmos de Dantzig-Wolfe e do subgradiente sugerem que eles podem ser combinados para melhorar os seus desempenhos respectivos que, como vimos, são fracos.

O problema central recai finalmente sempre sobre o fato de que, para encontrar a solução ótima, é preciso conhecer todos os subgradientes ativos neste ponto, ou seja, o cume da função poliedral só é encontrado conhecendo todas as faces adjacentes. A instabilidade das soluções p^{t+1} no algoritmo de Dantzig-Wolfe torna essa busca muito lenta. Por outro lado, as oscilações com passos cada vez menores do método do subgradiente tendem a fornecer os subgradientes desejados, mas o algoritmo é incapaz de terminar na solução ótima.

Outro ponto de importância é o custo das iterações: o método do subgradiente é sempre mais barato porque não há programa-mestre a otimizar.

Essas deduções motivaram a construção de um algoritmo composto da seguinte maneira:

1. Fase 1: Algoritmo de subgradiente durante T_1 iterações. Esta fase serve para aproximar a solução ótima p* a baixo custo.

2. Fase 2: O Algoritmo de subgradiente é modificado de maneira a armazenar os subgradientes gerados, parando na iteração T_2.

3. Fase 3: Monta-se o programa-mestre de Dantzig-Wolfe com os subgradientes gerados na fase 2 e prosegue-se com este algoritmo até a solução ótima.

Os resultados relatados em Mahey(1986a) mostram uma nítida aceleração do método de Dantzig-Wolfe. Inclusive, se T_1 é suficientemente grande ($T_1 = 2m_o$ e $T_2 = 3m_o$), é comum observar que a fase 3 é reduzida a uma única iteração, o que significa que todo o subdiferencial na solução ótima foi gerado na fase 2.

3.3. Outros modelos de decomposição

No método de decomposição pelas quotas

é efetuada uma alocação a priori dos recursos comuns b_o aos subproblemas sob a forma de vetores $u_i \in R^{m_i}$ tais que $\sum_{i=1}^{n} u_i \leq b_o$.

Cada subproblema resolve então:

$$\text{Minimizar } c_i^T x_i$$

$$A_i x_i = u_i \qquad (26)$$

$$x_i \in S_i$$

Se $v_i(u_i)$ é o custo ótimo de (26) e $\Pi_i(u_i)$ é o conjunto dos multiplicadores ótimos, então (21) equivale a:

$$\text{Minimizar } v(u) = \sum_{i=1}^{n} v_i(u_i)$$

$$\sum_{i=1}^{n} u_i = b_o \qquad (27)$$

v(u) é uma função-Max (escreva,por exemplo, o dual de (26)) e:

$$\partial v(u) = \{ (\pi_1 \ldots \pi_n) | -\pi_i \in \Pi_i(u_i) \} \qquad (28)$$

Obtém-se de novo um problema de otimização não diferenciável restrito (Ω é aqui uma variedade linear).

Em Mahey (1982), a decomposição dual (25) é combinada com a decomposição primal (27). Os passos nos espaços duais e primais são escolhidos segundo a estratégia (17). Os valores dos custos dual e primal são usados como aproximações do custo ótimo em (17) e a redução do "gap" entre os dois valores fornece um critério de parada adequado. Esta idéia é explorada também em Sen e Sherali (1986).

Em Mahey (1986b), o método do subgradiente é usado como fase 1 de um algoritmo de decomposição misto onde as não diferenciabilidades encontradas (soluções não únicas em (22)) são eliminadas pela alocação de recursos no subproblema. No fim, cada subproblema recebe preços e recursos e o nível de coordenação (programa-mestre) é eliminado garantindo a descentralização do processo de decisão.

4. CONCLUSÃO

As técnicas de subgradientes fornecem algoritmos de baixo custo e bem adaptados aos modelos de decomposição em programação linear. A convergência não depende da qualidade das informações locais usadas como direções de busca mas de parâmetros associados à geometria da função como a condição. Os estudos de J.L. Goffin (1977 e 1980) são fundamentais e precisam ser aprofundados apesar da aparente dificuldade de aproximar corretamente essas informações globais. Se isso fosse possível, teríamos algoritmos talvez lentos (taxas ligeiramente inferiores a 1), mas cujo número de iterações para obter uma dada redução do passo não depende da dimensão do problema.

A convergência finita pode ser obtida em seguida pelos métodos clássicos à partir da solução dada pelo método do subgradiente.

REFERÊNCIAS BIBLIOGRÁFICAS

Clarke F.H. (1975). "Generalized gradients and applications", Trans. Amer. Math. Soc., 205, pp . 247-262.

Dantzig G.B. & P. Wolfe (1960). "The decomposition algorithm for linear programs", Econometrica, 29,4, pp. 767-778.

Dirickx Y. & L.P. Jennergren (1979). System Analysis by Multilevel Methods,J.Wiley.

Goffin J.L. (1977). "On convergence rates of subgradient optimization methods" , Math. Prog., 13, pp. 329-347.

Goffin J.L. (1980). "The relaxation method for solving systems of linear inequalities". Math. Oper. Res.,5, pp. 388-414.

Held M. & R.M. Karp (1971). "The traveling-salesman problem and minimum spanning trees". Math. Prog., 1, pp. 6-25.

Held M.,P. Wolfe & H.P. Crowder (1984) . "Validation of subgradient optimization", Math. Prog., 6, pp. 62-88.

Hiriart-Urruty J.B. (1982). "Approximating a second-order directional derivative for nonsmooth convex functions", SIAM J. Control and Optimization, 20,6, pp. 783-807.

Kelley J.E. (1960). "The cutting plane method for solving convex programs" , J. SIAM, 8, pp. 703-712.

Kiwiel K.C. (1985). Methods of Descent for Nondifferentiable Optimization, Lecture Notes in Mathematics, 1133, Springer V.

Lemaréchal C. (1978). "Bundle methods", in Non Smooth Optimization, C. Lemaréchal & R. Mifflin, eds., Proc. IIASA, Pergamon Press.

Lemaréchal C. (1982). Basic Theory in Nondifferentiable Optimization, Rap. de Recherche nº 181, INRIA.

Mahey P. (1982). "Decomposition of large-scale linear programs by subgradient optimization", Mat. Aplic. Comp. 1, 2 , pp. 121-134.

Mahey P. (1985). "Subgradient techniques and combinatorial optimization", W.P. rep. nº 85397, Univ. Bonn, RFA.

Mahey P. (1986a). "A subgradient algorithm for accelerating the Dantzig-Wolfe decomposition method", Methods of Oper. Res., 53, pp. 697-707.

Mahey P. (1986b). "Methodes de décomposition et décentralisation en programmation linéaire", RAIRO-Recherche Opérationnelle, 20,4.

Polyak B.T. (1969). "Minimization of unsmooth functionals", USSR Compt. Math. and Math. Physics, 9, pp. 14-29.

Sen S. & H.D. Sherali (1986). "A class of convergent primal dual subgradient algorithms for decomposable convex programs", Math. Prog., 35,3, pp. 279-297.

Shor N.Z. (1985). Minimization Methods for Nondifferentiable Functions, Springer V.

Wolfe P. (1975). "A method of conjugate sub gradients for minimizing nondiffentia - ble convex functions", Math. Prog. Study 3, pp. 145-173.

Womersley R.S. (1982). "Optimality condi tions for piecewise smooth funtions", Math. Prog. Study 17, pp. 13-27.