1. Publicação nº	2. Versão	3. Data	5. Distribuição
INPE-3897-PRE/939		*May.,1986*	☐ Interna ☒ Externa
4. Origem *DIN*	Programa *POPES*		☐ Restrita

6. Palavras chaves - selecionadas pelo(s) autor(es)

MÉTODOS NUMÉRICOS EM OTIMIZAÇÃO

DUALIDADE EM PROGRAMAÇÃO INTEIRA

FUNÇÕES PENALIDADES EXATAS

7. C.D.U.: *519.852*

8. Título *INPE-3897-PRE/939*

USO DE UMA FUNÇÃO PENALIDADE EXATA PARA ELIMINAÇÃO DO "GAP" DE DUALIDADE EM PROBLEMAS DE PROGRAMAÇÃO LINEAR INTEIRA ZERO-UM

10. Páginas: *09*

11. Última página: *08*

12. Revisada por

Horacio Hideki Yanasse

9. Autoria *Luiz Antonio Nogueira Lorena*
Acioli Antonio de Olivo

Assinatura responsável

13. Autorizada por

Marco Antonio Raupp
Diretor Geral

14. Resumo/Notas

Recentemente Barcia apresentou um algoritmo para eliminar o "gap" de dualidade entre um problema de programação linear inteira zero-um e o seu problema dual. O algoritmo define uma sequência de cortes duais que proporcionam "bounds" que convergem em um número finito de passos para o valor ótimo do problema. O objetivo desse trabalho é mostrar que se podem usar os cortes duais em um método de penalidades onde se espera encontrar uma solução aproximada em tempos computacionais reduzidos.

15. Observações *Trabalho a ser apresentado no III CLAIO (Congresso Latino-Iberoamericano de Investigacion Operativa e Ingenieria de Sistemas) de 18 a 22 de agosto de 1986 - Santiago-Chile.*

USO DE UMA FUNÇÃO PENALIDADE EXATA PARA ELIMINAÇÃO DO "GAP" DE DUALIDADE EM PROBLEMAS DE PROGRAMAÇÃO LINEAR INTEIRA ZERO-UM

Luiz Antonio Nogueira Lorena
Acioli Antonio de Olivo

Ministério da Ciência e Tecnologia - MCT
Instituto de Pesquisas Espaciais - INPE
Caixa Postal 515 - 12201 - São José dos Campos - SP

ABSTRACT

An algorithm to eliminate the duality gap between a binary integer linear programming and its dual has been published by Barcia. The algorithm defines a sequence of dual cuts that provide bounds converging to the optimal value of the problem in a finite number of steps. In this work it is shown that is possible to use the dual cuts in a penalty method where we can expect to find an approximate solution within reduced computational times.

1. INTRODUÇÃO

Neste trabalho é apresentado um método de penalidades aplicado a problemas de programação linear inteira zero-um. O objetivo principal é cobrir o "gap" de dualidade característico destes tipos de problemas.

O método baseia-se nos trabalhos de Barcia [1] e [2] que usa cortes que atuam nos subproblemas lagrangenos do dual do problema de programação linear inteira zero-um. Faz-se a otimização do dual usando o método de subgradientes.

Na abordagem aqui usada propõe-se utilizar os cortes do problema lagrangeano com o enfoque de penalidade exata, onde não é necessário otimizar o dual.

O trabalho está organizado da seguinte forma: na seção 2 apresenta-se a teoria que justifica o presente método de penalidades; na seção 3 apresentam-se os testes computacionais; na seção 4 apresentam-se as conclusões com comentários gerais sobre o método.

2. FUNDAMENTOS TEÓRICOS

Parte da teoria a ser exposta apareceu em Lorena [4] e Lorena e Oliveira [5]. O problema de programação linear inteira é escrito como

$$(P) \qquad v = \max \quad cx \qquad \text{sujeito a} \quad Ax \leq b, \quad x \in \{0,1\}^n,$$

onde $c \in R^n$, $c \geq 0$, $b \in R^m$ e A é uma matriz real mxn.

Seu problema dual é

$$(D) \qquad w = \min_{\substack{\lambda \geq 0 \\ \lambda \in R^m}} \quad \max_{x \in \{0,1\}^n} \{cx - \lambda(Ax-b)\}$$

A função perturbação de (P), $\phi: R^m \to \overline{R}$ ($\overline{R} = R \cup \{\pm\infty\}$), é definida por:

$$\phi(d) = \sup_{x \in \{0,1\}^n} \{cx : Ax \leq d\}$$

Define-se a seguir um conjunto de funções que será usado para estabelecer um resultado de dualidade forte entre (P) e (D).

O conjunto de funções

$$\partial_H^b(\phi) = \{h \in H : h(d) - h(b) \geq \phi(d) - \phi(b), \ \forall \, d \in R^m\}$$

é chamado H-superdiferencial de ϕ em b. H é um conjunto de funções reais finitas definidas em R^m, as quais são "não-decrescentes", isto é, se $h \in H$, $h(d^1) \geq h(d^2)$, $\forall \, d^1, d^2 \in R^m$ tais que $d_i^1 \geq d_i^2$, $i=1,\dots,m$. Além disto, H possui a propriedade de translação, isto é, se $h \in H$, então $h' \in H$, onde $h'(d) = h(d) + r$, $\forall \, d \in R^m$, $r \in R$.

O seguinte lema relaciona funções de H com a função perturbação ϕ através dos problemas (P) e (D).

Lema 1: $\forall\, h \in H$, $\sup\limits_{x \in \{0,1\}^n} \{cx - h(Ax)\} =$

$$\sup_{d \in R^m} \{\phi(d) - h(d)\}.$$

Prova: $\sup\limits_{x \in \{0,1\}^n} \{cx - h(Ax)\} = \sup\limits_{x \in \{0,1\}^n} \{cx - \inf\limits_{\substack{d \\ Ax \leqq d}} h(d)\} =$

$$\sup_{d \in R^m} \{\sup_{\substack{x \in \{0,1\}^n \\ Ax \leqq d}} cx - h(d)\} = \sup_{d \in R^m} \{\phi(d) - h(d)\}$$

A proposição a seguir estabelece um resultado de dualidade forte entre (P) e (D).

Proposição 1: $\sup\limits_{x \in \{0,1\}^n} \{cx - h(Ax)\} + h(b) = \phi(b)$
se e só se $h \in \partial_H^b(\phi)$.

Prova: Pelo lema 1,

$$\sup_{x \in \{0,1\}^n} \{cx - h(Ax)\} + h(b) = \sup_{d \in R^m} \{\phi(d) - h(d)\} + h(b)$$

Assim, se $\sup\limits_{d \in R^m} \{\phi(d) - h(d)\} + h(b) = \phi(b)$,

$$\phi(d) - h(d) \leqq \phi(b) - h(b), \ \forall\, d \in R^m$$

$$h(d) - h(b) \geqq \phi(d) - \phi(b), \ \forall\, d \in R^m,$$

ou seja, $h \in \partial_H^b(\phi)$.

Fazendo o caminho inverso chega-se a:

$$\text{se } h \varepsilon \partial_H^b(\phi), \text{ então } \sup_{x\varepsilon\{0,1\}^n} \{cx-h(Ax)\}+h(b)=\phi(b).$$

Corolário 1: Se existir ao menos uma função h^*,

$h^* \varepsilon \partial_H^b(\phi)$, então $v = w$.

Na verdade, a operação de ínfimo na definição do dual (D) é supérflua para os resultados da proposição 1. Assim, pode-se definir o problema com penalidades

$$(PP) \qquad p_h = \max_{h\varepsilon\{0,1\}^n} \{cx - h(Ax-b)\}$$

Corolário 2: $p_{h^*} = v$ se e só se $h^* \varepsilon \partial_H^b(\phi)$.

Assim, a existência de uma função $h^* \varepsilon \partial_H^b(\phi)$ elimina o "gap" de dualidade entre (P) e (D) ou (P) e (PP). Observa-se que a função linear $h(d)=\lambda d$ em geral não pertence a $\partial_H^b(\phi)$.

Barcia [1] e [2] modifica o dual (D) através de uma seqüência de cortes da função objetivo do problema lagrangeano, gerando uma seqüência de problemas

$$(D_k) \quad w(k+1) = \min_{\lambda\geq 0} \max_{x\varepsilon\{0,1\}^n} \{cx-\lambda(Ax-b)\}$$

$$cx \leq \lfloor w(k) \rfloor,$$

onde $\lfloor w(k) \rfloor$ é o maior inteiro que não excede $w(k)$, $w(0)=w$ e cada $w(k)$ é um limitante superior de v. Barcia mostra que seu processo converge para v se existir um x^* viável em (P), tal que para $\lambda=0$, $(x^*,0)$ é um ponto ótimo do problema:

$$w(k) = \min_{\lambda \geq 0} \max_{x \varepsilon \{0,1\}^n} \{cx - \lambda(Ax-b)\}$$

$$cx \leq \lfloor w(k) \rfloor .$$

Observa-se pois que os cortes do dual geram uma seqüência de cortes também no primal (P) com os problemas

$$(P_k) \qquad V_{k+1} = \max_{x\varepsilon\{0,1\}^n} cx$$

$$Ax \leq b$$
$$cx \leq \lfloor w(k) \rfloor .$$

Como w(k) é um limitante superior de v, a cada corte eliminam-se soluções inviáveis de (P). Assim, a seqüência de funções perturbação $\{\phi_{(k)}\}$ converge para ϕ^*, onde:

$$\phi_{(k)}^{(d)} = \sup_{x\varepsilon\{0,1\}^n} \{cx : Ax \leq d,\ cx \leq \lfloor w(k) \rfloor\}$$

e

$$\phi^*(b) = v, \text{ pois } h^*(d) = \lambda d \varepsilon \partial_H^b(\phi^*) \text{ para } \lambda \to 0.$$

Observa-se portanto que:

(i) o multiplicador de Lagrange ótimo para o método de Barcia é nulo, isto é, a otimalidade estará garantida usando a função linear $h(d)=\lambda d$ quando $\lambda=0$ "perto do ótimo";

(ii) o corolário 2 mostra que não é necessário otimizar cada dual (D_k), $k = 1,2,\ldots$, pois pode-se usar em $\lambda = (\bar{\lambda},\bar{\lambda},\ldots,\bar{\lambda})$, $\bar{\lambda}$ prefixado, e fazer $\lambda_k \to 0$, o que caracteriza um método de penalidades.

Propõe-se então o seguinte algoritmo de penalidades.

Passo 1. Calcular $w(0)=w$ (resolve-se o dual ou o problema de programação linear relaxando as restrições $x\varepsilon\{0,1\}^n$ para $0 \leq x_j \leq 1, \forall j$). Fazer $\lambda(0) = (\lambda_0,\lambda_0,\ldots,\lambda_0)$.

Passo 2. Resolver

$$w(k+1) = \max_{x\varepsilon\{0,1\}^n} \{cx-\lambda(k)(Ax-b)\}$$

$$cx \leq \lfloor w(k) \rfloor,$$

obtendo-se $x(k)$.

Passo 3. Se $x(k)$ é viável ou $\lambda_k \leq \beta$, parar.

Senão: se $\lfloor w(k) - w(k+1) \rfloor \leq \Delta$, fazer $\lambda_{k+1}=\lambda_k/\alpha$ e ir para o passo 2.

Senão fazer $\lambda_{k+1} = \lambda_k$ e ir para o passo 2.

Δ é um inteiro pequeno que depende dos dados do vetor custo. Para β foi usado 10^{-12} e para α, valores entre 1 e 2 (o mais usado foi $\alpha = 1{,}1$ para o qual λ_k decresce pouco a cada iteração).

Para a solução do problema da mochila no passo 2, usou-se o algoritmo de Fayard e Plateau, [3].

3. TESTES COMPUTACIONAIS

O algoritmo de penalidades foi testado em problemas gerados aleatoriamente como se segue:

(i) os coeficientes de A e c foram gerados aleatoriamente no intervalo [0,100]. Os coeficientes de b foram obtidos somando a linha correspondente de A e multiplicando o resultado por $\gamma=0{,}1$ ou $\gamma=0{,}5$;

(ii) os problemas de tamanhos 20x20, 50x20, 75x20 e 100x20 foram testados, conforme se pode observar na tabela 1;

(iii) na tabela 1 observam-se ainda as colunas de resultados usando programação linear, do algoritmo Branch-and-Bound do Burroughs 6800 (MPS-Tempo), do algoritmo de penalidades e os respectivos tempos de processamento.

Uma característica diferente da de Barcia [1] foi notada, pois o método de penalidades sempre encontrou uma solução viável, embora em muitos casos não-ótima. Mesmo assim, a diferença entre os valores obtidos para a solução viável e os obtidos usando o algoritmo Branch-and-Bound não foram significativas. Para o problema 8, o resultado de penalidades foi melhor que para o Branch-and-Bound em 12 minutos de processamento.

TABELA 1

TESTES COM PROBLEMAS GERADOS ALEATORIAMENTE

PROBLEMA	TAMANHO	γ	PROGRAMAÇÃO LINEAR	B & B	TEMPO B & B (Seg.)	PENALIDADE	TEMPO PENALIDADE (seg.)	DIFERENÇA B & B
1	20 x 20	.1	159.06	99	20	99	4	0
2	50 x 20	.1	440.6	386	210	290	26	96
3	75 x 20	.1	681.27	568(*)	720	470	10	98
4	100 x 20	.1	902.31	790(*)	720	732	66	58
5	20 x 20	.5	716.26	679	41	663	1	16
6	50 x 20	.5	1 878.90	1 828(*)	720	1 803	10	25
7	75 x 20	.5	2 837.45	2 770(*)	720	2 663	17	107
8	100 x 20	.5	3 544.62	3 119(*)	720	3 454	20	-335

(*) melhor valor em 12 minutos

B & B = Branch and Bound

4. COMENTÁRIOS FINAIS

Algumas dificuldades encontradas com o uso do algoritmo de penalidades relacionam-se ao "tamanho" do parâmetro λ_k. Quando este é "muito pequeno", o problema da mochila torna-se difícil e sua soluçao demorada através do algoritmo de Fayard e Plateau. Teoricamente a convergência para a solução ótima de (P) seria obtida, mas na prática isto nem sempre ocorre devido ao tempo excessivo gasto. Acredita-se que esse mesmo problema afete o algoritmo de Barcia.

A vantagem da presente abordagem está em sempre alcançar uma "boa" solução viável, cercando o valor ótimo de (P).

REFERÊNCIAS

[1] P. Barcia, "The bound improving sequence algorithm". Operations Research Letters, 4:27-30, 1985.

[2] P. Barcia, "Constructive dual methods for discrete programming". School of Combinatorial Optimization. Anais. Rio de Janeiro, July 1985.

[3] D. Fayard, G. Plateau, "An algorithm for the solution of the 0-1 knapsack problem". Computing, 28:269-287, 1982.

[4] L.A.N. Lorena, "Uma nova classe de funções penalidade e sua aplicação a problemas discretos". Tese de doutorado. COPPE-UFRJ, junho de 1985.

[5] L.A.N. Lorena, P.R. Oliveira, "Uma abordagem geral de dualidade/penalidade em programação matemática. XVIII Sobrapo. Anais. São José dos Campos, novembro de 1985.