

DECOMPOSIÇÃO DE BENDERS E SUAS APLICAÇÕES EM MODELOS DE PROGRAMAÇÃO MISTA E EM PROBLEMAS DE ESTOQUE E ROTEIRIZAÇÃO

Patrícia Prado Belfiore

Universidade de São Paulo

Av. Prof. Luciano Gualberto, 908 - Sala G-121 - 1º andar - Cidade Universitária

e-mail: patricia.belfiore@poli.usp.br

Oswaldo Luiz do Valle Costa

Universidade de São Paulo

Av. Prof. Luciano Gualberto, 908 - Sala G-121 - 1º andar - Cidade Universitária

e-mail: oswaldo@lac.usp.br

Luiz Paulo Lopes Fávero

Universidade de São Paulo

Av. Prof. Luciano Gualberto, 908 - Sala G-121 - 1º andar - Cidade Universitária

e-mail: lpfaver@usp.br

Resumo

Os modelos de programação inteira mista de grande porte, devido ao grande número de variáveis inteiras, são de difícil resolução. Para contornar esta dificuldade, uma alternativa seria a implementação do método de decomposição de Benders. Este trabalho tem como objetivo descrever o método de decomposição de Benders para programação mista e suas aplicações em Problemas de Estoque e Roteirização. O algoritmo de Benders decompõe o problema principal em subproblemas, facilitando a resolução do problema.

Palavras-chave: Decomposição de Benders, Problema de Estoque e Roteirização, Pesquisa Operacional.

Abstract

The models of great integer mixed programming, due the great number of integer variables are of difficult resolution. In order to transpose this difficulty an alternative would be to implement the method of benders decomposition. The aim of this work is to describe this method for mixed programming and how it can be applied in Inventory Routing Problems. The benders algorithm decomposes the main problem into sub problems, making them easier to be solved.

Keywords: Benders Decomposition, Inventory Routing Problem, Operational Research.

1. Introdução

À medida que o número de variáveis inteiras cresce significativamente, os modelos de programação inteira mista de grande porte tornam-se de difícil solução. Quando o problema é muito complexo, pode ser interessante subdividi-lo em subproblemas.

J. F. Benders (1962) propôs um método de decomposição para resolver problemas de programação mista. Esta técnica decompõe o problema principal em subproblemas.

No caso de problemas de estoque e roteirização, o problema principal é decomposto em um subproblema de alocação de estoques e um subproblema de roteirização de veículos.

A apresentação do trabalho será detalhada a seguir. No item 2 será descrito o método de decomposição de Benders para programação mista. Já o item 3 apresenta o problema de estoques e roteirização (*inventory routing problem* – IRP) e o item 4 as aplicações de decomposição de Benders ao IRP. O item 5 mostra um exemplo do método de decomposição de Benders e o item 6 o algoritmo de decomposição de Benders. Finalmente no item 7 encontram-se as conclusões e futuras pesquisas.

2. Decomposição de Benders para programação mista

Um dos principais conceitos utilizados na resolução do algoritmo de decomposição é o da Projeção. O problema pode ser descrito como em Geoffrin (1970):

$$(2.1) \quad \begin{aligned} & \max_{x \in X; y \in Y} f(x, y) \\ & \text{s.a. } g(x, y) \leq 0 \end{aligned}$$

que envolve a otimização sobre o espaço conjunto das variáveis x e y . Define-se sua projeção sobre o espaço apenas da variável x como:

$$(2.2) \quad \begin{aligned} & \max_{x \in X} \text{Max}_{y \in Y} f(x, y) \\ & \text{s.a. } g(x, y) \leq 0 \end{aligned}$$

que pode ser escrito como:

$$(2.3) \quad \max_{x \in X} v(x)$$

onde:

$$(2.4) \quad \begin{aligned} v(x) &= \max_{y \in Y} f(x, y) \\ & \text{s.a. } g(x, y) \leq 0 \end{aligned}$$

O Problema de duas variáveis pode ser resolvido em dois estágios.

Considere agora o seguinte problema de programação mista:

$$(2.5) \quad \begin{aligned} & \min_{x, y} c'x + d'y \\ & Ax + By \geq b \\ & x \in I \\ & y \geq 0 \end{aligned}$$

onde I pode ser, por exemplo, um conjunto de números inteiros, ou um outro conjunto de restrições lineares.

Utilizando o conceito de projeção sobre o problema (2.5), pode-se reescrevê-lo como:

$$(2.6) \quad \begin{aligned} & \min c'x + Q(x) \\ & x \in I \end{aligned}$$

Fixando valores de x sobre o problema (2.5), o restante do modelo a ser resolvido é:

$$(2.7) \quad \begin{aligned} Q(\bar{x}) &= \min_y d' y \\ By &\geq b - A\bar{x} \\ y &\geq 0 \end{aligned}$$

O problema completo de minimização pode ser escrito da seguinte forma:

$$(2.8) \quad \min_{x \in I} \left[c'x + \min_y \{ d'y \mid By \geq b - Ax, y \geq 0 \} \right]$$

Cabe aqui relembrar o Teorema da Dualidade em programação linear.

Define-se dualidade através do par de problemas de PL abaixo:

Primal:

$$\begin{aligned} \min_x \quad & c'x \\ & x \in R^n \\ \text{s.a.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

Dual:

$$\begin{aligned} \max_u \quad & u'b \\ & u'A \leq c' \\ & u \geq 0 \end{aligned}$$

Tem-se o seguinte teorema (Winston, 1995):

Teorema da Dualidade: Se um dos problemas (primal ou dual) tem uma solução ótima finita então o outro também tem uma solução finita e os valores correspondentes das funções objetivos são iguais. Se um problema é ilimitado o outro não tem solução factível.

Pelo Teorema da Dualidade tem-se que:

$$(2.7) \quad \begin{aligned} Q(\bar{x}) &= \min_y d' y \\ \text{s.a.} \quad & By \geq b - A\bar{x} \\ & y \geq 0 \end{aligned}$$

é semelhante a:

$$\begin{aligned}
 (2.9) \quad Q(\bar{x}) = & \max_u u'(b - A\bar{x}) \\
 \text{s.a.} \quad & u'B \leq d' \\
 & u \geq 0
 \end{aligned}$$

Cabe agora relembrar alguns fatos básicos em programação linear.

Seja $\{u_1, \dots, u_k\}$ o conjunto de pontos extremos e $\{d_1, \dots, d_l\}$ o conjunto de direções extremas da região de viabilidade K de (2.9), definida como segue:

$$K = \{u \in R^m; u'B \leq d', u \geq 0\}$$

Se $(b - A\bar{x})'d_j > 0$ para algum $j = 1, \dots, l$ então a solução do problema é ilimitada $(-\infty)$. Se

$(b - A\bar{x})'d_j \leq 0$ para todo $j = 1, \dots, l$ então $u'(b - Ax)$ atinge seu mínimo sobre K em um ponto extremo de K .

Portanto, pelo Teorema da Dualidade, o problema (2.7) somente é viável se o problema (2.9) tem uma solução ótima finita, ou seja, se existe x que satisfaça a condição:

$$(2.10) \quad (d_j)'(b - Ax) \leq 0, \quad j = 1, \dots, l$$

Dessa forma podemos reescrever o problema (2.6) como:

$$\begin{aligned}
 (2.11) \quad \min & \{c'x + \max_{j=1, \dots, k} (u_j)'(b - Ax)\} \\
 & x \in I \\
 \text{s.a.} \quad & (d_j)'(b - Ax) \leq 0, \quad j = 1, \dots, l
 \end{aligned}$$

Dessa forma, adicionando e definindo a variável λ como a maior cota inferior de $Q(x)$, reescreve-se (2.11) como:

$$\begin{aligned}
 (2.12) \quad \min_{x, \lambda} & c'x + \lambda \\
 & x \in I, \lambda \in R
 \end{aligned}$$

$$(2.13) \quad \text{s.a.} \quad (\bar{d}_j)'(b - Ax) \leq 0, \quad j = 1, \dots, l,$$

$$\begin{aligned}
 (2.14) \quad & \lambda \geq (\bar{u}_j)'(b - Ax), \quad j = 1, \dots, k, \\
 & x \in I
 \end{aligned}$$

O método de decomposição de Benders baseia-se na aplicação dos conceitos de Dualidade, Projeção e do relaxamento das restrições (2.13) e (2.14) para a resolução do problema (2.5). Com base nestes conceitos o problema (2.5) é reescrito na forma do problema (2.12) com as restrições (2.13) e (2.14).

Passo 1: Resolve-se o problema (2.12) sem as restrições (2.13) e (2.14), caso estas restrições ainda não tenham sido acrescentadas pelo passo 2, encontrando uma solução ótima $(\hat{x}, \hat{\lambda})$. Caso a solução não seja encontrada, o problema (2.5) não tem solução.

Passo 2: Para testar a viabilidade da solução $(\hat{x}, \hat{\lambda})$, resolve-se o problema linear (2.9).

Se u tende a infinito em (2.9), adiciona-se a restrição $\sum u_i \leq M$ ao problema (2.9), onde M é uma constante positiva de larga escala.

Se o valor de $Q(x)$ é menor ou igual a $\hat{\lambda}$, então a solução $(\hat{x}, \hat{\lambda})$ encontrada é ótima para (2.12) a (2.14) e \hat{y} é obtido por $B\hat{y} = b - A\hat{x}$, terminando a otimização. Caso o valor $Q(x)$ for maior que $\hat{\lambda}$, adiciona-se uma restrição do tipo (2.14) ao problema (2.12) com os valores de u' encontrados em (2.9). Se o problema (2.9), com \hat{x} , não possuir solução ótima finita, o que significa que $(d_j)'(b - Ax) \geq 0$, então com os valores encontrados em (2.9) adiciona-se uma restrição do tipo (2.13) ao problema (2.12) e retorna-se ao passo 1.

Teorema: O algoritmo de Benders converge para a solução ótima do problema (2.5) em um número finito de iterações (Geoffrion, 1972).

3. Problema de Estoque e Roteirização

Iniciativas importantes visando elevar os ganhos totais da cadeia têm surgido nas indústrias, com o amparo do Movimento ECR – Efficient Consumer Response, ou Resposta Eficiente ao Consumidor. Uma das técnicas propostas pelo ECR é o VMI – Vendor Managed Inventory – Estoque Gerenciado pelo Fornecedor – que tem sido muito disseminada na indústria mundial. O VMI tem como objetivo a redução de custos através da integração dos componentes da cadeia de abastecimento. O processo de reposição através do VMI pode ocorrer em qualquer elo da cadeia de abastecimento.

O VMI é uma técnica no qual o fornecedor controla os níveis de estoque de seus clientes, e decide quando e quanto entregar de mercadoria para cada cliente. Sendo assim, os cálculos são realizados por um algoritmo cadastrado no fornecedor, formado por parâmetros pré-estabelecidos pelo vendedor e comprador e baseado nas informações obtidas do cliente. Nesse modelo o cliente é apenas informado da quantidade que será enviada. Desta maneira, ele faz um acompanhamento, monitora, mas não controla o processo (Klingenberg e Antunes, 2002).

Em muitas aplicações, o vendedor, além de controlar os estoques dos clientes, também administra uma frota de veículos para transportar os produtos aos clientes. Neste caso, o objetivo do vendedor é não só administrar o reabastecimento ótimo dos estoques como também a distribuição dos produtos. Este problema é chamado Problema de Estoque e Roteirização (IRP – Inventory Routing Problem). O IRP tem como característica a política VMI e desenvolve metodologias para solução deste problema (Campbell *et al.*, 1998).

3.1 Definição do Problema

O Problema de Estoque e Roteirização (IRP) trata da distribuição de um único produto, a partir de um único centro de distribuição, que atende N clientes dentro de um horizonte de planejamento T , possivelmente infinito. O cliente i consome o produto a uma taxa u_i e tem uma capacidade de armazenagem C_i . O nível de estoque do cliente i no instante t é I_i^t . A distribuição dos produtos é

feita através de M veículos homogêneos e cada veículo têm uma capacidade C_v . A quantidade entregue ao cliente i pela rota r no instante t é Q_{ir}^t . O número total de rotas de entrega é R . Define-se c_r o custo de executar a rota $r = 1, \dots, R$ e $c_{est,i}$ o custo de armazenagem do cliente i .

O objetivo é minimizar o custo médio diário de distribuição durante o horizonte de planejamento de modo que não haja falta de estoques para os clientes. Podem-se adicionar ao modelo custos de estoque e custos de falta de produtos.

A cada instante t , são tomadas decisões de roteamento de veículos e reabastecimento de estoque dos clientes.

O Problema engloba três fases:

1. Seleção das rotas
2. Determinar a cada dia quais rotas irão ocorrer e quanto entregar para cada cliente da rota durante cada visita
3. Roteirização de veículos: qual o melhor roteiro

3.2 Modelagem

Definiremos a seguir um modelo de programação inteira mista multi-período baseado nas idéias de Campbell *et al.* (1999).

Neste modelo, as decisões sobre quais rotas irão ocorrer, quanto entregar para cada cliente da rota e qual o melhor roteiro são tomadas diariamente.

Parâmetros:

$R \rightarrow$ total de rotas de entrega

$c_r \rightarrow$ custo de executar a rota r , $r = 1, \dots, R$

$c_{est,i} \rightarrow$ custo de armazenagem do cliente i (R\$/item dia)

$N \rightarrow$ número de clientes

$M \rightarrow$ número de veículos disponíveis

$T \rightarrow$ horizonte de planejamento

$u_i \rightarrow$ taxa de consumo do cliente i , $i = 1, \dots, N$

$C_i \rightarrow$ capacidade de armazenagem do cliente i , $i = 1, \dots, N$

$C_v \rightarrow$ capacidade do veículo

$I_i^0 \rightarrow$ estoque inicial do cliente i , $i = 1, \dots, N$

$I_i^t \rightarrow$ nível de estoque do cliente i no instante t , $i = 1, \dots, N$, $t = 1, \dots, T$

$T_r =$ duração da rota r (fração do dia)

$T_d =$ tempo total disponível por dia

Variáveis de decisão:

Q_{ir}^t = volume total entregue ao cliente i da rota r no dia t

x_r^t = variável binária, indica se a rota r é executada no dia t ($x_r^t = 1$) ou não ($x_r^t = 0$)

O nível de estoque do cliente i no instante t pode ser definido como:

$$(3.1) \quad I_i^t = I_i^{t-1} + \sum_{r=1}^R Q_{ir}^t - u_i$$

A quantidade mínima a ser entregue ao cliente i no final do dia t é:

$$(3.2) \quad d_i^t = \max(0, tu_i - I_i^0)$$

A quantidade máxima a ser entregue ao cliente i no final do dia t é:

$$(3.3) \quad D_i^t = tu_i + C_i - I_i^0$$

O Problema pode ser formulado da seguinte maneira:

$$(3.4) \quad \begin{aligned} \min \quad & \sum_t \sum_r c_r x_r^t + \sum_t \sum_i c_{est,i} I_i^t \\ & d_{it} \leq \sum_{s=1}^t \sum_{r \in R_i} Q_{ir}^s \leq D_{it} \quad i = 1, \dots, N, t = 1, \dots, T \quad (1) \\ & \sum_{i \in I_r} Q_{ir}^t \leq C_v x_r^t \quad r = 1, \dots, R, t = 1, \dots, T \quad (2) \\ & x_r^t \text{ binary}, 0 \leq Q_{i,r}^t \leq \min(C_v, C_i). \end{aligned}$$

O objetivo do modelo é minimizar os custos de transporte e estoque de modo que a demanda de todos os clientes sejam atendidas. O modelo pode minimizar apenas os custos de distribuição, desconsiderando os custos de estoque, sujeito às mesmas restrições.

A restrição (1) garante que não haverá falta de estoque para o cliente i e que a capacidade de armazenagem do cliente i não será excedida.

A restrição (2) garante que o volume total entregue aos clientes i da rota r não excederá a capacidade do veículo.

Caso o número de veículos (M) em cada período seja limitado, adiciona-se a seguinte restrição:

$$\sum_{r=1}^R x_r^t \leq M \quad t = 1, \dots, T \quad (3)$$

4. Aplicações de Decomposição de Benders ao Problema de Estoque e Roteirização

O objetivo do modelo de programação inteira em tempo discreto que não considera os custos de estoque é:

$$\min \sum_{r=1}^R \sum_{t=1}^T c_r^t x_r^t$$

Podemos definir as matrizes A e B e um vetor b de forma que:

$$Ax + By \geq b$$

represente o seguinte conjunto de restrições:

$$\begin{aligned} d_{it} &\leq \sum_{s=1}^t \sum_{r \in R_i} Q_{ir}^s \leq D_{it} & i = 1, \dots, N, t = 1, \dots, T \\ \sum_{i \in I_r} Q_{ir}^t &\leq C_v x_r^t & r = 1, \dots, R, t = 1, \dots, T \\ \sum_{r=1}^R x_r^t &\leq M & t = 1, \dots, T \\ x_r^t &\text{binary}, 0 \leq Q_{i,r}^t \leq \min(C_v, C_i). \end{aligned}$$

Logo o problema de programação mista a ser resolvido é:

$$\begin{aligned} \min \quad & c'x + 0'y \\ & Ax + By \geq b \\ & x \text{ binário} \end{aligned}$$

que pode ser resolvido aplicando a decomposição de Benders. O vetor x representa as rotas e o vetor y as quantidades entregues.

No caso do modelo que considera os custos de estoque o objetivo é:

$$\min \sum_{r=1}^R \sum_{t=1}^T c_r^t x_r^t + \sum_{i=1}^N \sum_{t=1}^T c_{est,i} I_i^t$$

onde:

$$I_i^t = I_i^{t-1} + \sum_{r=1}^R Q_{ir}^t - u_i$$

Definem-se as matrizes A e B e um vetor b de forma que:

$$Ax + By \geq b$$

represente o mesmo conjunto de restrições do modelo anterior.

O problema de programação mista é:

$$\begin{aligned} \min \quad & c'x + d'y \\ & Ax + By \geq b \\ & x \text{ binário} \end{aligned}$$

5. Exemplo utilizando algoritmo de decomposição de Benders

Consideremos o seguinte problema:

$$\begin{aligned}
 (5.1) \quad & \min_{x,y} 2x_1 + 2x_2 + 5y \\
 & \text{s.a. } 3x_1 + 2x_2 + y \geq 5 \\
 & \quad -x_1 + x_2 + 4y \geq 7 \\
 & \quad x_1 - x_2 + 2y \geq 4 \\
 & \quad x_1, x_2 \in I \\
 & \quad y \geq 0
 \end{aligned}$$

Reescrevendo o problema (5.1) de acordo com a equação (2.7) temos que $Q(x)$ é:

$$\begin{aligned}
 (5.2) \quad & \min_y 5y \\
 & \text{s.a.} \\
 & \quad \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix} y \geq \begin{bmatrix} 5 - 3\bar{x}_1 - 2\bar{x}_2 \\ 7 + \bar{x}_1 - \bar{x}_2 \\ 4 - \bar{x}_1 + \bar{x}_2 \end{bmatrix} \\
 & \quad \bar{x}_1, \bar{x}_2 \in I \\
 & \quad y \geq 0
 \end{aligned}$$

O dual de (5.2) é:

$$\begin{aligned}
 (5.3) \quad & \max_u (u_1, u_2, u_3) \begin{bmatrix} 5 - 3\bar{x}_1 - 2\bar{x}_2 \\ 7 + \bar{x}_1 - \bar{x}_2 \\ 4 - \bar{x}_1 + \bar{x}_2 \end{bmatrix} \\
 & \text{s.a.} \\
 & \quad u_1 + 4u_2 + 2u_3 \leq 5 \\
 & \quad u_1, u_2, u_3 \geq 0
 \end{aligned}$$

Abaixo será especificada cada passagem utilizando o algoritmo de decomposição de Benders.

Passo 1a) Problema principal: O primeiro passo é resolver o problema (5.1) sem as restrições utilizando o conceito de projeção:

$$\begin{aligned}
 (5.4) \quad & \min_{x,\lambda} 2x_1 + 2x_2 + \lambda \\
 & \text{s.a. } x_1, x_2 \in I \\
 & \quad \lambda \geq 0
 \end{aligned}$$

A solução é: $x_1 = 0, x_2 = 0, \lambda = 0$.

Passo 2a) Subproblema: Verificar se a solução $x_1 = 0, x_2 = 0, \lambda = 0$ é ótima. Substituindo os valores $x_1 = 0, x_2 = 0$ na equação (5.3) temos:

$$(5.5) \quad \begin{aligned} Q(x) &= \max_u 5u_1 + 7u_2 + 4u_3 \\ \text{s.a.} \quad &u_1 + 4u_2 + 2u_3 \leq 5 \\ &u_1, u_2, u_3 \geq 0 \end{aligned}$$

A solução é: $u_1 = 5, u_2 = 0, u_3 = 0$ e $Q(x) = 25$.

O valor $\lambda = 0$ é menor que $Q(x) = 25$, portanto a solução $x_1 = 0, x_2 = 0, \lambda = 0$ não é ótima. Com os valores $u_1 = 5, u_2 = 0, u_3 = 0$, adiciona-se uma restrição do tipo $\lambda \geq (\bar{u})'(b - Ax)$, ou seja, $\lambda \geq 5(5 - 3x_1 - 2x_2)$, ao problema principal (5.4).

Passo 1b) O problema principal será:

$$(5.6) \quad \begin{aligned} \min_{x, \lambda} \quad &2x_1 + 2x_2 + \lambda \\ \text{s.a.} \quad &\lambda \geq 25 - 15x_1 - 10x_2 \\ &x_1, x_2 \in I \\ &\lambda \geq 0 \end{aligned}$$

A solução é $x_1 = 1, x_2 = 1, \lambda = 0$.

Passo 2b) Verificar se a solução $x_1 = 1, x_2 = 1, \lambda = 0$ é ótima. Para isso resolve-se o subproblema (5.3) substituindo os valores de x_1 e x_2 encontrados em (5.6):

$$(5.7) \quad \begin{aligned} Q(x) &= \max_u 7u_2 + 4u_3 \\ \text{s.a.} \quad &u_1 + u_2 + u_3 \leq 5 \\ &u_1, u_2, u_3 \geq 0 \end{aligned}$$

A solução ótima é $u_1 = 0, u_2 = 0, u_3 = 2,5$ e $Q(x) = 10$. A solução encontrada no problema (5.6) não é ótima, pois $\lambda = 0 < Q(x) = 10$. Substituindo os valores $u_1 = 0, u_2 = 0, u_3 = 2,5$ em $\lambda \geq \bar{u}'(b - Ax)$, adiciona-se a restrição $\lambda \geq 2,5(4 - x_1 + x_2)$ ao problema principal (5.6).

Passo 1c) O problema principal será:

$$(5.8) \quad \begin{aligned} \min_{x, \lambda} \quad &2x_1 + 2x_2 + \lambda \\ \text{s.a.} \quad &\lambda \geq 25 - 15x_1 - 10x_2 \\ &\lambda \geq 10 - 2,5x_1 + 2,5x_2 \\ &x_1, x_2 \in I \\ &\lambda \geq 0 \end{aligned}$$

A solução é: $x_1 = 1, x_2 = 0, \lambda = 10$.

Passo 2c) Testar a solução $x_1 = 1, x_2 = 0, \lambda = 10$. Substituindo estes valores no subproblema (5.3) temos:

$$(5.9) \quad \begin{aligned} Q(x) &= \max_u 2u_1 + 8u_2 + 3u_3 \\ \text{s.a.} \quad &u_1 + u_2 + u_3 \leq 5 \\ &u_1, u_2, u_3 \geq 0 \end{aligned}$$

A solução é: $u_1 = 0, u_2 = 1.25, u_3 = 0$ e $Q(x) = 10$.

O valor $\lambda = 10$ é igual a $Q(x)$. Portanto, $x_1 = 1$ e $x_2 = 0$ é a solução ótima. Substituindo $x_1 = 1$ e $x_2 = 0$ em (5.2) temos:

$$(5.10) \quad \begin{aligned} \min_y & 5y \\ \text{s.a.} \quad & \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix} y \geq \begin{bmatrix} 2 \\ 8 \\ 3 \end{bmatrix} \\ & y \geq 0 \end{aligned}$$

A solução é $y = 2$ com $5y = 5 \times 2 = 10$. A solução ótima de (5.1) é $y = 2, x_1 = 1, x_2 = 0$ e $F_{\text{obj}} = 12$.

6. Algoritmo de decomposição de Benders

O algoritmo de decomposição de Benders está descrito abaixo:

x := variável inteira

y := real positiva

λ := $-\infty$

$Q(x)$:= ∞

enquanto $Q(x) - \lambda > \varepsilon$ **fazer**

 {resolver o problema principal}

$\min_{x, \lambda} \{c'x + \lambda \mid \text{cortes}, x \in I\}$

$\lambda = \bar{\lambda}$

 {resolver subproblema}

$\max_u \{u'(b - A\bar{x}) \mid u'B \leq d', u \geq 0\}$

se solução infactível **então**

 fixar solução infactível \bar{d}

 e adicionar restrição $\bar{d}'(b - Ax) \leq 0$ ao problema principal

senão

fixar ponto extremo \bar{u}
 e adicionar restrição $\lambda \geq \bar{u}'(b - Ax)$ ao problema principal
 $Q(x) := \bar{u}'(b - Ax)$
fim se
fim enquanto
fim

7. Conclusões e futuras pesquisas

Este trabalho teve como objetivo apresentar o algoritmo de decomposição de Benders e suas aplicações ao problema de estoque e roteirização. Como os problemas de programação mista são de difícil solução, uma alternativa é decompor o problema principal em subproblemas, como ocorre no método de Benders.

Como futuras pesquisas, este método pode ser aplicado em outros tipos de problemas de programação mista e pode também, ser utilizado simultaneamente com outros algoritmos.

Referências Bibliográficas

Benders, J.F. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4, p. 238-252, 1962.

Campbell, A.; Clarke, L.; Kleywegt, A. e Savelsbergh, M.W.P. The Inventory Routing Problem. In: Fleet Management and Logistics, Grainic, T.G., Laporte, G. (eds), Kluwer Academic Publishers, p. 95-112, 1998.

Campbell, A.; Clarke, L. e Savelsbergh, M. *An Inventory Routing Problem*, April 28, 1999.

Geoffrion, A.M. Elements of Large-Scale Mathematical Programming. *Management Science*, v. 16, n. 11, 1960.

Geoffrion, A.M. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, v. 10, n. 4, p. 237-260, 1972.

Klingenberg, C.O. e Antunes, J.A.V.J. Construção de um modelo orientativo para a implantação do VMI em empresas da indústria supermercadista brasileira. In: XXXVII Assembléia do Conselho Latino-Americano de Escolas de Administração, 2002.

Winston, W. L.; *Introduction to Mathematical Programming: Applications and Algorithms*. Belmont, 2a.edição, Ca.: Duxbury Press, 1995.

2º Congresso ECR Brasil. *O ECR no Brasil-Scorecard*. São Paulo, 2000.