UM ALGORITMO EFICIENTE PARA ENCONTRAR SOLUÇÕES APROXIMADAS DE PROBLEMAS DE PROGRAMAÇÃO BIVALENTE (0 - 1), UTILIZANDO DUALIDADE EM PROGRAMAÇÃO INTEIRA

Félix Eduardo Vaca Obando

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS (D.Sc.)

Aprovada por:

Prof. Nelson Maculan Filho
(Presidente)

Prof. Carlos Alberto Nunes Cosenza

Prof. João Lizardo R.H. de Araújo

Prof. Jack Schechtman

Prof. Claúdio T. Bornstein

RIO DE JANEIRO, RJ - BRASIL
AGOSTO DE 1978

VACA OBANDO, FÉLIX EDUARDO

Um Algoritmo Eficiente para Encontrar Soluções Aproximadas de Problemas de Programação Bivalente (0-1), Utilizando Dualidade em Programação Inteira. |Rio de Janeiro| 1978

IX, 110p. 29,7cm (COPPE-UFRJ, D.Sc., Engenharia de Sistemas e computação, 1978)

Tese - Univ. Fed. Rio de Janeiro, Fac. Engenharia

1. Programação Matemática I. COPPE/UFRJ II. Título (série).

Curriculum Vitae

Félix Eduardo Vaca Obando, nascido em 20 de fevereiro de 1943, na cidade de Tulcán - Equador. Diplomado em Engenharia Mecânica pela Escola Politécnica Nacional de Quito - Equador em 1967. Cursou Pós-Graduação em Engenharia Industrial na Escola Politécnica Nacional de Quito de 1967 a 1970. Trabalhou como Professor Auxiliar na mesma Escola desde 1967 até fevereiro de 1970. Realizou cursos na Companhia Siderúrgica Nacional - Volta Redonda, com bolsa do Itamarati, de agosto a outubro de 1968 e na Comissão Nacional de Energia Atômica em Buenos Aires - Argentina, com bolsa da OEA, de março a junho de 1970. Em julho de 1970 ingressou na COPPE/UFRJ onde obteve o título de Mestre em Ciências em Engenharia de Produção em 10 de dezembro de 1971. Em janeiro de 1972 foi contratado como Professor da COPPE/UFRJ e a partir de maio de 1977 é Professor do Departamento de Engenharia Industrial da Escola de Engenharia da UFRJ. Desenvolveu projetos relacionados com Modelos Matemáticos e outras áreas afins na COPPETEC. Participou de vários Congressos e reuniões científicas tendo publicado diversos artigos em Programação Matemática.

A Guadalupe

T. Eduardo e

Cristian

Agradecimentos

Meu agradecimento muito especial ao Prof, Nelson Maculan Filho que como orientador, colega e amigo foi um constante incentivador desta caminhada que aqui termina.

Aos colegas do Programa de Engenharia de Sistemas e Computação pelo apoio e compreensão que souberam brindar durante a elaboração deste trabalho.

Ao pessoal da Biblioteca do Centro de Tecnologia por seu espírito de colaboração durante a pesquisa bibliográfica.

Ao NCE pela utilização de suas instalações.

A todos aqueles que de uma ou forma colaboraram para a conclusão desta Tese.

A Consolação de Maria Moreira Lima por seu excelente trabalho de mecanografia.

Resumo

Este trabalho apresenta um procedimento que gera uma sequência finita de soluções de um problema de programação inteira bivalente (0 - 1), utilizando dualidade em programação inteira e o método de geração de coluna. O algoritmo gera simultaneamente soluções denominadas ξ - viáveis que podem ser mais interessantes que a própria solução ótima para determinada classe de problemas. São mostrados e analisados os resultados, de um conjunto de problemas teste, que foram obtidos mediante um programa elaborado em linguagem Fortran.

Abstract

This work presents a prscedure for generating a finite sequence of solutions to a given zero-one integer programming prsblem, using duality in integer programming and the generating column method. This algorithm generates simultaneously ξ - feasible solutions that could be more interesting than optimal solutions for a class of specific problems. We present and analyse, results for a set of test problems, obtained by means of a Fortran program.

INDICE

CAPÍTULO III

CAPÍTULO IV

CAPÍTULO I

1. Introdução

É perfeitamente conhecido que os modelos de programação matemática são uma abstração da realidade de um problema. Assim também sua solução ótima pode ser vista como uma solução próxima da "Ótima real". Daí a importância das técnicas de pós otimização, análise de sensibilidade e parametrização que pertencem uma informação mais completa do comportamento do modelo quando sujeito a determinadas mudanças em seus parâmetros e que encontram um conjunto de soluções das quais algumas serão de mais fácil implementação.

À medida que as técnicas de programação matemática são mais divulgadas, sua utilização se difunde cada vez mais e vão aumentando as áreas de aplicação destas técnicas.

Deve-se sempre lembrar que as técnicas de programação matemática são uma ferramenta para o planejamento e para tomadas de decisões e por isto fornecem uma série de alternativas, entre as quais uma é selecionada inicialmente e na sua implementação, algumas alterações deverão ser efetuadas, a fim de conseguir os melhores resultados possíveis na solução de um determinado problema. Também se deve lembrar que, a construção de um modelo matemático é complementada com os dados, os quais nem sempre são confiáveis cem por cento, portanto soluções ótimas obtidas para um modelo continuaram a estar próximas de ótima real. Assim sendo, para tomar uma decisão em base a estas informações, é necessário encontrar várias soluções em torno da Ótima para que sirvam como alternativas, caso não seja conveniente implementar a solução ótima. Se alguma destas soluções é inviável e esta inviabilidade se verifica em uma percentagem mínima de um ou vários dos recursos utilizados, esta solução poderá tornar-se mais atrativa que alguma viável porém mais afastada da Ótima.

Dentro deste contexto, será apresentado um método para resolver problemas de programação inteira zero-um, que pode ser utilizado para obter soluções de vários problemas reais.

O método aqui apresentado, gera uma sequência finita de soluções, utilizando dualidade em programação inteira e o método de geração de colunas, E devido a sua eficiência e simplicidade possue vantagens sobre outros métodos tais como, os algoritmos clássicos de programação inteira. O tempo de computação gasto para a obtenção de soluções de problemas de programação zero-um, com um número relativamente grande de variáveis, é pequeno em relação ao tempo utilizado por outros algoritmos.

A eficiência do método pode ser melhorada ainda mediante a transformação de problemas com várias restrições, em um problema equivalente com uma restrição.

O método aqui apresentado, pode ser utilizado também para uma Análise de Pós-Otimização dos problemas de programação zero-um, dado que na procura da melhor solução viável são encontradas outras soluções viáveis de menor valor (para maximizar) e soluções que chamaremos de ξ - viáveis, de maior valor, sem nenhum esforça computacional adicional.

2. O Problema de Programação Linear Inteira

Um problema de programação linear, em que as variáveis devem assumir valores não fracionários ou inteiros, são classificados como problemas de programação linear inteira.

Estes problemas, são importantes porque um grande número de problemas reaistno campo da pesquisa operacional, podem ser representados como problemas com variáveis inteiras.

O problema geral de programação inteira pode ser definido como:

$$\text{Maximizar } z = CX$$

sujeito a (1)

$$AX \leq b$$

$$x_j \geq 0 \text{ e inteiro } j = 1,\ldots,n$$

onde:

$C = (c_j)$ é um vetor linha de n componentes

$X = (x_j)$ é um vetor coluna de n componentes

$A = (a_{ij})$ é uma matriz m x n

$b = (b_i)$ é um vetor coluna com m componentes

Se todas as variáveis x_j são inteiras teremos um problema chamado totalmente inteiro. Se somente algumas das variáveis x_j são inteiras sendo as restantes continuas teremos um problema chamado de programação mista.

A partir da técnica apresentada por Gomory, em 1958, para resolver o problema de programação linear inteira, várias outras técnicas foram desenvolvidas. Ainda mais, técnicas especiais, para problemas específicos de programação inteira, foram pesquisadas e continuam a ser pesquisadas, constituindo-se assim este campo numa fonte geradora de técnicas, que pretendem ser cada vez mais eficientes e especializadas.

Dentre os problemas especiais de programação inteira que vem despertando maior interesse podemos citar o problema de programação linear inteira bivalente ou zero-um.

3. O Problema de Programação Linear Inteira Bivalente

O problema de programação linear inteira bivalente ou zero-um, que daqui por diante denominaremos simplesmente de problema de programação bivalente ou zero-um, é um problema em que as variáveis só podem assumir valores zero ou um, pode ser expressado como:

$$\text{Maximizar } z = \sum_{j=1}^{n} c_j x_j$$

$$\text{sujeito a } \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad i = 1,\ldots,m \tag{2}$$

$$x_j = 0 \text{ ou } 1 \qquad j = 1,\ldots,n$$

Consideremos que:

a) os $c_j \geq 0$, $(j = 1,2,\ldots,n)$, caso isto não aconteça podemos utilizar a mudança de variáveis seguinte, $x_j = 1 - x_j'$ quando $c_j < 0$.

b) os $b_i \geq 0$ para $i = 1,\ldots,m$

c) os $a_{ij} \geq 0$ para $i = 1,\ldots,m$ e $j = 1,\ldots,n$

O problema de programação zero-um é considerado importante, por suas aplicações práticas e também devido a que todo problema de programação inteira com variáveis limitadas, pode ser transformado em um problema de programação zero-um.

4. Problema de Programação Zero-Um Equivalente ao Problema de Programação Inteira com Variáveis Limitadas

Consideremos o problema (2) onde as variáveis x_j inteiras possuem limites superiores conhecidos. Seja uma variável $x_k \le u_k$, x_k poderá ser expressada como

$$x_k = \sum_{p=0}^{K} 2^p y_{kp} \ , \ y_{kp} = 0 \text{ ou } 1$$

onde K será determinado como $2^{K+1} \ge u_{k+1}$. Substituindo x_k em função de y_{kp} temos o resultado desejado.

Como uma desvantagem desta transformação temos que, o problema resultante possuirá um grande número de variáveis zero-um, o que poderá dificultar a procura da solução do problema, devido à capacidade da memória do computador disponível ou, o tempo de computação que seria necessário para encontrar a solução poderá ser inaceitável.

5. Aplicações de Programação Zero-Um

Problemas de programação zero-um podem ser gerados a partir de problemas de programação inteira, programação não-linear, porém existem problemas cujo modelo pode ser definido diretamente como um problema de programação zero-um. Nesta classe de problemas se enquadram os problemas de orçamento de capital, seleção de projetos,alocação de recursos, fluxos em redes de comunicação, recuperação de informações, assignação de experimentos, localização industrial, dimensionamento de frotas de veículos, alocação de tripulação em linhas aéreas, sequenciamento, multicomodidade, o problema "cutting-stock", problema do caixeiro viajante, problemas de recobrimento (set covering), alocação de horários, problema Knapsack, alocação de tarefas, balanceamento de linhas de montagem, planejamento de vôos, seleção de jornais e revistas, etc. O problema knapsack (mochila) será abordado detalhadamente no Capítulo II , assim como alguns outros problemas similares.

6. Métodos de Solução

Teoricamente existe uma infinidade de métodos para resolver os problemas de programação zero-um, porém estão sempre relacionados com métodos que podem ser considerados padrões, já que nestes são introduzidas algumas mudanças a fim de torná-los mais eficientes ou adaptá-los a estruturas especiais da matriz de coeficientes. Basicamente esses métodos podem ser considerados como:

i) Métodos dos planos de corte

ii) Métodos que utilizam Teoria dos Grupos

iii) Métodos de "branch and bound"

iv) Métodos do subgradiente

v) Métodos heurísticos

vi) Métodos dos multiplicadores de Lagrange

vii) Métodos de programa dinâmica

viii) Métodos de enumeração implícita

Alguns algoritmos resolvem os problemas de programação zero-um utilizando um ou mais dos métodos acima.

Métodos de enumeração implícita

Em 1965 Balas [10] propôs o algoritmo aditivo que serviu de base para a apresentação de outros trabalhos, tais como os que foram desenvolvidos par Glover [51], Mao e Wallingford [74], Balas [11], Geoffrion [49], Petersen [83], Garfinkel e Nemhauser [43], [46], Cabot [20], Eilon e Christofides [26], De Maio e Roveda [23], Ellwein [25], Salkin e Koncal [89], Bradley e Wahi [15], Etcheberry [24], Arora [1] entre outros.

Métodos de Programação Dinâmica

Dentre os trabalhos que utilizam programação dinâmica podemos citar os de Weingartner e Ness [98], Nemhauser e Ullman [78], Shapiro e Wagner [93], Shapiro [90], Nemhauser e Yu [77], Kraft e Hill [65], Gearing, Swart e Var [38].

Métodos dos Planos de Corte

Alguns destes métodos foram desenvolvidos por: Bellmore e Ratliff [17], Salkin e Koncal [89], Glover e Klingman [40], Fisher e Shapiro [29], Chames, D.Granot e F.Granot [21], Balas e Zemel [4], [5], Kianfar [64] e Balas [2].

Métodos que utilizam Teoria dos Grupos

Cada vez mais esta técnica está sendo divulgada, sen

vii) Métodos de programa dinâmica

viii) Métodos de enumeração implícita

Alguns algoritmos resolvem os problemas de programação zero-um utilizando um ou mais dos métodos acima.

. Métodos de enumeração implícita

Em 1965 Balas [10] propôs o algoritmo aditivo que serviu de base para a apresentação de outros trabalhos, tais como os que foram desenvolvidos por Glover [51], Mao e Wallingford [74], Balas [11], Geoffrion [49], Petersen [83], Garfinkel e Nemhauser [43], [46], Cabot [20], Eilon e Christofides [26], De Maio e Roveda [23], Ellwein [25], Salkin e Koncal [89], Bradley e Wahi [15], Etcheberry [24], Arora [1] entre outros.

. Métodos de Programação Dinâmica

Dentre os trabalhos que utilizam programação dinâmica podemos citar os de Weingartner e Ness [98], Nemhauser e Ullman [78], Shapiro e Wagner [93], Shapiro [90], Nemhauser e Yu [77], Kraft e Hill [65], Gearing, Swart e Var [38].

. Métodos dos Planos de Corte

Alguns destes métodos foram desenvolvidos por: Bellmore e Ratliff [17], Salkin e Koncal [89], Glover e Klingman [40], Fisher e Shapiro [29], Charnes, D.Granot e F.Granot [21], Balas e Zemel [4], [5], Kianfar [64] e Balas [2].

. Métodos que utilizam Teoria dos Grupos

Cada vez mais esta técnica está sendo divulgada, sen do que pode ser apontado como um dos trabalhos pioneiros o de Shapiro [91], ao qual se seguiram os trabalhos de Bell [13], Houck e Vemuganti [56], Wosley [99], Gorry, Northup e Shapiro [37].

Métodos de "Branch and Bound"

Métodos especificamente desenvolvidos para programação zero-um e que utilizam a técnica "Branch and Bound" estão descritos em: Kolesar [67], Greenberg e Herich [47], Lernke-Salkin e Spielberg [71], Held e Karp [59], Fisher e Shapiro [29], Etcheberry [24], Pierce [82], Shih [87], Nauss [76] e Ingariola e Korch [62].

Métodos do Subgradiente

Entre estes métodos temos os desenvolvidos por Held, Wolfe o Crowder [58], Etcheberry [24], Golden [35], Legendre e Minoux [70].

Métodos Heurísticos

Devido a dificuldade em encontrar soluções ótimas para problemas com um grande número de variáveis e restrições, cada vez mais estão sendo utilizados métodos heurísticos que fornecem soluções próximas da ótima com um esforço computacional muito reduzido se comparado com o tempo necessário para achar soluções ótimas por outros métodos. Entre estes métodos podemos anotar os apresentados por Hiller [60], Eilon e Christofides [26], Salkin e Koncal [89], Magazine, Nemhauser e Trotter [73], Hu e Lenard [57], Zanakis [100], Pierce [82] e muito recentemente o trabalho de Balas e Martin [3].

Outros métodos que sem ser heurísticos fornecem soluções aproximadas do problema são os desenvolvidos por Senju e Toyoda [92], Toyoda [96], Soyster-Ley e Slivka [86].

Métodos dos Multiplicadores de Lagrange

Pode-se afirmar que o trabalho que deu início à utilização do método dos multiplicadores de Lagrange foi o de Everett [27] em 1963 e a este se seguiram os trabalhos de Fisher [28], Bell [13], Kaplan [68], Brooks e Geoffrion [18], Greenberg [34], Fox e Landi [31], Legendre e Minoux [70]. Esses métodos se preocupam em calcular os multiplicadores para o problema irrestrito em que é transformado o problema original.

CAPÍTULO II

1. O Problema Knapsack

Será descrita aqui uma classe de problemas de programação zero-um chamados algumas vezes de problemas combinatórios. Estes problemas possuem um grande número de soluções viáveis e poderão ser resolvidos mediante enumeração e cálculo das sobuções viáveis.

Entre estes problemas estão os problemas Knapsack com uma única restrição e o problema Knapsack multidimensional. Existem na literatura diversos métodos para resolver esta classe de problemas porém aqui pretendemos discutir a sua solução como um problema de programação zero-um.

2. O Problema Knapsack Unidimensional

O nome do problema, Knapsack, se refere a uma situação em que se quer carregar uma mochila que possa suportar um peão máximo. Para encher a mochila se selecionam vários itens cada um dos quais possue um valor e um peso, O problema consiste em encher a mochila de tal forma que sua capacidade não seja excedida e o valor total dos itens, isto é, a soma de todos os valores relativos, seja maximizada.

O problema será apresentado como:

$$\text{Maximizar } z = \sum_{j=1}^{n} c_j x_j$$

sujeito a (1)

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j \geq 0 \quad \text{e inteiro } (j=1,\ldots,n)$$

onde

c_j é o valor de cada item j

a_j é o peso do item j

b é o peso que a mochila pode suportar

Sem perder generalidade pode-se assumir que os $c_j \geq 0$ $(j = 1,\ldots,n)$, $b \geq 0$ e $a_j \geq 0$. Porém, se $c_j = 0$ implica que $x_j = 0$ em toda solução ótima, se $b = 0$ implica que a única solução viável é $x_j = 0$ $(j = 1,\ldots,n)$ desde que $a_j > 0$.

Se algum $a_j = 0$ com $c_j > 0$, a solução ótima é ilimitada desde que x_j possa ser qualquer valor inteiro arbitrariamente grande,

O problema que se pretende abordar aqui é o problema Knapsack em que só pode ser carregado um elemento de cada item, isto é, $x_j = 1$ se o elemento é carregado e $x_j = 0$, caso contrário.

O problema Knapsack tem sido estudado com muito interesse por poder representar muitos problemas reais, tais como: alocação de recursos, seleção de projetos, investimentosde capital, problemas de carga, seleção de jornais e revistas, o problema "cutting stock", e também aparece com sub-problema de alguns algoritmos de programação inteira.

3, O Problema Knapsack Multidimensional

O problema Knapsack multidimensional, é o problema Knapsack descrito em (2) com outras limitações adicionais assim por exemplo, limitação de peso, volume, largura ou comprimento. As variáveis poderão ser inteiras, porém assumimos aqui que estas variáveis só poderão ser 0 ou 1, considerando que não existe a possibilidade de incluir mais do que um artigo.

Um tipo de aplicação para esta classe de problemas é o problema de alocação de capital em que os melhores projetos de investimentos são selecionados sujeitos a limitações de gastos ou despesas em diversos períodos de tempo ou a limitações de várias entradas (de capital). Em algumas aplicações de alocação de recursos onde as constantes do segundo membro não são rígidas pode-se analisar o problema mediante o método que no Capítulo III será apresentado.

O modelo pode ser descrito como segue:

$$\text{Maximizar } z = \sum_{j=1}^{n} c_j x_j$$

sujeito a (2)

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad (i = 1,\ldots,m)$$

$$0 \leq x_j \leq 1$$

$$x_j \text{ inteiro}$$

Sendo c_j a receita devido à inclusão do projeto j,

a_{ij} é a quantidade de recurso requerido pelo projeto j no período i

b_i a quantidade de recurso disponível para ser gas to no período i

$x_j = 1$ se o projeto é aceito

$x_j = 0$ se o projeto é rejeitado

Se assome também que os a_{ij} são não negativos

4. O Problema de Alocação de Recursos

O problema de alocação de recursos se relaciona com a avaliação de uma variedade de projetos de investimento e seu objetivo é selecionar os melhores de um conjunto de projetos sob a condição de ter disponibilidade de recursos limitados.

O problema poderá ser estabelecido para um período ou para vários períodos de planejamento. Existe uma grande variedade de formulações de problemas de alocação de recursos. Talvez a fonte principal destas diversas formulações é a definição da função objetivo, dado que em geral este problema não é muito determinístico. Alguns pesquisadores acham que a utilização do retorno esperado não é suficiente como um critério objetivo dado que ignora a dispersão em torno da média.

O modelo é formulado da seguinte forma:

$$\text{Maximizar } E(z) = \sum_{j=1}^{n} E(c_j)\, x_j$$

sujeito a

$$\sum_{j=1}^{n} E(a_{ij})\, x_j \leq E(b_i) \qquad i = 0, 1, 2, \ldots, m$$

$$x_j = 0 \text{ ou } 1 \qquad j = 1, \ldots, n$$

onde:

$x_j = 1$ se o j-ésimo projeto é aceito e $x_j = 0$ se o j-ésimo projeto é rejeitado.

c_j é uma variável aleatória que representa o valor atual do j-ésimo projeto de investimento e $E(c_j)$ é seu valor esperado.

a_{ij} é a variável aleatória que representa o custo do j-ésimo projeto de investimento no i-ésimo período e $E(a_{ij})$ é seu valor esperado.

b_i é a variável aleatória representando o valor disponível no i-ésimo período para alocação dos projetos de investimento e $E(b_i)$ é seu valor esperado.

Se assume que as funções de densidade das variáveis aleatórias podem ser avaliadas.

O modelo também poderá ser formulado substituindo cada variável aleatória por seu valor esperado.

5. Métodos de Solução para o Problema Knapsack

O número de soluções para a classe de problemas apresentados nos ítens (2), (3) e (4) e da ordem de 2^n, onde n é o número de variáveis do problema.

Nesse caso a utilização do esquema enumerativo das soluções puro e simples, para selecionar a melhor solução se torna inviável. Existe na literatura uma grande quantidade de métodos que resolvem estes problemas de forma eficiente. Estes métodos foram desenvolvidos utilizando programação dinâmica, técnicas de "branch and bound", enumeração implícita, planos de corte, técnicas heurísticas e os que utilizam dualidade em programação inteira e entre os quais se enquadra o método que será apresentado no Capítulo III.

Podemos classificar os métodos para resolver o proble ma Knapsack em 2 grandes grupos, aqueles que resolvem o problema Knapsack unidimensional, e os que resolvem o problema . Knapsack multidimensional. Ainda dentro destas duas classes existem os métodos que consideram as variáveis inteiras ($x_j \geq 0$ e inteira) e aqueles em que as variáveis só podem assumir valores zero ou um. Particularmente desenvolveremos um método que resolverá o proble ma Knapsack multidimensional com variáveis zero-um, sendo que o método poderá resolver também o problema Knapsack unidimensional.

Dentre os métodos que utilizam as técnicas acima indi cadas podem citar:

Programação dinâmica: Weingartner e Ness [98] , Shapi ro e Wagner [93], Shapiro [90] e Nemhauser e Ullman [78].

Enumeração Implícita: Petersen [83], Mao e Wallinford [74], Pierce [82] e Cabot [20].

Técnicas de "Branch and Bound": Kolesar [67],e Greenber e Hegerich [47].

Planos de corte: Kianfar [64].

Heurísticos: Magazine, Nemhauser e Trotter [73], Hu e Lenard [57] e Balas e Martin [3].

Dualidade: Nauss [76], Legendre e Minoux [70]; Maculan, Vaca Obando e Villares [72].

Outros métodos: Senju e Toyoda [92], Toyoda [96] e Soyster, Lev e Slivka [86].

6. Redução de um Problema Multidimensional num Problema Unidimensional

Um problema Knapsack multidimensional pode ser reduzido a um problema Knapsack equivalente com uma restrição. como se poderá ver posteriormente, a procura da solução de um problema com uma restrição se torna mais fácil, razão pela que em alguns problemas poderá ser utilizado este tipo de transformação.

Consideremos o conjunto de restrições

$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i \qquad i = 1, \ldots, m$$

$$0 \le x_j \le u_j \qquad j = 1, \ldots, n \qquad (3)$$

$$x_j \text{ inteiro} \qquad j = 1, \ldots, n$$

onde a_{ij}, b_i e u_j são positivos e inteiros.

O problema consiste em encontrar valores $w_1, \ldots, w_m$ tais que toda solução não-negativa e inteira para o conjunto de restrições

$$\sum_{j=1}^{n} a_j\, x_j = b \qquad (4)$$

$$0 \le x_j \le u_j$$

$$x_j \text{ inteiro}$$

onde $a_j = \sum_{i=1}^{m} w_i\, a_{ij}$ $(j = 1, \ldots, n)$ e $b = \sum_{j=1}^{m} w_i b_i$ é uma solução para (3). Dado que as restrições (3) implicam em (4), toda solução inteira não-negativa para (3) é uma solução para (4). Porém para valores arbitrários dos w_i, o conjunto de inteiros não-negativos $X = (x_1, \ldots, x_n)$ que satisfazem (41 é geralmente maior que o conjunto que satisfaz (3).

Exemplo 1 - Balas [11]

O problema

Minimizar $z = 5x_1 \; 7x_2 + 10x_3 + 3x_4 + x_5$

sujeito a

$$- x_1 + 3x_2 - 5x_3 - x_4 + 4x_5 + x_6 = -2 \quad (i)$$

$$2x_1 - 6x_2 + 3x_3 + 2x_4 - 2x_5 + x_7 = 0 \quad (ii)$$

$$x_2 - 2x_3 + x_4 + x_5 + x_8 = -1 \quad (iii)$$

$$x_j = 0 \text{ ou } 1 \; (j = 1, \ldots, 5)$$

$$x_j \geq 0 \text{ e inteiro } (j = 6, 7, 8)$$

possue como soluções viáveis (0,1,1,0,0) e (1,1,1,0,0,). Se selecionamos arbitrariamente os valores $w_1 = w_2 = w_3 = 1$ as três equações somadas fornecem a equação

$$x_1 - 2x_2 - 4x_3 + 2x_4 + 3x_5 + x_6 + x_7 + x_8 = -3$$

que possue como soluções viáveis as duas anteriores mais outras 5 que são (0,0,1,0,0), (0,1,1,0,1), (0,1,1,1,0), (1,0,1,0,0) e (1,1,1,1,0).

6.1 - Métodos de Transformação

Existem na literatura um grande número de trabalhos relativos à agregação de restrições. Alguns destes métodos combinam duas restrições em uma que logo combinam com a terceira e assim sucessivamente até que todas as restrições sejam combinadas.

Um trabalho pioneiro neste sentido é o de Mathews (1) que com seu Teorema mostra como adicionar duas restrições de tal forma que o conjunto de soluções inteiras não-negativasnão aumente. Isto permite reduzir qualquer conjunto de m restrições a uma Única restrição.

(1) Para maiores detalhes ver Salkin [85]

Teorema de Mathews[1]

Considere o sistema de duas equações lineares[2]

$$S_1 \equiv \sum_{j=1}^{n} a_{1j}\, x_j = b_1 \qquad \text{(i)}$$

$$S_2 \equiv \sum_{j=1}^{n} a_{2j} x_j = b_2 \qquad \text{(ii)}$$

Os coeficientes a_{ij} $(i = 1, 2, j = 1, \ldots , n)$ são estritamente positivos e inteiros.

a) se existem valores não-negativos $x_1, \ldots, x_n$ que satisfazem (i) e (ii) logo

$b_2\, a_{1j}/a_{2j} \geq b_1$ ao menos para um j $(1 \leq j \leq n)$

b) se w é qualquer inteiro positivo tal que

$w \,r\, b_2 \max_{3} \{a_{1j}/a_{2j}\}$

logo o conjunto de soluções de (i) e (ii) é o mesmo que a equação

$$S_1 + wS_2 = b_1 + wb_2 \quad \text{(iii)}$$

Posteriormente[3], foi observado que o Teorema de Mathews pode ser utilizado para agregar um sistema de equações como as de (3) pela utilização das condições

$$S_1 + S_2 = b_1 + b_2 \qquad \text{(i)'}$$

$$S_1 + 2S_2 = b_1 + 2b_2 \qquad \text{(ii)'}$$

que Mathews indica também como aplicação do Teorema, onde as duas equações são substituidas por i ' e (ii)'. Isto é, as duas primeiras equações do sistema (3) são substituidas por (i)' e (ii)' e logo são agregadas. A equação resultante será a primeira equação do novo sistema, substituindo às duas primeiras, agora o sistema possue uma equação a menos. O processo se repete até obter uma equação só.

(2) O símbolo $\equiv$ significa que S_i toma o valor do somatório

(3) Ver Salkin [85]

Utilizando o problema do exemplo 1 é obtida a restrição equivalente ao sistema

$$32093\,x_1 - 95322x_2 + 7666x_3 + 33050x_4 + 1915x_5 +$$

$$+ 11017x_6 + 21555x_7 + 957x_8 = -\,22991$$

O conjunto de soluções viáveis desta equação é (0,1,1,0,0) (1,1,1,0,0)

A transformação de um sistema de equações numa equação só possue o incoveniente que os coeficientes crescem na sua ordem de grandeza o que muitas vezes impede utilizar o computador devido a sua capacidade limitada para armazenar valores inteiros especialmente. Devido a esta dificuldade alguns trabalhos tem sido publicados visando a transformação com coeficientes menores. Porém isto tem sido conseguido a custas de cálculos adicionais na transformação.

Glover[3] apresenta um método de redução agregando as restrições de duas em duas

Teorema de Glover

Considere o sistema de duas equações

$$S_1 \equiv \sum_{j=1}^{n} a_{1j}x_j = b_1 \qquad \text{(i)}$$

$$S_2 \equiv \sum_{j=1}^{n} a_{2j}x_j = b_2 \qquad \text{(ii)}$$

onde todos os coeficientes (a_{ij}, b_i) são inteiros e ao menos um dos b_i é diferente de zero. Sejam w_1 e w_2 inteiros diferentes de zero e primos entre si. Se existe ao menos uma solução inteira não-negativa para (i) e (ii), logo toda solução inteira não-nega tiva para

$$w_1 s_1 + w_2 s_2 = w_1 b_1 + w_2 b_2 \qquad \text{(iii)}$$

é uma solução inteira não negativa para (i) e (ii), e inversamente, dado que

$$w_1 a_{1j} + w_2 a_{2j} \geq \left| b_2 a_{1j} - b_1 a_{2j} \right| \qquad \text{(iv)}$$

para $j=1,\ldots, n$. (iv) é uma desigualdade estrita para j em J, onde J é qualquer conjunto não vazio de $\{1, \ldots, n\}$ tal que todas as soluções não-negativas para (iii) satisfazem $x_j > 0$ para ao menos um j em J.

Utilizando a redução de Glover o conjunto de restrições do Exemplo 1 tem como equivalente a restrição que segue

$$10x_1+73x_2-254x_3+113x_4+147x_5+18x_6+14x_7+103x_8 = -139$$

O conjunto de soluções viáveis da restrição é também (0,1,1,0,0) e (1,1,1,0,0).

6.1.1. Método de Kendall e Zionts

Num trabalho mais recente, Kendall e Zionts [63] apresentam dois métodos de transformação que reduzem o tamanho dos coeficientes da restrição equivalente.

Sejam duas restrições

$$\sum_{j=1}^{n} a_{1j} x_j = b_1 \qquad \text{(i)}$$

$$\sum_{j=1}^{n} a_{2j} x_j = b_2 \qquad \text{(ii)}$$

$$0 \leq x_j \leq u_j$$

x_j inteiro.

Se define uma função

$$g_i(x) = b_i - \sum_{j=1}^{n} a_{ij} x_j$$

para qualquer conjunto de valores x_j inteiros e $0 \leq x_j < u_j$. O vetor X é uma solução para a i-ésima restrição se e somente se $g_i(x) = 0$.

Um conjunto de multiplicadores w_1 e w_2 gera uma equação

$$w_1 g_1(x) + w_2 g_2(x) = 0 \qquad \text{(iii)}$$

Somente são de interesse os multiplicadores para os quais a equação (iii) implica que

$$g_1(x) = g_2(x) = 0$$

Expressando (iii) em outra forma temos

$$g_1(x) = (-w_2/w_1)\, g_2(x) \qquad \text{(iv)}$$

e utilizando os limites

$$b_i - \sum_{j=1}^{n} a_{ij}^{+} u_j \leq g_i(x) \leq b_i - \sum_{j=1}^{n} a_{ij}^{-} u_j$$

onde $a_{ij}^{+} = \max\left\{a_{ij}, 0\right\}$ e $a_{ij}^{-} = \min\left\{a_{ij}, 0\right\}$, temos que $w_1 = 1$ e

$$w_2 > \max\left\{b_1 - \sum_{j=1}^{n} a_{ij}^{-} u_j, \; - b_1 + \sum_{j=1}^{n} a_{ij}^{+} u_j\right\}$$

é um conjunto de multiplicadores.

O Teorema 3 fornece um conjunto de multiplicadores.

Teorema 3. (Kendall e Zionts [63]).

Se $0 \leq x_j \leq u_j$ e inteiros, as restrições

$$\sum_{j=1}^{n} a_{1j}x_j = b_1 \qquad (g_1(x) = 0) \qquad \text{(v)}$$

e

$$\sum_{j=1}^{n} a_{2j}x_j = b_2 \qquad (g_2(x) = 0) \qquad \text{(vi)}$$

são equivalentes a

$$w_1 g_1(x) + w_2 g_2(x) = 0 \qquad \text{(vii)}$$

se os multiplicadores w_i satistazem as seguintes condições:

a) Para qualquer valor da solução inteira $0 \leq x_j \leq u_j$, $j = 1,\ldots n$ $g_2(x)$ não é divisível por w_1 e $g_1(x)$ não é divisível por w_2

b) w_1 e w_2 são primos entre si (o maior divisor comum é um)

Prova

$$w_1 g_1(x) = 0$$

$$w_2 g_2(x) = 0$$

portanto $\quad w_1 g_1(x) + w_2 g_2(x) = 0$

Para provar que (vii) implica (vi) e (v), escreve-se (vii) como

$$g_1(x) = (-w_2/w_1)\, g_2(x)$$

Por definição o membro da esquerda da equação é inteiro, consequentemente o membro da direita é inteiro. Pela condição (b) $g_2(x)/w_1$ e $g_1(x)/w_2$ são inteiros. Porém isto contradiz a condição (a), portanto $g_1(x) = g_2(x) = 0$. ▨

Em principio o Teorema 3 pode ser utilizado junto com a condição de que:

$$g_1(x) \neq (-w2/w1)\ g_2(x) \qquad \text{(viii)}$$

a menos que $g_1(x)$ e $g_2(x)$ sejam zero. Assim o Teorema junto com (viii) fornecem uma sistemática de enumeração para encontrar bons multiplicadores. Para um conjunto de multiplicadores primos entre si, poderemos primeiro ver se todas as soluções satisfazem a condição (a) do Teorema 3. Se a condição (a) não é satisfeita, os multiplicadores serão válidos se podemos mostrar que (viii) é satisfeita.

Alguns métodos propostos para calcular os multiplicadores w_1 e w_2, utilizam valores primos entre si fora dos intervalos especificados pelos limites superior e inferior de cada equação. Esses métodos também satisfazem as condições (a) e (b).

Kendall e Zionts [63] propõem que multiplicadores menores que aqueles gerados por outros métodos podem ser encontrados eficientemente mediante a obtenção de valores de $g_1(x)$ e $g_2(x)$ que não estão entre os limites de $g_1(x)$ e $g_2(x)$ especificados para cada equação. Estes multiplicadores w_1 e w_2 serão válidos dado que satisfazem as condições (a) e (b).

O primeiro método proposto consiste em escolher os multiplicadores

$$w_1 > b_2 - \sum_{j=1}^{n} a_{2j}u_j - \min_{a_{2j}\neq 0}\left\{|a_{2j}|\right\} \quad \text{e}$$

$$w_2 > b_1 - \sum_{j=1}^{n} a_{1j}u_j - \min_{a_{1j}\neq 0}\left\{|a_{1j}|\right\}$$

tais que

$$(b_2 - \sum_{j=1}^{n} a_{2j}^{-} u_j)/w_1 \qquad e$$

$$(b_1 - \sum_{j=1}^{n} a_{1j}^{-} u_j)/w_2$$

não sejam inteiros e w_1 e w_2 sejam primos entre si

O segundo método consiste em

a) Enumerar uns poucos valores de

$$g_i(x) = b_i - \sum_{j=1}^{n} a_{ij} x_j \qquad \text{para } 0 \leq x_j \leq u_j$$

iniciando com a solução, $x_j = u_j$ se $a_{ij} < 0$ e $x_j = 0$ em caso contrário, assim que $g_i(x)$ decresce estritamente e na sequência não são omitidos valores. Também a enumeração pode iniciar a partir do limite inferior

$$b_i - \sum_{j=1}^{n} a_{ij}^{+} u_j$$

Os primeiros valores são facilmente enumeráveis porque incrementamos ou diminuimos a variável cujo $|a_{ij}|$ é mínimo e assim sucessivamente, utilizando um esquema ordenado lexicograficamente.

b) Se escolhe w_1 maior que o menor valor de $g_2(x)$ enumerado e w_2 maior que o menor valor de $g_1(x)$ enumerado, $w_1 \neq$ qualquer $g_2(x)$, $w_2 \neq$ qualquer $g_1(x)$, w_1 e w_2 são primos entre si, e

$$w_1 > (b_2 - \sum_{j=1}^{n} a_{2j}^{-} u_j)/2 \qquad e$$

$$w_2 > (b_1 - \sum_{j=1}^{n} a_{1j}^{-} u_j)/2$$

Utilizando o segundo método e as restrições do Exemplo 1 calculamos w_1 e w_2 para as duas primeiras restrições.

De (a) calculamos alguns valores de g1(x) e g2(x). Os primeiros valores de $g_1(x)$ e $g_2(x)$ serão

$$g_1(x) = -2 - (-1 - 5 - 1) = 5$$

$$g_2(x) = 0 - (2 - 6 + 3 + 2 - 2 + 1) = 8$$

os valores seguintes são calculados diminuindo o $|a_{ij}|$ mínimo ao primeiro valor de $g_i(x)$.

$g_1(x)$	5	4	2	1	
$g_2(x)$	8	7	6	5	2

de onde $w_2 > 1$, porém por (b) deverá ser $w_2 > 5/2$, isto é $w_2 = 3$ e $w_1 > 2$, porem por (b) deverá ser $w_1 > 8/2$ e $w_1 \neq$ de qualquer $g_2(x)$, então $w_1 = 9$. Temos que 3 e 9 não são primos, então $w_2 = 6$; novamente 9 e 6 não são primos. Finalmente $w_2 = 7$ e $w_1 = 9$ satisfazem todas as condições,

Multiplicando a equação (i) por 9, a equação (ii) por (7) e somando temos a equação (i)' seguinte

$$5x_1-15x_2-24x_3+5x_4+22x_5+9x_6+7x_7 = -18 \qquad (i)'$$

Enumerando valores para $g_3(x)$ e $g_{1'}(x)$ temos

$g_{1'}(x)$	21	16	14	12	11
$g_3(x)$	1	0	-1	-2	

os valores de $w_{1'}$ e w_3 serão $w_3 = 2$ e $w_{1'} = 13$ e a equação resultante será

$$10x_1-17x_2-74x_3+23x_4+57x_5+18x_6+14x_7+13x_8 = -49$$

É necessário observar que a ordem em que as restrições são combinadas fornece resultados diferentes podendo fornecer coeficientes maiores ou menores de acordo com a combinação escolhida, aqui pode-se recomendar a necessidade de um estudo prévio a fim de obter valores menores para os coeficientes.

6.2. Dificuldades e vantagens

Todos os métodos propostos até agora, para reduzir um problema Knapsaek multidimensional a um unidimensional, possuem a mesma desvantagem : a ordem de grandeza dos coeficientes resultantes aumenta enormemente, e que por ser inteiros poderão acarretar dificuldades em sua manipulação quando introduzidos no computador, para cálculo da solução.

Entre as vantagens que estas transformações proporcionam, temos que é muito mais facil resolver um problema com uma restrição, acrescentando-se o fato de que a memória do computador, necessária para armazenar esses coeficientes será reduzida em comparação à que se necessitaria caso se queira resolver o problema original. Também, não é necessário considerar a função objetivo se não somente na etapa de solução do problema.

CAPÍTULO III

1. Introdução

Será apresentado neste capítulo um procedimento para resolver o problema de programação zero-um, mediante a solução de um programa mestre que será obtido a partir do desenvolvimento de dualidade em programação inteira.

As idéias utilizadas para a construção de dualidade em programação inteira são derivadas da teoria de dualidade em programação matemática e que é aplicável a qualquer problema de otimização definido num espaço vetorial dimensionalmente finito.

A seguir apresentaremos alguns resultados básicos da teoria de dualidade em programação não linear baseados nos trabalhos de Varaiya [97], Geoffrion [42] e Zanwill [101].

Seja a problema primal de programação matemática

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

sujeito a

$$g_i(x) \leq b_i \qquad i = 1, \ldots, m \qquad (1)$$

$$x \in X \subseteq R^n$$

onde $g_i: R^n \rightarrow R$, $i=1, \ldots, m$, são funções convexas

$f: R^n \rightarrow R$, é função côncava

X é um subconjunto convexo de R^n, e

b, um vetor com m componentes

Seja $v(b) = \sup\{f(x) \mid x \in X,\ g(x) \leq b\}$

se x^* é uma solução ótima de (1) $v(b^*) = f(x^*)$. Consideremos também o problema seguinte, Seja $u \in R^m$ e $u \geq 0$

$$\text{Maximizar} \quad f(x) - u'\{g(x) - b\}$$
$$\text{sujeito a} \qquad x \in X \tag{2}$$

e definindo

$$z(u) = \sup\{f(x) - u'\{f(x) - b\} \mid x \in X\} \tag{2'}$$

o problema

$$\text{Minimizar } z(u)$$
$$\text{sujeito a} \qquad u \geq 0 \tag{3}$$

será o dual de (1). Seja $z^* = \inf \{z(u) \mid u \geq 0\}$

Geralmente o conjunto X em (1) é igual a R^n. Porém algumas vezes é possível incluir algumas das restrições em X, assim o cálculo de z(u) por (2) e a solução do problema dual (3) é mais simples.

A seguir o Lema 1 mostra que a função custo do problema dual é convexa e o Lema 2 mostra que o valor ótimo do problema dual é sempre um limite superior para o valor ótimo do primal.

<u>Lema 1</u>. A função $z: R_+^m \to R \cup \{+\infty\}$ é uma função convexa. Onde $R_+^m = \{u \in R^m \mid u \geq 0\}$

<u>Lema 2</u>. (Dualidade fraca), Se x é viável para (1), isto é, satisfaz $g(x) \leq b$ e $x \in X$, e se $u \geq 0$, tem-se que

$$f(x) \leq \sup\{f(x) \mid x \in X,\ g(x) \leq b\} \leq \inf\{z(u) \mid u \geq 0\} \leq z(u) \tag{4}$$

O problema básico da Teoria da Dualidade é determinar condições sob as quais

$$\sup\{f(x) \mid x \in X,\ g(x) \leq b\} = \inf\{z(u) \mid u \geq 0\}$$

isto é $\quad v(b) = z^*$

Definição: Se diz que um par (x^*, u^*), $x^* \varepsilon X$ e $u^* \geq 0$ satisfaz as condições de otimalidade se:

x^* é uma solução ótima de (2) com $u = u^*$ (5)

x^* é viável para (1), isto é, $g_i(x^*) \leq b_i$ (6)

$u_i^* = 0$, quando $g_i(x^*) < b_i$, o que equivale a $u^*(g(x^*)-b) = 0$ (7)

u^* é chamado um vetor de preços ótimo se existe $x^* \varepsilon X$ tal que (x^*, u^*) satisfaz as condições de otimalidade.

Teorema 1. (suficiência). Se (x^*, u^*) satisfaz as condições de otimalidade, a solução x^* é uma solução ótima para o primal, e u^* é uma solução ótima para o dual e

$$\sup\{f(x) \mid x \varepsilon X, \; g(x) \leq b\} = \inf \{z(u) \mid u \geq 0\}$$

2. Dualidade em Programação Inteira

Da teoria de dualidade em programação matemática podemos derivar algumas idéias para o problema de programação inteira e neste caso, particularmente para programação zero-um.

Seja o problema primal

Maximizar cx

sujeito a $Ax \leq b$ (8)

$x \varepsilon \{0,1\}^n$

onde:

c é um vetor linha com n componentes,

x é um vetor coluna com n componentes,

A é uma matriz com m linhas e n colunas,

b é um vetor coluna com m componentes

A partir de (8) se define outro problema

$$\text{Maximizar } \{cx - u(Ax - b)\} \quad (9)$$

sujeito a

$$x \in \{0,1\}^n$$

onde u é um vetor linha com m componentes não-negativas ($u \geq 0$), ou seu equivalente:

$$z(u) = \max \{cx - u\,(Ax - b) \mid x \in \{0,1\}^n\} \quad (9)'$$

então o problema

$$\text{Minimizar } z\,(u) \quad (10)$$

sujeito a $u \geq 0$

será o <u>dual</u> de (8)

Dos resultados o tidos por Everett [25] podemos extrair o seguinte teorema.

<u>Teorema 2</u>: Considerando o problema (8), se x^* é a solução ótima do problema (9), então x^* é também solução ótima do problema se b, substituído por Ax^*. Portanto se u é calculado de modo que para a solução ótima x^*, $b = Ax^*$, o problema (8) tem x^* como solução ótima.

<u>Demonstração</u>

x^* fornece o valor máximo de $cx - u\,(Ax - b)$, então $cx - u\,(Ax - b) \leq cx^* - u(Ax^* - b)$

assim para todo $x \in \{0,1\}^n$

$$cx + u\,(Ax^* - b - Ax + b) \leq cx^*$$

esta desigualdade é verdadeira para $Ax \leq Ax^*$. Porém para estas soluções $Ax^* - Ax \geq 0$ e portanto $cx \leq cx^*$, desde que $u \geq 0$. Por tanto x^* é solução ótima do problema

Maximizar cx

sujeito a $Ax \leq Ax^*$

$x \in \{0,1\}^n$ ▨

O problema (8) poderá ser resolvido mediante a inclusão das restrições na função objetivo. Porém selecionar $u \geq 0$ que satisfaçam $b = Ax^*$ se torna difícil e em muitos casos esses valores de u poderão não existir, assim valores de $u \geq 0$ para os quais Ax^* se aproxima de b podem ser utilizados, então x^* fornecerá uma solução próxima da ótima e que pode ser considerada como uma "boa" solução. Um outro problema consiste em como avaliar os $u \geq 0$. Um procedimento geral consistirá em calcular um vetor $u \geq 0$ e posteriormente calcular x^* e Ax^*, se este valor está próximo de b o processo para, em caso contrario outro vetor $u \geq 0$ será selecionado e assim por diante até encontrar valores de Ax^* aceitáveis ou concluir que estes não existem,

3. Propriedades dos Problemas Primal e Dual

Propriedade 1. A função z: $R^m_+ \to R$ é uma função convexa, onde R é o conjunto dos números reais e $R^m_+ = \{u \in R^m \mid u \geq 0\}$ tal que $z(u) = \text{máximo } \{cx - u(Ax - b) \mid x \in \{0,1\}^n\}$

Demonstração

Seja u^0 e u^1 quaisquer, tal que $u^0, u^1 \in R^m_+$, $1 \geq \lambda \geq 0$ e $x \in \{0,1\}^n$

$$z(\lambda u^0 + (1-\lambda)u^1) = \max \{cx - (\lambda u^0 + (1-\lambda)u^1)(Ax-b)\}$$

$$= \max \{cx - \lambda u^0(Ax-b) - (1-\lambda)u^1(Ax-b) + \lambda cx - \lambda cx\}$$

$$= \max \{\lambda cx - \lambda u^0(Ax-b) + (1-\lambda)cx - (1-\lambda)u^1(Ax-b)\}$$

$$\leq \lambda \max \{cx - u^0(Ax-b)\} + (1-\lambda)\max\{cx-u^1(Ax-b)\}$$

$$\leq \lambda z(u^0) + (1-\lambda)z(u^1) \quad ▨$$

Propriedade 2. Se x é uma solução viável de (8), isto é, x satisfaz $Ax \leq b$ e $x \varepsilon \{0,1\}^n$, e se $u \geq 0$, então $cx \leq z(u)$.

Demonstração

Dado que $Ax - b \leq 0$ e $u \geq 0$, temos que $u(Ax-b) \leq 0$ e $cx \leq cx - u(Ax-b)$ para todos os x viáveis de (1) e todos os $u \geq 0$, logo

$$cx \leq \text{máximo} \{cx-u(Ax-b) \mid Ax-b \leq 0,\ x \varepsilon \{0,1\}^n\}$$

$$\leq \text{máximo} \{cx-u(Ax-b) \mid x \varepsilon \{0,1\}^n\} = z(u)$$

o que implica que $cx \leq z(u)$ □

Da propriedade 2 pode-se concluir que:

$$\max \{cx \mid Ax \leq b,\ x \varepsilon \{0,1\}^n\} \leq \min \{z(u) \mid u \geq 0\}, \qquad (11)$$

então se x^* for ótimo de (8) e u^* for Ótimo de (10) tem-se

$$cx^* \leq z(u^*)$$

Portanto é interessante resolver o problema (10) para que se possa calcular uma cota superior de cx em (8), dessa forma se pode ter uma idéia do valor de cx, para um x viável de (8) em relação a seu ótimo. Podem ser determinadas soluções próximas da ótima, consideradas "boas" para um certo tipo de problemas.

4. Considerações Sobre as Soluções Geradas

A seguir serão feitas algumas considerações, em termos de programação linear, sobre as soluções geradas pelo método do Dual de Decomposição em sua 1a. Fase.

Seja o Problema P

$$\max \sum_{j=1}^{n} c_j x_j \tag{12}$$

sujeito a

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad (i = 1,\dots,m) \tag{13}$$

$$0 \leq x_j \leq 1 \qquad (j = 1,\dots,n) \tag{14}$$

O método Dual de Decomposição na sua 1a. Fase pode produzir soluções ótimas para o problema (8) somente se estas soluções são pontos extremos de um conjunto convexo de equações lineares.

Assim P pode ser resolvido utilizando o método Simplex e (8) poderá ser resolvido utilizando o método Dual de Decomposição (1a. Fase). Porem para um vetor b dado, se no problema P não é obtida uma solução inteira, não poderá ser obtida uma solução ótima para (8) utilizando o método Dual de Decomposição. Isto é, as únicas soluções inteiras que podem ser achadas utilizando o método Dual de Decomposição são os pontos extremos do conjunto convexo do problema P. Não seria recomendada a utilização do método Dual de Decomposição caso se queira achar uma solução ótima com um b específico. Porém se é desejado obter uma solução ou soluções ótimas para um $\bar{b}$ em torno de h, o método é recomendado.

Considerem-se as condições de otimalidade primal-dual para o problema P. Sejam α_i $(i = 1,\ldots,m)$ as variáveis duais que correspondem às restrições $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$, e $\gamma_j (j=1,\ldots,n)$ as variáveis duais que correspondem às restrições $0 \leq x_j \leq 1$. Então o problema dual de P, será

$$\min \sum_{i=1}^{m} \alpha_i b_i + \sum_{j=1}^{n} \gamma_j \tag{15}$$

sujeito a

$$\sum_{i=1}^{m} \alpha_i a_{ij} + \gamma_j \geq c_j \qquad j = 1,\ldots,n \tag{16}$$

$$\alpha_i \geq 0 \qquad i = 1,\ldots,m \tag{17}$$

$$\gamma_j \geq 0 \qquad j = 1,\ldots,n \tag{18}$$

Teorema 3. As soluções ótimas de (8) que possam ser geradas mediante o método Dual de Decomposição (1a. Fase) coincidem com as soluções totalmente inteiras de P

Demonstração

Assumamos que o método Dual de Decomposição (1a. Fase) gera uma solução ótima $\hat{x}$ para o problema (8) com $b = \hat{b}$. Então $\hat{x}$ é uma solução viável para P e satisfaz as condições (13) e (14). Seja $\beta_j = c_j - \sum_{i=1}^{m} u_i a_{ij}$. Portanto existe $u_i \geq 0$ tal que

$$\hat{x}_j = 1, \text{para } \beta_j > 0$$

$$\hat{x}_j = 0, \text{para } \beta_j < 0$$

$$\hat{x}_j = 0 \text{ ou } 1, \text{para } \beta_j = 0$$

$$\sum_{j=i}^{n} a_{ij}\hat{x}_j = b_i \quad \text{para} \quad u_i > 0, \text{ e}$$

$$\sum_{j=1}^{n} a_{ij}\hat{x}_j \leq b_i \quad \text{para } u_i = 0$$

Se escolhe um $\alpha_i = u_i$ tal que a condição (17) e a condição (Teorema das folgas complementares)

$$\alpha_i \left(\sum_{j=1}^{n} a_{ij} x_j - b_i \right) = 0 \qquad (19)$$

sejam satisfeitas, então de (16) ternos

$$\sum_{i=1}^{m} u_i a_{ij} + \gamma_j \geq c_j$$

$$\gamma_j \geq c_j - \sum_{i=1}^{m} u_i a_{ij}$$

$$\gamma_j \geq \beta_j$$

Seja a condição (Teorema das folgas complementares)

$$\hat{x}_j \left(c_j - \sum_{j=1}^{n} a_{ij} \alpha_i - \gamma_j \right) = 0 \qquad (20)$$

de onde

$$\hat{x}_j \left(\beta_j - \gamma_j \right) = 0$$

Tendo $\gamma_j = \max (0, \beta_j)$

as condições (16), (18), (20) e

$$\gamma_j \left(1 - x_j \right) = 0 \qquad (21)$$

são satisfeitas

Portants a solução totalmente inteira $\hat{x}$ ê uma solução ótima de P.

Agora, suponha que existe m a solução ōtima totalmente Inteira x* para o problema P, essa solução estar5 associada com um determinado α_i e γ_j. Seja

$$u_i = \alpha_i \geq 0$$

suponhamos que $x_3^* = 0$, então por (21), temos que $\gamma_j = 0$, e de (16) temos que $\gamma_i = \beta_j$ e de (18) se tem $\beta_j \geq 0$. Da condição (13) temos que $\sum_{j=1}^{n} a_i x_j^* \leq b_i$ e para $u_i \geq 0$ se tem de (19) que $\sum_{j=1}^{n} a_{ij} x_j^* = b_i$. Portanto x* pode ser gerada pelo método Dual de Decomposição. ▨

5. Resolução do Problema Irrestrito

Até agora mostramos como transformar o problema (8) em um problema irrestrito (9) onde as restrições sãs incluídas na função objetivo, porém nada foi dito ainda sobre a forma de calcular u para obter o máximo da função z(u).

O Teorema 2 garante que qualquer máximo da função irrestrita (9) necessariamente é uma solução do problema de máximo restrito para $Ax \leq Ax^*$.

5.1, Soluções ξ - viáveis

Em alguns casos os valores de $u \geq 0$ que fizerem $Ax - b \leq 0$ poderão não existir e neste caso existirão valores de u para os quais Ax* se aproximará de b; x* é considerada então uma "boa" solução e estas soluções serão chamadas de ξ-viáveis. ξ é um escalar $\xi > 0$ e é definido arbitrariamente. Assim uma solução será ξ - viável se satisfaz.

$$Ax^* - b \leq \xi\, b.$$

Um outro problema para achar a solução ótima de (9) está na avaliação das componentes de u.

6. Existência de "gaps"

O fato de existirem vetores restrição que não são gerados por qualquer vetor u se deve à existência de regiões inacessíveis, chamadas "gaps", ver Everett [27].

Considere-se o espaço da função objetivo vs. os níveis de restrições utilizados. Toda solução $x \in \{0,1\}^n$ corresponde a um ponto neste espaço, cx e Ax, então o problema é representado por um conjunto de pontos no espaço considerado. Assim o problema de achar o máximo da função objetivo cx sujeito às restrições $Ax \leq b$, é simplesmente o problema de escolher que ponto desse conjunto com o máximo de cx está contido no espaço.

Exemplo 1.

Seja o problema

$$\text{Max} \quad x_1+4x_2+2x_3+3x_4+5x_5+6x_6+8x_7+9x_8+7x_9$$

sujeito a

$$2x_1+6x_2+5x_3+8x_4+9x_5+7x_6+11x_7+12x_8+5x_9 \leq 30$$

$$x_j = 0 \text{ ou } 1 \qquad j = 1,\ldots,9$$

Se constroe um gráfico, Figura 1, cx. vs. Ax para as soluções possíveis,

o, a, b, c, d, e são soluções geradas por $u_o, u_a, u_b, u_c, u_d, u_e$.

Algumas destas soluções geradas serão inviáveis, e dentre as viáveis podemos escolher aquela que forneça o maior valor para a função objetivo, (ponto c), porém esta não será ótima. O ponto d corresponde a uma solução inviável. Assim sabemos que a solução ótima será maior ou igual que 22 (ponto c)

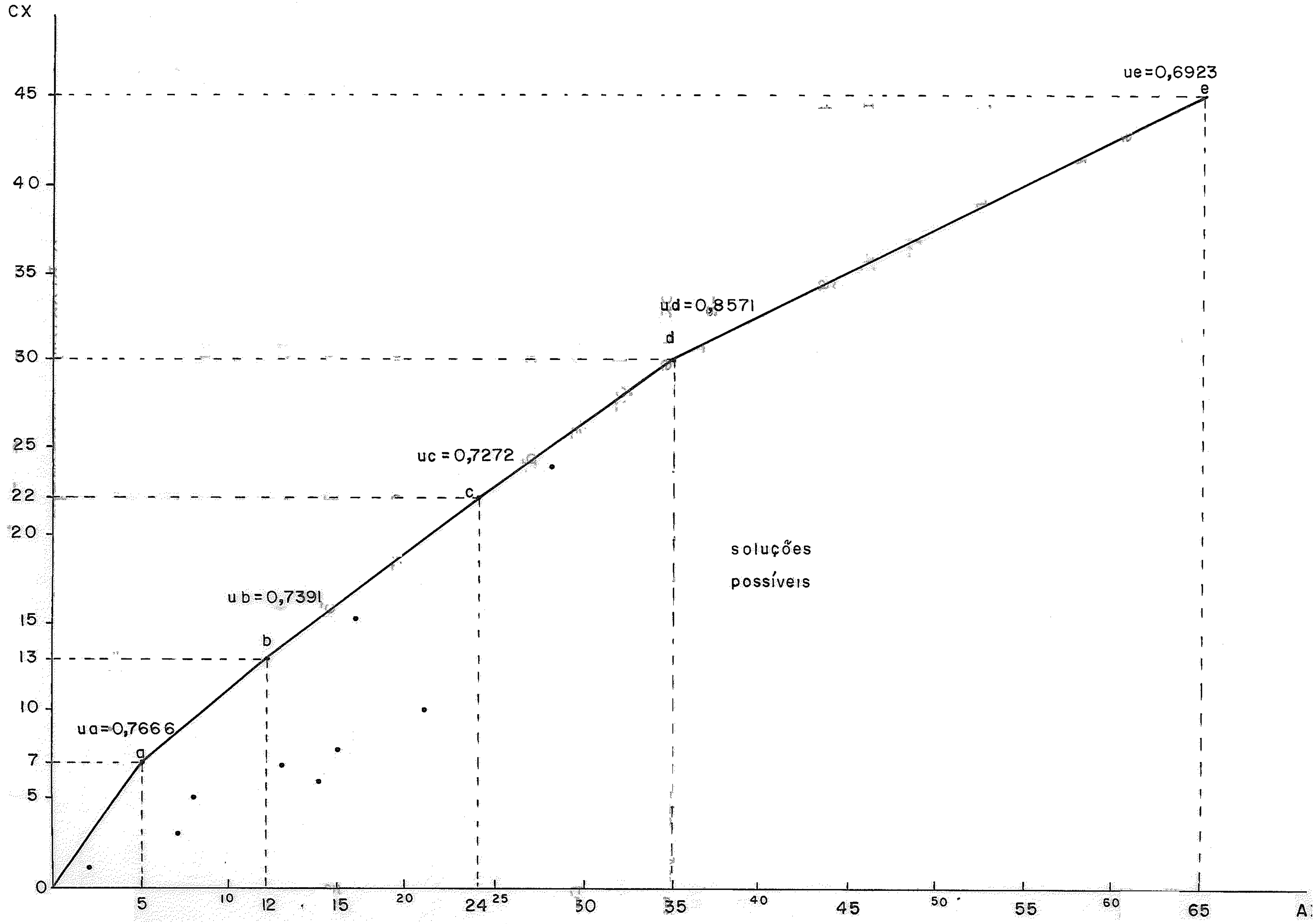

CX
45
40
35
30
25
22
20
15
13
10
7
5
0
5
10
12
15
20
24
25
30
35
40
45
50
55
60
65
A
ue=0,6923
e
ud=0,8571
d
uc = 0,7272
c
ub=0,7391
b
ua=0,7666
a
soluções
possíveis

FIGURA - I

e menor que 30 (ponto d).

Com o método a ser apresentado podemos gerar as soluções associadas aos u e posteriormente tentar melhorar a solução viável com maior valor para a função objetivo.

7. Avaliação de u

De acordo com o Teorema 2 pode-se escolher simplesmente um conjunto arbitrario de $u \geq 0$, achando um máximo irrestrito da função $z(u)$ e se pode obter a solução do problema restrito.

Pode-se proceder à avaliação de u por tentativas, calculando um vetor u e com este calcular x e Ax, se este Ax está suficientemente próximo de b o processo para, em caso contrário outro vetor u será selecionado e assim sucessivamente até achar u ou determinar que este não existe para o valor de ξ estimado,

7.1. Cálculo de u por inspeção. Ver Salkin [85]

Considere-se o problema Knapsack com uma restrição

$$\text{Maximizar } z = \sum_{j=1}^{n} c_j x_j$$

$$\text{sujeito a } \quad \sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j = 0 \text{ ou } 1 \quad (j = 1,\ldots,n)$$

O problema irrestrito será:

$$\text{Maximizar } \sum_{j=1}^{n} c_j x_j - u_1 \sum_{j=1}^{n} a_j x_j$$

sujeito a $x_j = 0$ ou 1 $(j = 1,\ldots,n)$

ou apresentado de outra forma

$$\text{Maximizar} \sum_{j=1}^{n} (c_j - u_1 a_j)\, x_j$$

$$x_j = 0 \text{ ou } 1 \quad (j = 1,\ldots,n)$$

O problema poderá ser resolvido por inspeção:

$$x_j^* = 1, \quad \text{se } c_j - u_1 a_j > 0$$

$$x_j^* = 1 \text{ ou } 0, \text{se } c_j - u_1 a_j = 0$$

$$x_j^* = 0, \quad \text{se } c_j - u_1 a_j < 0$$

A solução ótima depende do sinal de $(c_j - u_1 a_j)$ e não de sua magnitude. Para $c_j > 0$ e $a_j > 0$ se ordenam as variáveis de maneira que

$$c_j/a_j \geq c_{j+1}/a_{j+1} \qquad (j = 1,\ldots,n-1)$$

Dado que x* somente muda quando $c_j - u_1 a_j = 0$ ou $u > c_j/a_j$ o valor de x* permanece constante nos intervalos

$$0 \leq u_1 < c_j/a_j,\ c_j/a_j < u_1 < c_{j-1}/a_{j-1}, \ldots, c_2/a_2 < u_1 < c_1/a_1$$

Assim os valores de u_1 a serem testados são $0, c_n/a_n, c_{n-1}/a_{n-1}, \ldots, c_1/a_1$

7.2. Resolução do problema por biseção. Ver Greenberg [35]

Neste caso também será analisado o problema (8) com uma restrição. Para a procura do u no problema (9), se utilizará a biseção. Aqui também se procura um u que faça $\sum_{j=1}^{n} a_j x_j - b$ aproximadamente igual a zero. Na pratica pode-se escolher um valor $\xi > 0$ pequeno e a procura da solução ótima terminará quando seja achado um u que satisfaça

$$\sum_{j=1}^{n} a_j x_j - b \leq \xi b$$

Suponhamos que foram achados valores u_1 e u_2 tais que:

para u_1 temos $\sum_{j=1}^{n} a_j x_j > b$

para u_2 temos $\sum_{j=i}^{n} a_j x_j < b$

a biseção divide o intervalo u_1, u_2 achado, em duas partes. Em cada iteração, uma metade do intervalo anterior é retida. O processo termina na k-ésima iteração se o ponto médio u_k do intervalo retido satisfaz a condição da solução ser ξ - viável. **Em** caso que para u^k, $\sum_{i=1}^{n} a_j x_j > b$, a metade da direita é retirada e se $\Sigma a_j x_j < b$ a metade da esquerda é retirada,

Pode acontecer que para $\sum_{j=1}^{n} a_j x_j - b$ não exista um zero, e também é possível que o ξ especificado possa não ser atingido. Porém, dado que se trata de localizar um intervalo e não um ponto, a biseção em muitas aplicações é eficiente.

8. Método Dual de Decomposição

Utilizando as idéias, de dualidade em programação inteira, do método de decomposição de Dantzig-Wolfe e especialmente o método de geração de colunas será apresentado um método, que daqui por diante chamaremos de Método Dual de Decomposição, para resolver o problema (8).

Este método compreende duas fases: a primeira fase que corresponde as Método Dual de Decomposição propriamente dito e em que se procura uma solução viável, para o problema (8), que esteja próxima da ótima, gerando-se simultaneamente as soluções chamadas de ξ- viáveis, E a segunda fase que procura uma solução que se aproxima ainda mais da ótima quando nas restrições da solução viável calculada na primeira fase existem folgas que permitem fazer outras variáveis assumirem o valor 1. Para isto, se constroe um problema auxiliar, cada vez que seja necessário investigar, se mais uma variável pode assumir o valor 1.

8.1. Primeira Fase

Seja dado $u \geq o$, resolveremos então o problema

$$z(u) = \max \{cx-u(Ax-b)\}$$

sujeito a

$$x \in \{0,1\}^n$$

teremos agora o conjunto $X = \{x \in R^n \mid 0 \leq x \leq \eta\}$, snde $\eta = (1\ 1\ \ldots 1)^T$, facilmente verificamos que $\{0,1\}^n$ é o conjunto de todos os vértices de X. Poderemos então substituir o problema acima por:

$$z(u) = \max \{cx-u(Ax-b)\}$$

sujeito a $\qquad\qquad\qquad\qquad$ (22)

$$x \in X$$

pois (22) é um problema de programação linear, admitindo ao menos uma solução ótima em um vértice de X, ver Dantzig [22]. Como esse vértice de X é um ponto do conjunto $\{0,1\}^n$ e como $\{0,1\}^n \in X$, logo a solução de (22) é a mesma de (9)', Escreveremos (22) de outra maneira:

$z(u) = ub+\max(c-uA)x,$

sujeito a

$x \varepsilon X$

como o máximo se dará para um ponto extremo de X, seja então o conjunto $Y = \{x^1, x^2, .., x^p\}$, o conjunto de todos os vêrtices de X (lembraremos que $p = 2^n$), então

$$z(u) = ub+\text{máximo}(c-uA)x^k \quad ,k = 1,2,...,p \qquad (23)$$

Para encontrarmos o mínimo de $z(u)$ tal que $u \geq 0$, procederemos da seguinte maneira:

minimizar ω

sujeito a

$$\omega \geqq ub+(c-uA)x^k \;, \; k = 1,2,...,p \qquad (24)$$

$$u \geqq 0$$

Lembremos que os vetores x^k são considerados como dados para (24), caso o par (ω^*, u^*) seja o ótimo de (24), então $z(u^*)=\omega^*$, resolvendo dessa maneira (10), No entanto o problema (24) é um problema de muitas linhas, para resolvê-lo poderíamos optar por um método de geração de linhas, ver Lasdon [69]. Escolheremos uma outra maneira para o calculo de ω^* Colocaremos (24) sob outra forma:

minimizar $t = \omega$

sujeito a

$$\omega+u(Ax^k-b) \geq cx^k \;, \qquad k = 1,2,...,p \qquad (25)$$

$$u \geq 0$$

cujo dual será:

$$\text{maximizar } v = \sum_{k=1}^{p} cx^k \lambda_k \tag{26}$$

$$\sum_{k=1}^{p} \lambda_k = 1 \tag{27}$$

$$\sum_{k=1}^{p} (Ax^k - b)\lambda_k \leq 0 \tag{28}$$

$$\lambda_k \geq 0, \; k = 1, \ldots, p \tag{29}$$

Modificaremos a forma de (28)

$$\sum_{k=1}^{p} Ax^k \lambda_k \leq b \sum_{k=1}^{p} \lambda_k$$

como em (27). temos que $\sum_{k=1}^{p} \lambda_k = 1$, (28) se tornará finalmente

$$\sum_{k=1}^{p} Ax^k \lambda_k \leq b$$

O dual de (25) será apresentado então da seguinte forma:

$$\text{maximizar } v = \sum_{k=1}^{p} cx^k \lambda_k$$

$$\sum_{k=1}^{p} \lambda_k = 1 \tag{30}$$

$$\sum_{k=1}^{p} Ax^k \lambda_k \leq b$$

$$\lambda_k \geq 0, \; k = 1, 2, \ldots, p$$

O problema (30) terá inúmeras colunas ($p=2^n$), para encontrarmos sua solução será utilizado o método de decomposição do tipo de Dantzig-Wolfe [69],[70]. isto é partiremos utilizando o método revisado do Simplex [22], utilizando uma solu ção básica inicial, podendo ser artificial e geraremos a seguir colunas que entrarão na base em substituição de outras.

Introduzindo o vetor $s \in R_+^m$, cujos componentes representarão as variáveis de folga, transformaremos (30) na forma padrão de problemas de programação linear.

$$\text{maximizar } v = \sum_{k=1}^{p} cx^k \lambda_k \quad , \qquad \text{sujeito a}$$

$$\sum_{k=1}^{p} \lambda_k = 1$$

$$\sum_{k=1}^{p} Ax^k \lambda_k + s = b$$

$$\lambda_k \geq 0, \; k = 1, 2, \ldots, p$$

que sob forma matricial poderia ficar:

maximizar v

sujeito a

$$\begin{bmatrix} 1 & -cx^1 & -cx^2 & \ldots & -cx^p & 0 & 0 & \ldots & 0 \\ 0 & 1 & 1 & \ldots & 1 & 0 & 0 & \ldots & 0 \\ 0 & Ax^1 & Ax^2 & \ldots & Ax^p & e_1 & e_2 & & e_m \end{bmatrix} \begin{bmatrix} v \\ \lambda_1 \\ \lambda_2 \\ \cdot \\ \cdot \\ \cdot \\ \lambda_p \\ s_1 \\ s_2 \\ \cdot \\ \cdot \\ \cdot \\ s_m \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ b \end{bmatrix} \qquad (31)$$

A partir de uma solução básica viável, quando não for possível sua determinação imediata, começaremos com uma solução básica artificial, construiremos o quadro inicial do Simplex Revisado, seja B a matriz m+1 por m+1 associada à base inicial. c_B

vetor linha dos coeficientes (cx^k) de h_k e (0) de s_i na função objetivo, associados à B. Seja a_j vetor coluna tal que

$a_k = [1, Ax^k]$ ou $a_i = [o, e_i]$ e $\bar{b} = B^{-1}b$, seja $u = c_B B^{-1}$ e $\bar{v} = c_B \bar{b}$.

<u>Quadro Inicial</u>

$\bar{v}$	1	$u^0 \; u_1 \; u_2 \; \ldots\ldots u_m$	
$\bar{b}$	0	B^{-1}	

→ coluna onde será colocado o vetor que entrará na base, após atualização.

para saber se este quadro é ótimo, temos **que** ter $u_1 \geq 0$, $u_2 \geq 0, \ldots$ $u_m \geq 0$ *e* todos os $ua_j - cx^k \geq 0$ para todos os k associados aos λ_k não básicos, ver [70], Procederemos da seguinte maneira: Verificaremos se os $u_i \geq 0$ para $i = 1, \ldots, m$, caso exista um $u_i < 0$ a variável s_a é candiãata a entrar na base e o vetor $a_i = [o, e_i]$ atualizado, isto é, $B^{-1}a_i$, no caso a própria $(i+1)$ésima coluna de B^{-1}, já explícita no quadro do simplex. Faremos o pivoteamento com a coluna $i+1$ e s_i, entrará na base. Quando todos os u_i forem ≥ 0 **para i** $= 1, 2, \ldots, m$, então procuraremos o mínimo dos $ua_k - cx^k$, para $k = 1, 2, \ldots, p$. Para isso resolveremos o problema de programação linear:

$$\text{minimizar } ua_k - cx^k$$

sujeito a

$$x^k \varepsilon X$$

ou airada

$$\text{minimizar } u\,[1, Ax] - cx$$

sujeito a

$$x \varepsilon X$$

ou ainda fazendo $u = [u^o, u^1]$

$$\text{minimizar } u^o + (u^1 A - c)x$$

sujeito a (32)

$$x \in X$$

A solução de (32) é imediata, supondo $x=(x_1, x_2 .. x_n)^T$; $x_j=1$ se o coeficiente de x_j na função objetivo de (32) for negative e $x_j=0$ caso seu coeficiente na função objetivo seja positivo. Quando o coeficiente for nulo x_j pode ser igual a 0 ou igual a 1. Supondo que $\hat{x}$ seja o ótimo de (32), teremos também $\{u^o+(u^1A-c)\hat{x}\}$, caso $\{u^o+(u^a A-c)\hat{x}\} < 0$ então o vetor

$[u^o+(u^1A-c)\hat{x}, B^{-1}[1, A\hat{x}]]$ entrará na base, caso contrário, isto é, $u^o+ u^1A-c)\hat{x} \geq 0$, o quadro é ótimo.

<u>A Melhor Solução Viável</u>

Das soluções calculadas será escolhida a solução x^* que forneça o maior valor de cx^* (caso de maximizar cx) e que satisfaça a condição $Ax^* \leq b$.

Conforme anotado anteriormente a solução encontrada para o problema poderá não ser a Qtima porém poderá estar próxima dela.

Se nesta 1a. Fase a solução x^* ,e tal que

$$\sum_{j=1}^{n} a_{ij} x_j^* - b_i < a_{ij}^o \qquad i = 1,\ldots,m$$

onde $a_{ij}^o = \min a_{ij}$ para os j tais que, $x_j = 0$, o processo para e a solução x^* é considerada a melhor solução viável desde que

$$\sum_{j=1}^{n} a_{ij} x_j^* \leq b_i \qquad i = 1,\ldots,m$$

e $$cx^* = \max \{cx^k\}$$

8.2, Segunda Fase

Esta Fase foi introduzida a fim de melhorar a solução, isto é, a melhor solução viável. É utilizada caso existam nas restrições coeficientes a_{ij}, para as variáveis $x_j=0$, menores que as folgas obtidas na 1a. Fase com a melhor solução viável.

Se selecionam índices t das $x_j = 0$ e se contrõe um novo problema, auxiliar

$$\max c_t x_t$$

$$\sum_t a_{it} x_t \leq \bar{b}_i$$

$$x_t = 0 \text{ ou } 1$$

onde c_t são os coeficientes da função objetivo das variáveis $x_j = 0$

a_{it} são os coeficientes das variáveis nas restrições para os $x_j = 0$

$\bar{b}_i$ é a folga da restrição i i = 1,...,m) para a solução x*

Se eliminam do problema auxiliar as x_t tais que

$$a_{it} > \bar{b}_i$$

Se calculam funções H_t tais que

$$H_t = c_t / \sum_i^m (a_{it}/\bar{b}_i) \qquad t = 1,\dots,n_1$$

Se escolhe um H_r tal que

$$H_r = \max H_t$$

para o índice r escolhido se faz $x_r = 1$

9. Algoritmo

Inicialização

Para encontrar uma solução inicial se introduz no problema (30) uma variável artificial λ^a e se minimiza $\gamma = \lambda^a$ utilizando o método das Duas Fases, de programação linear,

1a. Fase

Seja $\hat{u}$ o vetor dual que corresponde à solução inicial $\hat{x}$

Faça $u = \hat{u}$ e $u = (u^o, u^1)$

Passo 1; Calcular

$$z(u) = \min_{x\varepsilon X}\{u^o + u^1(Ax-b)-cx\} = \min_{x\varepsilon X}\{u^o + (u^1A-c)x - u^1b\}$$

e faça $x_j = 1$ para $u^1A_j - c_j < 0$

$x_j = 1$ ou 0 para $u^1A_j - c_j = 0$

$x_j = 0$ para $u^aA_j - c_j > 0$

Passo 2. A solução x é viável?

2.1. Se x é viável .compara com a solução viável anterior e gùarda a solução que fornece o maior valor da função objetiva x^*. Va para 3

2.2. Caso contrário calcule $\beta_i = \sum_j a_{ij}x_j - b_i \ \forall i$

2.2.1. se $\beta_i \leq \xi_i$, $x_\xi = x$ e quarda x_5, vapara 3

(x_ξ = solução ξ - viável).

2.2.2. Se $\beta_i > \xi_i$ va para 3

Passo 3. Com a solução x calcular $\rho_j = (u1A_j - c_j)x_j + uo$

3.1. Se $\rho_j \geq 0 \quad \forall j$ va para 2a. Fase

3.2, Se $\rho_j < 0$ para ao menos um j calcule $\rho_K = \min \rho_j$

o vetor $\begin{pmatrix} \rho_K \\ 1 \\ Ax \end{pmatrix}$ entra na base. Atualizar o vetor que entra,mediante $B^{-1}\begin{pmatrix} \rho_K \\ 1 \\ Ax \end{pmatrix}$

Passo 4. Escolher o vetor que sae e calcular o novo B^{-1} e o novo u. Va para 1.

2a. Fase

Passo 5. Seja T = conjunto de índices j para $x_j^* = 0$ e $\bar{b}_i = b_i - \sum_j a_{ij} x_j^*$

se $b_i - \sum_j a_{ij} x_j^* < \min_{j \varepsilon T} a_{ij}$, $\forall$ i, PARE

Caso contrário continue

Passo 6. Construir um novo problema

$$\max \sum_{t \varepsilon T} c_t x_t$$

tal que

$$\sum_{t \varepsilon T} a_{it} x_t \leq \bar{b}_i$$

$$x_t = 0 \text{ ou } 1, \ t \varepsilon T$$

Passo 7. Seja T_a = conjunto de índices dos $(T_1 \subset T)$ x_t tais que:

$$a_{it} > \bar{b}_i \, , \ t \varepsilon T$$

Excluir do problema auxiliar as $\{x_t | t \varepsilon T_1\}$

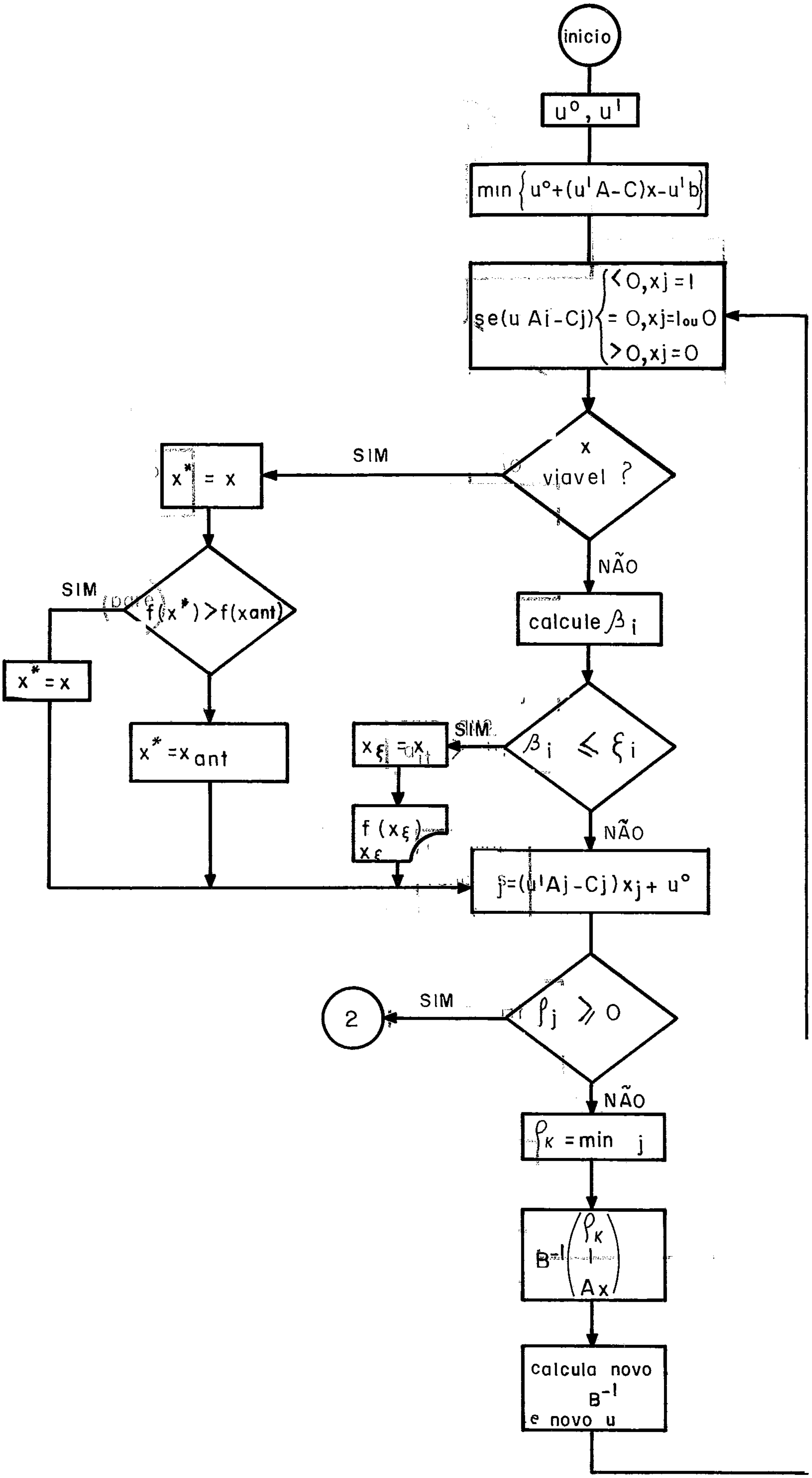
Passo 8. **Calcular as funções** H_t **tais que**

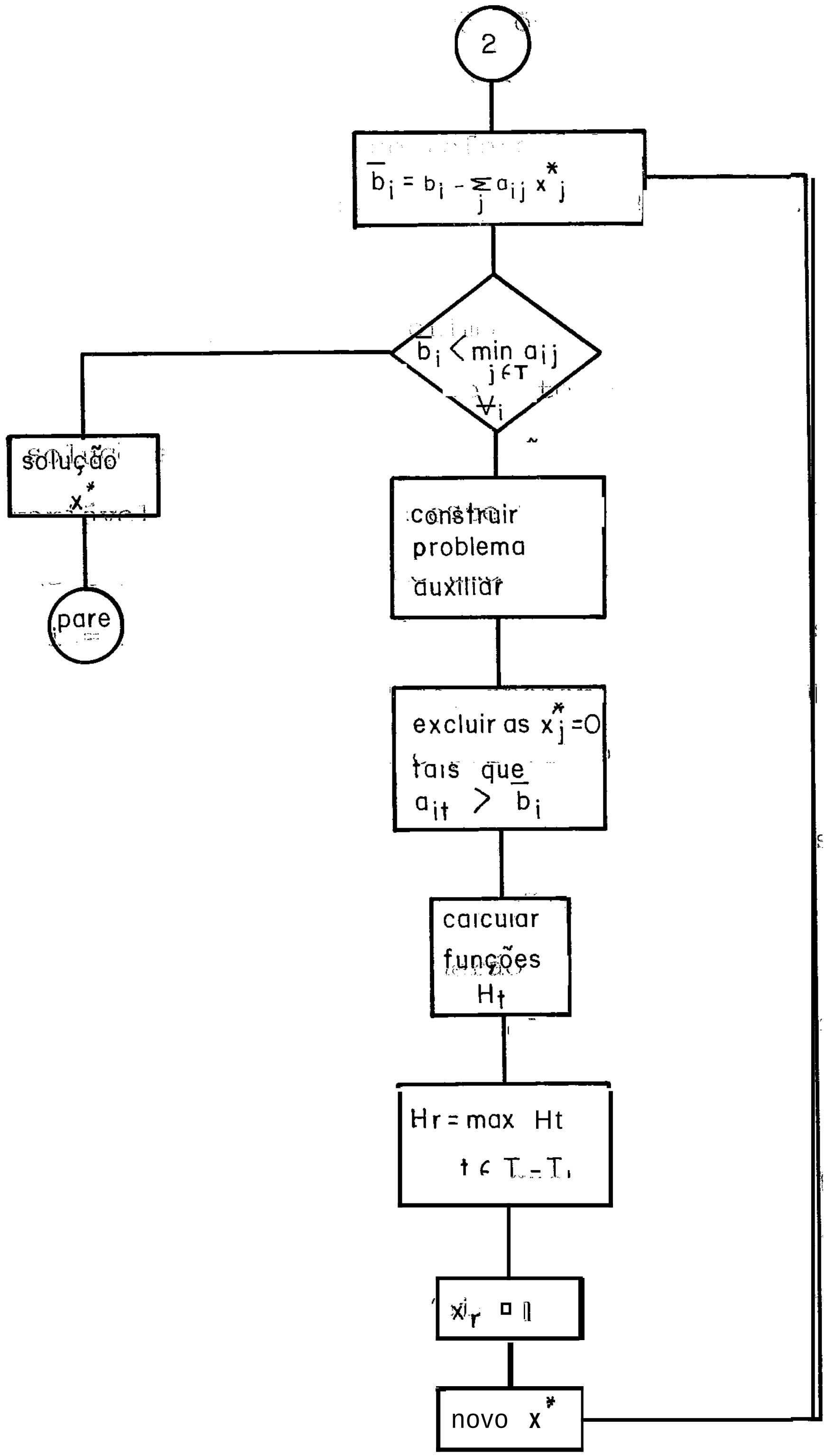
$$H_t = c_t/(\sum_i (a_{it}/\bar{b}_i)) \qquad t \varepsilon T - T_1$$

e faça $H_r = \max_{t \varepsilon T - T_1} H_t$

Passo 9. **Fazer** $x_r = 1$. **Se tem um novo** x_j^*. Va **para 5**

Fluxograma do Algoritmo Dual de composição




2
$\bar{b}_i = b_i - \sum_j a_{ij} x^*_j$
$\bar{b}_i < \min_{j \in T} a_{ij} \quad \forall i$
solução x^*
pare
construir problema auxiliar
excluir as $x^*_j = 0$ tais que $a_{it} > \bar{b}_i$
calcular funções H_t
$H_r = \max_{t \in T_0 - T_1} H_t$
$x_r = 1$
novo x^*

10. Conclusões

A procura de uma solução ótima, de soluções viáveis ou de soluções ξ- viáveis utilizando o método Dual de Decomposição possue uma limitação no que se refere à 1a. Fase, já que o método não garante que uma resposta seja encontrada. Na 1a. Fase poderá não ser encontrada uma solução ótima, uma solução "boa", isto é, uma solução viável próxima da ótima e também poderão não ser encontradas soluções ξ - viáveis. Se isto acontecer, a 2a. Fase encontrará soluções viáveis introduzindo em cada iteração (da 2a.Fase) uma variável $x_j = 1$, porém este procedimento seria mais demorado porque teria que partir de uma solução $x_j = 0$, $(j=1,\ldots,n)$ e $\bar{b}_i = b_i$, $(i= 1,\ldots,m)$, isto é, o problema original seria resolvido utilizando somente a 2a. Fase, apesar disto o tempo de computação seria menor que o utilizado por outros métodos.

Pode-se afirmar que para problemas que possuem grande número de variáveis e poucas restrições existirá uma coleção razoável de soluções viáveis que poderão ser obtidas com a 1a.Fase do método e dentre as quais pode ser selecionada a melhor para passar à 2a. Fase.

Também pode-se observar que a solução do problema dual se torna mais fácil que a do primal, fornece os limites superiores de z e as soluções ξ - viáveis que poderão estar próximas da ótima.

O procedimento é válido para ser utilizado como uma técnica de pós-otimização desde que se conheçam ou sejam estabelecidos os limites de tolerância para as soluções inviáveis.

Conforme será mostrado no próximo capitulo o tempo de computação para problemas de grande porte (0-1) é reduzido se comparado com outros métodos. Assim como também o espaço de memória utilizado no computador é reduzido, é necessário trabalhar com uma matriz (inversa) de dimensão $(m+2) \times (m+1)$, onde m representa o número de restrições. O número de iterações para encontrar as soluções também é pequeno se comparado com outros métodos.

As soluções encontradas na 1a. Fase são todas diferentes.

O algoritmo pode ser prsgramado em qualquer linguagem, de forma simples e eficiente como pode-se concluir do fluxograma apresentado anteriormente.

- Exemplo 2

Seja o problema primal

maximizar $3x_1+2x_2-3x_3$

sujeito a

$$x_1+x_2+x_3 \leq 2$$

$$2x_1+x_2-x_3 \leq 2$$

$$x_1 \varepsilon \{0,1\},\ x_2 \varepsilon \{0,1\},\ x_3 \varepsilon \{0,1\}$$

queremos calcular $z(u^*)$. Partiremos diretamente para calcular v^*,. solução de (30), pois sabemos que $z(u^*)=v^*$. Para resolvermos (30) partiremos da solução básica artificial, utilizaremos a seguir o método das duas fases.

γ	1	1	0	0
v	0	0	0	0
λ^a	1	1	0	0
s_1	2	0	1	0
s_2	2	0	0	1

λ^a é **uma variável artificial** minimizaremos $\gamma = \lambda_a$ **na primeira Fa-** se.

$$\max \ (1 \ 0 \ 0) \begin{pmatrix} 1 \\ x_1+x_2+x_3 \\ x_1+x_2+x_3 \end{pmatrix} - 0$$

sujeito a

$$0 \leqq x_1 \leqq 1, \quad 0 \leqq x_2 \leqq 1, \quad 0 \leqq x_3 \leqq 1$$

que fornece

$$\max \ 1$$

sujeito a

$0 \leqq x_1 \leqq 1$, $0 \leqq x_2 \leqq 1$, $0 \leqq x_2 \leqq 1$, **qualquer solução serve. pois a função objetivo independe de s, coloquemos** então $\bar{x}_1=0$, $\bar{x}_2=0$ **e** $\bar{x}_3=0$, **logo o vetor** que **entrará na base será da forma** $[1,0,1,0,0]$, **temos** então

					λ_1
γ	1	1	0	0	1
v	0	0	0	0	0
← λ^a	1	1	0	0	1
s_1	2	0	1	0	0
s_2	2	0	0	1	0

após pivoteamento **e abandonando a Linha de** γ, **temos, lembrando** que $x^1 = [0, 0, 0]$.

v	0	0	0	0
λ_1	1	1	0	0
s_1	2	0	1	0
s_2	2	0	0	1

que é um quadro inicial viável.

Calculemos

$$\underline{\min}\ (0\ 0\ 0)\begin{pmatrix} 1 \\ x_1+x_2+x_3 \\ 2x_1+x_2-x_3 \end{pmatrix} - (3\ 2\ -3)\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

sujeito a

$$0 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 1, \quad 0 \leq x_3 \leq 1$$

ou ainda

$\underline{\min}\ -3x_1-2x_2+3x_3$

sujeito a $0 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 1, \quad 0 \leq x_3 \leq 1$

A solução ótima será $\hat{x}_1 = \hat{x}_2 = 1$ e $\hat{x}_3 = 0$ e $-3\hat{x}_1-2\hat{x}_2+3\hat{x}_3 = -5 < 0$, logo o vetor que entrará na base será

$$B^{-1}\begin{pmatrix} -5 \\ 1 \\ 1+1+0 \\ 2+1-0 \end{pmatrix}$$

ou seja $[-5, 1, 2, 3]$, pois $B^{-1} = I$

					λ_2
v	0	0	0	0	-5
λ_1	1	1	0	0	1
s_1	2	0	1	0	2
s_2	2	0	0	1	3

com o mesmo procedimento anterior teremos

$$\min\ (0\ 0\ 5/3)\begin{pmatrix} 1 \\ x_1+x_2+x_3 \\ x_1+x_2+x_3 \\ 2x_1+x_2-x_3 \end{pmatrix} - (3\ 2\ -3)\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

sujeito a

$0\leq x_1\leq 1,\quad 0\leq x_2\leq 1\quad$ e $\quad 0\leq x_3\leq 1$, ou

$\min\ 1/3x_1-1/3x_2+4/3x_3$

sujeito a

$0\leq x_1\leq 1,\ 0\leq x_2\leq 1,\ 0\leq x_3\leq 1$, solução $\hat{x}_1=\hat{x}_3=0,\ \hat{x}_2=1$ e

$1/3\hat{x}_1-1/3\hat{x}_2+4/3\hat{x}_3=-1/3<0$, $\hat{x}=[0,1,0]$,

então $a_3=\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ e $B^{-1}a_3=\begin{pmatrix} 1 & 0 & -1/3 \\ 0 & 1 & -2/3 \\ 0 & 0 & 1/3 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}=\begin{pmatrix} 2/3 \\ 1/3 \\ 1/3 \end{pmatrix}$

o vetor que entrará na **base** será $[-1/3,\ 2/3,\ 1/3,\ 1/3]$

					λ_3
v	10/3	0	0	5/3	-1/3
λ_1	1/3	1	0	-1/3	2/3
s_2	2/3	0	1	-2/3	1/3
λ_2	2/3	0	0	1/3	1/3

após pivoteamento, lembrando que $x_3 = [0,1,0]$:

v	7/2	1/2	0	3/2	0
λ_3	1/2	3/2	0	-1/2	1
s_2	1/2	-1/2	1	-1/2	0
λ_2	1/2	-1/2	0	1/2	0

$$\min\ (1/2,0,3/2)\begin{pmatrix} 1 \\ x_1+x_2+x_3 \\ 2x_1+x_2-x_3 \end{pmatrix} - (3,2,-3)\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \text{s.a.} \begin{pmatrix} 0\leq x_1\leq 1 \\ 0\leq x_2\leq 1 \\ 0\leq x_3\leq 1 \end{pmatrix}$$

$$\min\ (1/2 + 3x_1 + 3/2\ x_2 - 5/2\ x_3) - (3x_1 + 2x_2 - 3x_3)$$

sujeito a $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 1$, $0 \leq x_3 \leq 1$

$$\min\ (1/2 + 0x_1 - 1/2\ x_2 + 3/2\ x_3)$$

sujeito a $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 1$, $0 \leq x_3 \leq 1$

Solução $x^4 = [0,\ 1,\ 0]$

A solução x^4 é a mesma que x^3

Calculamos

$$u\begin{pmatrix} 1 \\ Ax \end{pmatrix} - cx = 1/2 - 0x_1 - 1/2\ x_2 + 3/2\ x_3 = 0$$

portanto h* é a solução ótima para o problema (30)

$A^* = (0,\ 1/2,\ 1/2)$, $v = 7/2$

Solução do problema (8)

$x_1 = 0$, $x_2 = 1$, $x_3 = 0$, $cx = 2$

O quadro seguinte apresenta os resultados obtidos neste exemplo

k	x^k	cx^k	u_0^k	u_1^k	$2u_2^k$	v	solução
1	(0,0,0)	0	0	0	0	0	viável
2	(1,1,0)	5	0	0	5/3	10/3	inviável
3	(0,1,0)	2	1/2	0	3/2	7/2	viável
4	(0,1,0)						

Exemplo 3

Maximizar $x_1+x_2+2x_3+3x_4$

sujeito a $4x_1+7x_2+7x_3+3x_4 \leq 10$

$x_1 \varepsilon \{0,1\}$, $x_2 \varepsilon \{0,1\}$, $x_3 \varepsilon \{0,1\}$, $x_4 \varepsilon \{0,1\}$

Quadro inicial

γ	1	1	0
v	0	0	0
λ^a	1	1	0
λ_1	10	0	1

max 1, sujeito a $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 1$, $0 \leq x_3 \leq 1$, $0 \leq x_4 \leq 1$; $x^1 = (0,0,0,0)$

vetar, que entra na base (1,0,1,0)

				λ_1
γ	1	1	0	1
v	0	0	0	0
← λ^a	1	1	0	1
λ_1	10	0	1	0

v	0	0	0
λ_1	1	1	0
s_1	10	0	1

Calculamos $\min\limits_{x \varepsilon X} \quad u \begin{pmatrix} 1 \\ Ax \end{pmatrix} - cx$

$$\min \; (0 \;\; 0) \begin{pmatrix} 1 \\ \\ 4x_1+7x_2+7x_3+3x_4 \end{pmatrix} - (1 \;\; 1 \;\; 2 \;\; 3) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \quad , \; x \; \varepsilon \; X$$

$$\min \; \left[(0) \;\; (4 \;\; 7 \;\; 7 \;\; 3) - (1 \;\; 1 \;\; 2 \;\; 3 \;) \right] \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \quad , \; x \; \varepsilon \; X$$

$$\min\limits_{x \varepsilon X} \; (- x_1 - x_2 - 2x_3 - 3x_4)$$

Solução $x^2 = (1,1,1,1)$

Calculamos o vetor que entra

$$Ax^2 = (\; 4 \;\; 7 \;\; 7 \;\; 3) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = 21$$

vetor que entra $\begin{pmatrix} 1 \\ 21 \end{pmatrix}$

$(u^1 A - c) x^2 + u^o = - 7$

$$B^{-1}\begin{pmatrix}1\\21\end{pmatrix} = \begin{pmatrix}1\\21\end{pmatrix}$$

				λ_2
v	0	0	0	-7
λ_1	1	1	0	1
← s_1	10	0	1	21

Após o pivoteamento temos

v	10/3	0	1/3
λ_1	11/21	1	-1/21
λ_2	10/21	0	1/21

$$\min_{x \varepsilon X} \left((1/3)\ (4,\ 7,\ 7,\ 3) - (1,\ 1,\ 2,\ 3)\right)\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix}$$

$$\min_{x \varepsilon X} \left((4/3,\ 7/3,\ 7/3,\ 1) - (1,\ 1,\ 2,\ 3)\right)\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix}$$

$$\min_{x \varepsilon X} (1/3\ x_1 + 4/3\ x_2 + 1/3\ x_3 - 2x_4)$$

Solução $x^3 = (0,\ 0,\ 0,\ 1)$

Vetor que entra

$$Ax^3 = (\ 4\ \ 7\ \ 7\ \ 3)\begin{pmatrix}0\\0\\0\\1\end{pmatrix} = 3$$

$$\begin{pmatrix} 1 \\ Ax^3 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \rightarrow B^{-1} \begin{pmatrix} 1 \\ Ax^3 \end{pmatrix} = \begin{pmatrix} 1 & -1/21 \\ 0 & 1/21 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 6/7 \\ 1/7 \end{pmatrix}$$

$$(u^1A - c)x^3 + u^o = 1/3 \cdot 3 - 3 = -2$$

				λ_3
v	10/3	0	1/3	-2
$\leftarrow$ λ_1	11/21	1	-1/21	6/7
λ_2	10/21	0	1/21	1/7

após pivoteamento temos

v	41/9	7/3	2/9
λ_3	11/18	7/6	-1/18
λ_2	7/18	-1/6	1/18

$$\min_{x\varepsilon X} \left((2/9)\,(4,\ 7,\ 7,\ 3) - (1,\ 1,\ 2,\ 3) \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$\min_{x\varepsilon X} \left((8/9,\ 14/9,\ 14/9,\ 6/9) - (1,\ 1,\ 2,\ 3) \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$\min_{x\varepsilon X} (-1/9x_1 + 5/9x_2 - 4/9x_3 - 21/9x_4)$$

Solução $x^4 = (1,\ 0,\ 1,\ 1)$

Vetor **que entra**

$$Ax^4 = (4,\ 7,\ 7,\ 3) \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = 14$$

$$\begin{pmatrix} 1 \\ Ax^4 \end{pmatrix} = \begin{pmatrix} 1 \\ 14 \end{pmatrix} \to B^{-1}\begin{pmatrix} 1 \\ 14 \end{pmatrix} = \begin{pmatrix} 7/6 & -1/18 \\ -1/6 & 1/18 \end{pmatrix}\begin{pmatrix} 1 \\ 14 \end{pmatrix} = \begin{pmatrix} 7/18 \\ 11/18 \end{pmatrix}$$

$$(u^1 A - c)x^4 + u^o = -1/9 - 4/9 - 21/9 + 7/3 = -5/9$$

				λ_4
v	41/9	7/3	2/9	-5/9
λ_3	11/18	7/6	-1/18	7/18
λ_2	7/18	-1/6	1/18	11/18

após o pivoteamento temos

v	162/33	24/11	3/11
λ_3	4/11	14/11	-1/11
λ_4	7/11	-3/11	1/11

$$\min_{x\varepsilon X} \left((3/11)(4,\ 7,\ 7,\ 3) - (1,\ 1,\ 2,\ 3)\right)\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$\min_{x\varepsilon X} \left((12/11,\ 21/11,\ 21/11,\ 9/11) - (1,1,2,3)\right)\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$\min_{x\varepsilon X} (1/11x_1 + 10/11x_2 - 1/11x_3 - 24/11x_4)$$

Solução $x^5 = (0,\ 0,\ 1,\ 1)$

Vetor que entra

$$Ax^5 = (4,\ 7,\ 7,\ 3)\begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = 10$$

$$\begin{pmatrix} 1 \\ Ax^5 \end{pmatrix} = \begin{pmatrix} 1 \\ 10 \end{pmatrix} \rightarrow \quad B^{-1}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 14/11 & -1/11 \\ -3/11 & 1/11 \end{pmatrix}\begin{pmatrix} 1 \\ 10 \end{pmatrix} = \begin{pmatrix} 4/11 \\ 7/11 \end{pmatrix}$$

$$(u^1A - c)x^5 - u^o = -1/11 - 24/11 + 24/11 = -1/11$$

				λ_5
v	162/33	24/11	3/11	-1/11
← λ_3	4/11	14/11	-1/11	4/11
λ_4	7/11	-3/11	1/11	7/11

após o pivoteamento **ternos**

v	5	5/2	1/4
λ_5	1	7/2	-1/4
λ_4	0	-5/2	1/4

$$\min_{x \in X} \left| (1/4)\ (4,\ 7,\ 7,\ 3) - (1,\ 1,\ 2,\ 3) \right| \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$\min_{x \in X} \left| (1,\ 7/4,\ 7/4,\ 3/4) - (1,\ 1,\ 2,\ 3) \right| \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$\min_{x \in X} (0x_1 + 3/4x_2 - 1/4x_3 - 1/4x_4)$$

$x^6 = (0,\ 0,\ 1,\ 1)$ é **o** mesmo **que** x^5

O vetor **que entra portanto** será $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$$(u^1A - c)x^6 + u^o = 1/4 - 9/4 + 5/2 = -10/4 + 5/2 = 0$$

A solução h^* com $\lambda_5 = 1$, $\lambda_4 = 0$, $v = 5$ é ótima.

O quadro a seguir mostra os resultados obtidos neste problema em que é achada a solução ótima $cx^* = v$.

k	x^k	cx^k	uo^k	$u1^k$	v	Ax^k	solução
1	(0,0,0,0)	0	0	0	0	0	viável
2	(1,1,1,1)	7	0	1/3	10/3	21	inviável
3	(0,0,0,1)	3	7/3	2/9	41/9	3	viável
4	(1,0,1,1)	6	24/11	3/11	162/33	14	inviável
5	(0,0,1,1)	5	5/2	1/4	5	10	viável
6	(0,0,1,1)	5					

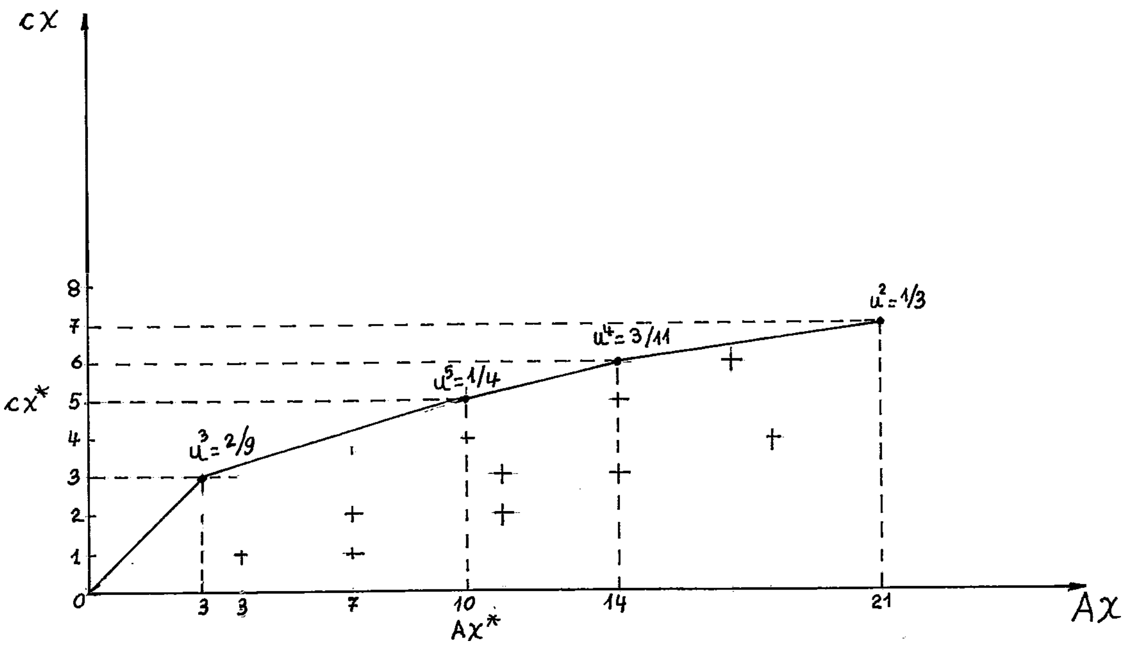


Figura 2

A figura 2 mostra as soluções encontradas (.) e os respectivos u. As soluções (+) representam as soluções possíveis não calculadas. O valor de u* fornece a solução ótima do problema para o valor cx* da função objetivo e para o valor b* da restrição. Neste caso a solução ótima encontrada é uma solução totalmente inteira para o problema P da seção 4.

Exemplo 4

Resolver o problema do exemplo 1

Na primeira fase são obtidas as soluções possíveis seguintes:

k	x^k	cx^k	u_0^k	u_1^k	Ax^k	solução
1	(0,0,0,0,0,0,0,0,0)	0	0	0	0	viável
2	(1,1,1,1,1,1,1,1,1)	45	0	0.6923	65	inviável
3	(0,0,0,0,0,1,1,1,1)	30	0	0.8571	35	inviável
4	(0,0,0,0,0,0,0,0,1)	7	3.1666	0.7666	5	viável
5	(0,0,0,0,0,1,0,0,1)	13	4.1304	0.7391	12	viável
6	(0,0,0,0,0,1,0,1,1)	22	4.5454	0.7272	24	viável

Portanto a melhor solução viável será a 6a.

2a. Fase

$T = \{1, 2, 3, 4, 5, 7\}$

$$\bar{b}_i = -\sum_{j=1}^{n} a_{ij} x_j^* + b_i = 30 - 24 = 6$$

Se conãtroi o problema auxiliar

$$\max_{t \varepsilon T} x_1 + 4x_2 + 2x_3 + 3x_4 + 5x_5 + 8x_7$$

sujeito a

$$2x_1 + 6x_2 + 5x_3 + 8x_4 + 9x_5 + 11x_7 \leq 6$$

$$x_t = 0 \text{ ou } 1 \qquad t \varepsilon T$$

$$T_1 = \{4, 5, 7\}$$

Excluindo do problema auxiliar as xt para $t \varepsilon T_1$ temos

$$\max x_1 + 4x_2 + 2x_3$$

sujeito a

$$2x_1 + 6x_2 + 5x_3 \leq 6$$

$$x_1 \varepsilon \{0,1\},\ x_2 \varepsilon \{0,1\},\ x_3 \varepsilon \{0,1\}$$

Calcula-se H_t para $t \varepsilon T - T_1$

Calcular $\underset{i}{C}\ (a_{it}/\bar{b}_i)$ para $t = 1,2,3$; $\{2/6,\ 6/6,\ 5/61\ \doteq \{0.33, 1, 0.83\}$

$$H_r = \max_{t=1,2,3} H_t = \max \{1/0.33, 4/1, 2/0.83\} = \max \{3, 4, 2.4\}$$

$$H_r = 4, \quad r = 2$$

$$x_2 = 1$$

Nova solução $(0,1,0,0,0,1,0,1,1)$

Novo $T = \{1,3\}$

$$\bar{b}_i = 30 - 30 = 0$$

Então a solução será $x^* = (0,1,0,0,0,1,0,1,1)$, $z = 26$, folga $= 0$

CAPÍTULO IV

1. Introdução

Neste capftulo serão apresentados um conjunto de problemas e suas soluções, obtidas pelo Método Dual de Decomposição.

Alguns dos problemas aqui resolvidos foram extraídos das referências indicadas em cada exemplo. A seleção destes problemas,foi feita em face às dificuldades que apresentam para serem resolvidos mediante outros métodos e também por possuírem as soluções obtidas por esses métodos,o que nos permite comparac os resultados e os tempos utilizados para obter as soluções. Os dados dos problemas sem referência foram gerados arbitrariamente e na sua maioria servem para mostrar a eficiência do método à medida que as variáveis aumentam. Inicialmente são mostrados resultados completos de problemas de programação zero - um com uma restrição e posteriormente problemas com mais de uma restrição.

As soluções apresentadas foram obtidas mediante um programa em linguagem Fortran, desenvolvido para o algoritmo , utilizando um computador Burroughs 6700.

Os problemas resolvidos podem ser representados da seguinte forma:

$$\text{Maximizar } z = \sum_{j=1}^{n} c_j x_j$$

$$\text{sujeita a} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad i = 1,\ldots,m$$

$$x_j = 0 \text{ ou } 1 \qquad j = 1,\ldots,n$$

2. Resultados e Análise dos Resultados

Problema 1. m = 1, n = 9

j	1	2	3	4	5	6	7	8	9	b_i
c_j	1	4	2	3	5	6	8	9	7	
a_{ij}	2	6	5	8	9	7	11	12	5	30

As soluções apresentadas no Quadro 1 representam todas as soluções geradas pelo Método de Decomposição na 1a. e 2a. Fases.

Quadro 1

k	x^k	$a_{ij}x_j^k$	$c_jx_j^k$	u^{0k}		estado da solução
1a. Fase						
1	(0,0,0,0,0,0,0,0,0)	0	0	0	0	viável
2	(1,1,1,1,1,1,1,1,1)	65	45	0	0.6923	inviável
3	(0,0,0,0,0,1,1,1,1)	35	30	0	0.8571	inviável
4	(0,0,0,0,0,0,0,0,1)	5	7	3.1666	0.7666	viável
5	(0,0,0,0,0,1,0,0,1)	12	13	4.1304	0.7391	viável
6	(0,0,0,0,0,1,0,1,1)	24	22	4.5454	0.7272	viável
2a. Fase						
7	(0,1,0,0,0,1,0,1,1)	30	26			viável

A figura 1 mostra os valores de $(a_{ij}x_j^k)$ v.s. $(c_jx_j^k)$ assim como os valores de u^{1k}

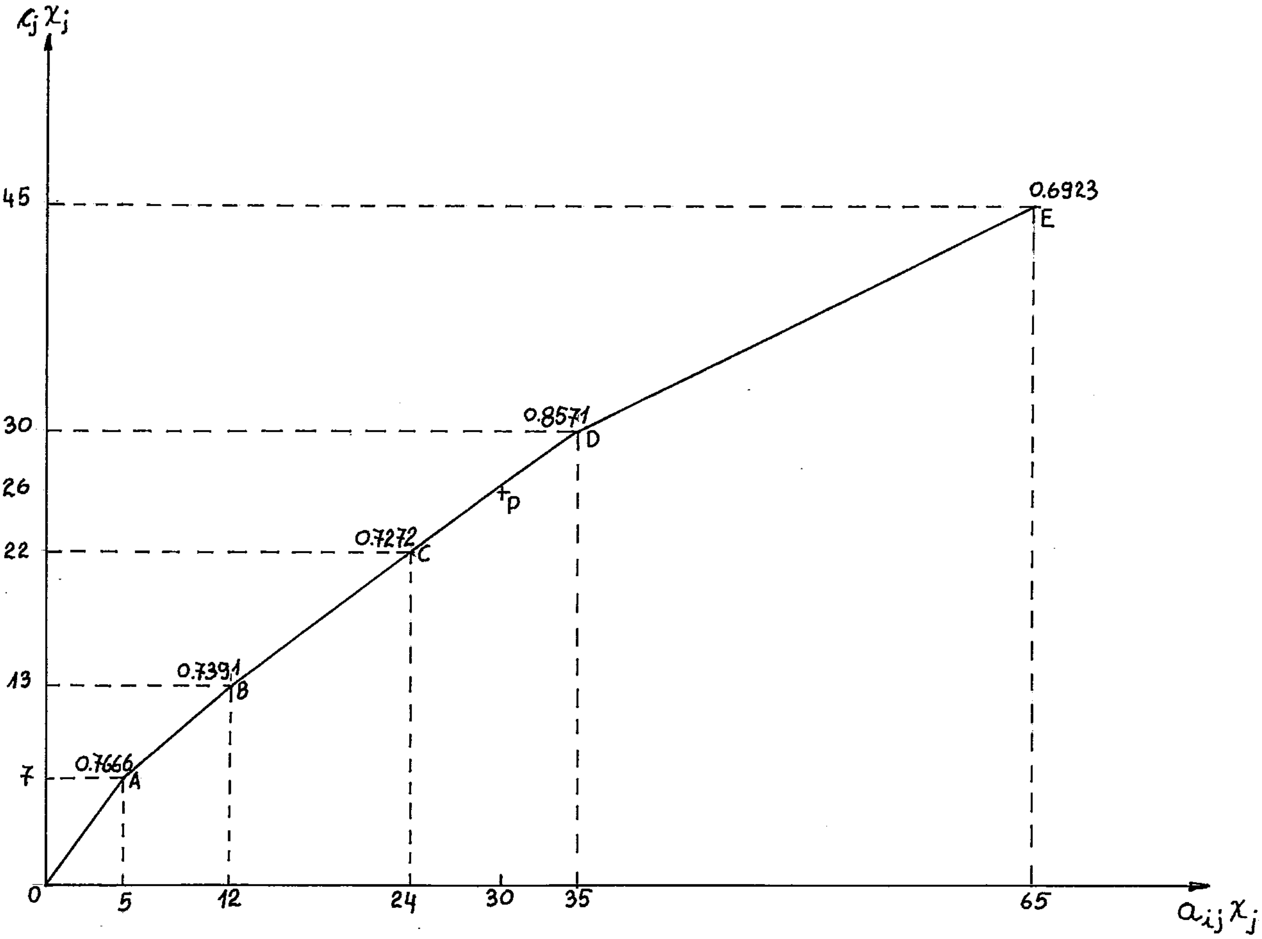


FIGURA 1

Todas as soluções viáveis estarão sob a figura OABCDE e a solução ótima estará entre a solução viável de maior valor (ponto C) e a solução inviável de menor valor (ponto D). Então é possível melhorar ainda mais a solução viável do ponto C utilizando o que foi denominada a Fase 2 do método e se achou o ponto P como melhor solução.

Problema 2. $m = 1$, $n = 40$

Os testes feitos com o problema 2 se referem a uma variação no segundo membro, os dados utilizados constam no Quadro 2.1 e os resultados nos Quadros 2.2.1, 2.2.2 e 2.2.3 mostram as melhores soluções viável e inviável na primeira Fase, a solução após a 2a. Fase, assim como também o tempo gasto pelo computador para obter estas soluções.

Quadro 2.1

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
c_j	6	3	5	6	3	9	5	6	2	1	5	6	8	4
a_{ij}	9	7	5	6	4	9	8	6	5	3	4	2	9	6

j	15	16	17	18	19	20	21	22	23	24	25	26	27	28
c_j	2	8	3	2	5	9	9	6	2	5	4	7	5	3
a_{ij}	7	5	3	4	2	1	9	1	2	5	7	8	9	6

j	29	30	31	32	33	34	35	36	37	38	39	40
c_j	9	2	1	3	6	5	8	9	3	2	5	8
a_{ij}	4	3	4	4	2	1	7	3	2	9	8	5

Quadro 2.2.1

		1a. Fase		
		Solução viável de maior valor		
Problema	b	z	$x_j = 1$, zero as outras	Folga
2.1	50	87	11,12,16,19,20,22,29,33,34,35, 36,37,40	11
2.2	60	87	idem 2.1	21
2.3	70	87	idem 2.1	31
2.4	80	87	11,12,16,19,20,22,29,33,34,35, 36,37,40	41
2.5	90	132	3,4,6,8,11,12,16,17,19,20,21,22, 23,24,29,33,34,35,36,37,40	6.
2.6	100	140	3,4,6,8,11,12,13,16,17,19,20,21, 22,23,24,29,33,34,35,36,37,40	7
2.7	110	153	3,4,5,6,8,11,12,13,16,17,19,20,21, 22,23,24,26,29,32,33,34,35,36,37,40	1
2.8	120	153	idem 2.7	1
2.9	130	165	1,3,4,5,6,8,11,12,13,14,16,17,19,20, 21,22,23,24,26,29,30,32,33,34,35,36, 37,40	3
2.10	140	165	idem 2.9	13.
2.11	150	179	1,3,4,5,6,7,8,11,12,13,14,16,17,19, 20,21,22,23,24,25,26,29,30,32,33,34, 35,36,37,39,40	0

Quadro 2.2.2

	1a. Fase		
	Solução inviável de menor valor		
Problema	z	$x_j = 1$, zero as outras	valor da inviabilidade
2.1	132	3,4,6,8,11,12,16,17,19,20,21,22, 23,24,29,33,34,35,36,37,40	34
2.2	132	idem 2.1	24
2.3	132	idem 2.1	14
2.4	132	idem 2.1	4.
2.5	140	3,4,6,8,11,12,13,16,17,19,20,21,22, 23,24,29,33,34,35,36,37,40	3.
2.6	147	3,4,6,8,11,12,13,16,17,19,20,21,22, 23,24,26,29,33,34,35,36,37,40	1.
2.7	165	1,3,4,5,6,8,11,12,13,14,16,17,19,20,21, 22,23,24,26,29,30,32,33,34,35,36,37,40	17.
2.8	165	idem 2.7	7.
2.9	175	1,3,4,5,6,7,8,11,12,13,14,16,17,19,20,21, 22,23,24,26,29,30,32,33,34,35,36,37,39,40	13.
2.10	175	idem 2.9	3.
2.11	200	1,2,.......,40	54.

Quadro 2.2.3

Problema	2a. Fase Solução viável - quase ótima z	 $x_j = 1$	 Folga	Tempo total de processamento(seg.) B.6700
2.1	98	3,4,11,12,16,19,20,22,29,33, 34,35,36,37,40	0.	6.48
2.2	107	3,4,6,11,12,16,19,20,22,29, 33,34,35,36,37,40	1.	6.39
2.3	118	3,4,6,8,11,12,16,17,19,20,22, 23,29,33,34,35,36,37,40	0.	8.07
2.4	127	3,4,6,8,11,12,16,17,19,20,21, 22,23,29,33,34,35,36,37,40	1.	7.10
2.5	135	3,4,5,6,8,11,12,16,17,19,20,21, 22,23,24,29,33,34,35,36,37,40	2.	6.27
2.6	145	3,4,5,6,8,11,12,13,16,17,19,20, 21,22,23,24,29,30,33,34,35,36, 37,40	0.	6.05
2.7	153	3,4,5,6,8,11,12,13,16,17,19 20, 21,22,23,24,26,29,32,33,34,35, 36,37,40	1	6.50
2.8	159	3,4,5,6,8,11,12,13,14,16,17,19, 20,21,22,23,24,26,29,30,32,33, 34,35,36,37,40	2.	6.84
2.9	166	1,3,4,5,6,8,10,11,12,13,14,16, 17,19,20,21,22,23,24,26,29,30, 32,33,34,35,36,37,40	0.	6.30
2.10	172	1,3,4,5,6,7,8,11,12,13,14,16,17 18,19,20,21,22,23,24,26,29,30, 32,33,34,35 36,37,40	1.	6.16
2.11	179	idem la. Fase	0.	7.08

Dos quatro 2.2.1 e 2.2.3 pode-se observar que em alguns casos não seria necessária a utilização da 2a. Fase do método. Assim temos o problema 2.7 cujo valor da função objetivo obtido na primeira Fase é o mesmo que aquele obtido no fim da 2a. Fase, dado que o valor da folga é menor que os coeficientes das variáveis iguais a zero na solução da la. Fase. Também no problema 2.11 o valor da função objetivo obtido na la. Fase é igual ao da 2a. Fase dado que o valor da folga é nulo. Assim a 2a. Fase será um complemento da la. Fase caso existam coeficientes a_{ij} das variáveis nulas menores que a folga encontrada na la. Fase.

Também se pode observar que soluções inviáveis encontradas poderão ser aceitas como "boas" soluções desde que as inviabilidades não ultrapassem uma determinada tolerância, por exemplo 5%, nesse caso estão os problemas que constam no Quadro 3.

A percentagem da inviabilidade é definida como

$$\% = 100 \left(\sum_{j=1}^{n} a_{ij} \hat{x}_j - b_i \right) / b_i \quad (i = 1, \ldots, m)$$

onde $\hat{x}_j$ $(j = 1, \ldots, n)$ são os valores que as variáveis assumem na solução inviável.

Quadro 3

Problema	Valor da inviabilidade	Valor da função objetivo da solução inviável	Valor da função objetivo após a 2a. Fase	% de b na inviabilidade
2.4	4	132	127	5.0
2.5	3	140	135	3.3
2.6	1	147	145	1.0
2.10	3	175	172	2.1

Analisando os problemas 2.7 e 2.8 temos que após a 1a. Fase a solução (x_j), o valor da função objetivo ($c_j x_j$) da solução-viável e da solução inviável são os mesmos. Porém as soluções após a 2a. Fase são diferentes, o mesmo acontece com os problemas 2.9 e 2.10. Assim se for utilizada somente a 1a. Fase as soluções seriam iguais para os dois pares de problemas e neste caso se justifica a utilização da 2a. Fase,

Problema 3. ver Kaplan [68]

Este exemplo apresenta o problema Lorie-Savage para destinação de recursos em diversos projetos.

O problema foi resolvido para m = 2, utilizando inicialmente as 9 primeiras variáveis [seleção entre 9 projetos) e posteriormente acrescentando uma 10a. variável (mais um projeto).

j	1	2	3	4	5	6	7	8	9	10	b
c_j	14	17	17	15	40	12	14	10	12	15	
a_{1j}	12	54	6	6	30	6	48	36	18	6	50
a_{2j}	3	7	6	2	35	6	4	3	3	7	20

Para o problema com 9 Projetos foram encontrados os seguintes resultados

$$\sum_{j=1}^{9} c_j x_j = 70, \quad \sum_{j=1}^{9} a_{1j} x_j = 48, \qquad \text{folga} = 2$$

$$\sum_{j=1}^{9} a_{2j} x_j = 20, \qquad \text{folga} = 0$$

$$x_1 = x_3 = x_4 = x_6 = x_9 = 1, \qquad x_2 = x_5 = x_7 = x_8 = 0$$

Esta solução já foi encontrada na 1a. Fase.

Dentre as soluções inviáveis encontradas, nenhuma me receceu ser escolhida como uma alternativa, dado que as inviabilidades são muito elevadas em relação aos segundos membros das restrições.

Deve-se salientar que esta solução é a solução Ótima obtida por Kaplan [68] utilizando outro método.

Para o problema com 10 projetos a solução encontrada após a 2a. Fase é a mesma anterior, porém dentre as soluções inviáveis foi obtida uma com os valores seguintes:

$$\sum_{j=1}^{10} c_j x_j = 73, \quad \sum_{j=1}^{10} a_{1j} x_j = 48, \quad \text{folga} = 2$$

$$\sum_{j=1}^{10} a_{2j} x_j = 21, \quad \text{inviabilidade} = 1$$

isto é, com o acréscimo de uma unidade na 2a. restrição consegue-se um incremento de 3 unidades na função objetivo e os projetos selecionados correspondem às variáveis $x_1=x_3=x_4=x_9=x_{10}=1$, e os não incluídos serão $x_2=x_5=x_6=x_7=x_8 = 0$. Em relação à solução anterior, saiu o projeto número 6 e foi incluído o projeto número 10.

<u>Problema 4.</u> m = 2, n = 20

até agora só foram mostrados os testes com problemas com uma restrição e com duas restrições e poucas variáveis. Este problema aqui apresentado tem por objetivo testar o método com maior número de variáveis (20) e duas restrições.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	b_i
c_j	6	3	5	6	3	9	5	6	2	1	5	6	8	4	2	8	3	2	5	9	
a_{1j}	9	7	5	6	4	9	8	6	5	3	4	2	9	6	7	5	3	4	2	1	80
a_{2j}	9	1	2	5	7	8	9	6	4	3	4	4	2	1	7	3	2	9	8	5	60

Solução obtida:

$x_1=x_2=x_3=x_4=x_6=x_8=x_{11}=x_{12}=x_{13}=x_{14}=x_{16}=x_{17}=x_{19}=x_{20} = 1;$

$x_5=x_7=x_9=x_{10}=x_{15}=x_{18} = 0$

Valor da função objetivo = **83**

Folga da 1a. restrição = 6

Folga da 2a. restrição = 0

Tempo de computação = 6.26 segundos

Para este problema todas as soluções inviáveis ultrapassam o limite de 10% de inviabilidade e portanto não foram consideradas para serem analisadas.

Problemas 5, 6, 7, 8, 9 e 10

A seguir serão apresentados os resultados de 5 problemas extraídos do artigo de C.C. Petersen [83] que serão comparados com os resultados, por ele obtidos. Segundo Petersen, o tempo de computador gasto para resolver problemas de alocação de capital com mais de 50 variáveis é de 36 minutos ou menos utilizando um computador SDS 930.

O Quadro 4 mostra o tempo gasto pelo computador e o número de iterações necessárias, obtidos por Petersen,para encontrar as soluções ótimas dos problemas,

Quadro 4

Problema nº	6	7	8	9	10
número de restrições	10	10	10	5	5
número de variáveis	15	20	28	39	50
tempo(min) SDS 930	1	4	3	5	36
nº de iterações	169	1805	1041	2628	16778

O quadro 5 apresenta os resultados para os mesmos problemas utilizando o método Dual de Decomposição.

Quadro 5

Problema nº		5	6	7	8	9	10
nº de restrições		10	10	10	10	5	5
nº de variáveis		10	15	20	28	39	50
tempo(seg.) B 6700		6.03	7.72	7.36	8.07	7.99	7.21
nº de iterações	1a. fase	7	17	11	19	29	30
	2a. fase	2	0	0	1	0	2

Problema 5. $m = 10$, $n = 10$

j	c_j	a_{1j}	a_{2j}	a_{3j}	a_{4j}	a_{5j}	a_{6j}	a_{7j}	a_{8j}	a_{9j}	a_{10j}
1	600.0	20	20	60	60	60	60	5	45	55	65
2	310.5	5	7	3	8	13	13	2	14	14	14
3	1800.0	100	130	50	70	70	70	20	80	80	80
4	3850.0	200	280	100	200	250	280	100	180	200	220
5	18.6	2	2	4	4	4	4	2	6	6	6
6	198.7	4	8	2	6	10	10	5	10	10	10
7	882.0	60	110	20	40	60	70	10	40	50	50
8	4200.0	150	210	40	60	90	105	60	100	140	180
9	402.5	80	100	6	16	20	22	0	20	30	30
10	327.0	40	40	12	20	24	28	0	0	40	50
b_i =		450	540	200	360	440	480	200	360	440	480

Solução ótima obtida por Petersen

$z = 8706.1$

$x_j = 1$ para $j = 2,4,5,8,10$

$x_j = 0$ para $j = 1,3,6,7,9$

A melhor solução viável obtida com o método Dual de Decomposição é a seguinte:

$z = 8577.8$

$x_j = 1$ para $j = 2,4,5,6,8$

$x_j = 0$ para $j = 1,3,7,9,10$

Porém foi obtida a solução inviável (ξ- viável) seguinte:

$z = 9159.30$

$x_j = 1$ para $j = 1,2,4,6,8$

$x_j = 0$ para $j = 3,5,7,9,10$

As restrições que tornam inviável a solução são:

$i = 3$, valor da inviabilidade = 3, % (b_3) = 1.5%

$i = 10$, valor da inviabilidade = 9, % (b_{10}) = 2.0%

Em relação ao valor da solução Ótima obtida por Petersen a função objetivo tem um acréscimo de 453 unidades, o que significa um incremento de 5.2%.

Problema 6. m = 10, n = 15

Os problemas 6, 7 e 8 utilizaram a matriz de dados seguinte

j	c_j	a_{1j}	a_{2j}	a_{3j}	a_{4j}	a_{5j}	a_{6j}	a_{7j}	a_{8j}	a_{9j}	a_{10j}
1	100	8	8	3	5	5	5	0	3	3	3
2	220	24	44	6	9	11	11	0	4	6	8
3	90	13	13	4	6	7	7	1	5	9	9
4	400	80	100	20	40	50	55	10	20	30	35
5	300	70	100	20	30	40	40	4	14	29	29
6	400	80	90	30	40	40	40	10	20	20	20
7	205	45	75	8	16	19	21	0	6	12	16
8	120	15	25	3	5	7	9	6	12	12	15
9	160	28	28	12	18	18	18	0	10	10	10
10	580	90	120	14	24	29	29	6	18	30	30
11	400	130	130	40	60	70	70	32	42	42	42
12	140	32	32	6	16	21	21	3	9	18	20
13	100	20	40	3	11	17	17	0	12	18	18
14	1300	120	160	20	30	30	35	70	100	110	120
15	650	40	40	5	25	25	25	10	20	20	20
16	320	30	60	0	10	15	20	0	5	15	20
17	480	20	55	5	13	25	25	0	6	18	22
18	80	6	10	3	5	5	5	0	4	7	7
19	60	3	6	0	1	1	2	0	1	2	3
20	2550	180	240	20	80	100	110	0	20	40	50
21	3100	220	290	30	60	70	70	30	50	60	60
22	1100	50	80	40	50	55	55	10	30	50	55
23	950	30	90	10	20	20	20	0	5	25	25
24	450	50	70	0	30	50	50	10	20	25	30
25	300	12	27	5	10	15	20	10	20	25	25
26	220	5	17	0	5	15	15	5	10	15	15
27	200	8	8	0	3	6	6	0	10	10	10
28	520	18	28	10	20	20	20	10	20	28	28

sendo que o problema 6 utiliza as 15 primeiras variáveis, 10 restrições e seu segundo membro é

i	1	2	3	4	5	6	7	8	9	10
b_i	550	700	130	240	280	310	110	205	260	275

Solução Ótima obtida por Petersen

$z = 4015$

$x_j = 1$ para $j = 1,2,4,6,7,9,10,14,15$

$x_j = 0$ para $j = 3,5,8,11,12,13$

A melhor solução viável obtida com o método Dual de Decomposição é a mesma que a ótima obtida por Petersen, porem não foi achada nenhuma solução ξ - viável que satisfaça a csndição de ser menor ou igual a 5% dos b_i.

Problema 7, $m = 10$, $n = 20$

Utiliza a mesma matriz e o mesmo segundo membro anteriores; porém com as 20 primeiras variáveis.

Solução ótima de Petersen

$z = 6120$

$x_j = 1$ para $j = 1,10,14,15,16,17,18,19,20$

$x_j = 0$ para $j = 2,3,4,5,6,7,8,9,11,12,13$

Melhor solução viável obtida pelo método Dual de Decomposição

$a = 6010$

$x_j = 1$ para $j = 1,2,3,9,14,15,16,17,18,19,20$

$x_j = 0$ para $j = 4,5,6,7,8,10,11,12,13$

Também foi obtida uma solução ξ - viável com

$z = 6500$

$x_j = 1$ para $j = 1,2,9,10,14,15,16,17,18,19,20$

$x_j = 0$ para $j = 3,4,5,6,7,8,11,12,13$

As inviabilidades estão localizadas nas restrições

$i = 2$, com 71, $\%\ (b_2) = 10.1$

$i = 9$, com 1, $\%\ (b_9) = 0.38$

$i = 10$, com 18, $\%\ (b_{10}) = 6.5$

Problema 8. $m = 10$, $n = 28$

O segundo membro é, como segue:

i	1	2	3	4	5	6	7	8	9	10
b_i	930	1210	272	462	532	572	240	400	470	490

Solução ótima de Petersen

$z = 12400$

$x_j = 1$ para $j = 1,2,3,9,14,15,16,17,18,19,20,21,22,$
$23,25,26,27,28$

$x_j = 0$ para $j = 4,5,6,7,8,10,11,12,13,24$

Pelo método Dual de Decomposição melhor solução viável

$z = 12370$

$x_j = 1$ para $j = 1,2,3,14,15,17,18,19,20,21,22,23,24,$
$25,26,27,28$

$x_j = 0$ para $j = 4,5,6,7,8,9,10,11,12,13,16$

Também foram obtidas as soluções ξ - viáveis que são mostradas no Quadro 6.

Quadro 6

	z	x_j = 1. Zero as outras	i das restrições inviáveis	inviabilidade % b_i
1a.	12890	j = 1,3,8,9,12,14,15,16,17,18, 19,20,21,22,23,24,25,26,27, 28	2 9 10	6.3 6.8 11.6
2a.	12850	j = 1,2,3,9,14,15,16,17,18,19, 20,21,22,23,24,25,26,27,28	2 9 10	5.2 1.7 6.1
3a.	13340	j = 1,2,9,10,14,15,16,17,18,19, 20,21,22,23,24,25,26,27,28	1 2 9 10	1.3 14.1 6.1 10.4
4a.	12440	j = 1,2,9,14,15,17,18,19,20,21, 22,23,24,25,26,27,28	10	0.2
5a.	13020	j = 1,2,9,10,14,15,17,18,19,20, 21,22,23,24,25,26,27,28	2 9 10	9.1 2.9 6.3

As cinco soluções ξ - viáveis sao superiores ao valor da função objetivo, porém pode ser considerada como melhor (das inviáveis) a 4a. solução que para um valor da função objetivo igual a 12440, sua inviabilidade atinge um valor desprecível que é de 0.2%. Podemos considerar as soluções 2a. e 5a. como "boas" dado que suas inviabilidades atingem valores menores que 10%, especialmente a 2a. solução cuja maior inviabilidade, na 1a. restrição, atinge 6.1%.

Problema 9. m = 5, n = 39

j	c_j	a_{1j}	a_{2j}	a_{3j}	a_{4j}	a_{5j}
1	560	40	16	38	8	38
2	1125	91	92	39	71	52
3	300	10	41	32	30	30
4	620	30	16	71	60	42
5	2100	100	150	80	200	170
6	431	20	23	26	18	9
7	68	3	4	5	6	7
8	328	12	18	40	30	20
9	47	3	6	8	4	0
10	122	18	0	12	8	3
11	322	9	12	30	31	21
12	196	25	8	15	6	4
13	41	1	2	0	3	1
14	25	1	1	1	0	2
15	425	10	0	23	18	14
16	4260	280	200	100	60	310
17	416	10	20	0	21	8
18	115	8	6	20	4	4
19	82	1	2	3	0	6
20	22	1	1	0	2	1
21	631	49	70	40	32	18
22	132	8	9	6	15	15
23	420	21	22	8	31	38
24	86	6	4	0	2	10
25	42	1	1	6	2	4
26	103	5	5	4	7	8
27	215	10	10	22	8	6
28	81	8	6	4	2	0
29	91	2	4	6	8	0
30	26	1	0	1	0	3
31	49	0	4	5	2	0
32	420	10	12	14	8	10
33	316	42	8	8	6	6
34	72	6	4	2	7	1
35	71	4	3	8	1	3
36	49	8	0	0	0	0
37	108	0	10	20	0	3
38	116	10	0	0	20	5
39	90	1	6	0	8	4
40	738	40	28	6	14	0
41	1811	86	93	12	20	30
42	430	11	9	6	2	12
43	3060	120	30	80	40	16
44	215	8	22	13	6	18
45	58	3	0	6	1	3
46	296	32	36	22	14	16
47	620	28	45	14	20	22
48	418	13	13	0	12	30
49	47	2	2	1	0	4
50	81	4	2	2	1	0

O problema 9 está definido pelas 39 primeiras variáveis, 5 restrições e o seguinte segundo membro

i	1	2	3	4	5
b_i	600	500	500	500	600

Solução ótima de Petersen

$z = 10618$

$x_j = 1$ para $j = 1,2,4,6,8,9,11,13,15,16,17,18,19,20,$
$23,25,27,28,29,31,32,34,35,36,37,38,$
39

as outras $x_j = 0$

Melhor solução viável pelo método Dual de Decomposição

$z = 10278$

$x_j = 1$ para $j = 1,4,6,8,9,11,13,14,15,16,17,18,19,$
$20,21,23,25,26,27,28,29,30,31,32,$
$34,35,36,37,38,39$

As soluções ξ - viáveis obtidas constam no Quadro 7.

Quadro 7

	z	x_j = 1, zero as outras	i das restrições inviáveis	inviabilidade % b_i
1a.	10979	j = 2,3,4,6,8,9,11,13,14,15,16, 17,19,20,21,23,25,26,27,28, 29,30,31,32,34,35,37,38,39	1 2 3 5	1.1 18.2 2.6 3.1
2a.	10696	j = 1,2,4,6,8,11,13,14,15,16,17, 19,20,23,24,25,26,27,28,29, 30,31,32,34,35,36,37,38,39	5	3.1
3a.	11002	j = 2,4,6,8,9,11,13,14,15,16,17, 19,20,21,23,26,27,28,29,30, 31,32,33,34,35,36,37,38,39	1 2	7.6 11.4
4a.	10608	j = 1,2,4,6,8,9,11,13,14,15,16, 17,19,20,23,25,26,27,28,29, 30,31,32,34,35,37,38,39	5	1.5
5a.	10772	j = 1,2,4,6,8,9,11,13,14,15,16, 17,18,19,20,23,25,26,26,28, 29,30,31,32,34,35,36,37,38,39	1 2 5	0.6 0.4 2.1

Dentre estas soluções podemos dizer que 2a., 4a. e 5a. sãs melhores. A 2a. fornece um valor um pouco superior à solução ótima com uma inviabilidade de 3.1%. A 4a. solução fornece um valor, da função objetivo, um pouco inferior ao da solução ótima. E a 5a. solução pode ser considerada a melhor de todas, porque fornece um valor superior as da solução ótima e as inviabilidades nas três restrições podem ser consideradas insignificantes.

<u>Problema 10</u>. $m = 5$, $n = 50$

Este problema possue 50 variáveis e 5 restrições e seu segundo membro será:

i	1	2	3	4	5
b_i	800	650	550	550	650

Solução ótima de Petersen

$z = 16537$

$x_j = 1$ para $j = 4,6,8,9,11,12,13,15,16,17,19,20,23,$
$25,26,27,28,29,31,32,34,35,36,37,$
$38,39,40,41,42,43,44,47,48,49,50$

zero para as outras

Melhor solução viável pelo método Dual de Decomposição

$z = 16499$

$x_j = 1$ para $j = 4,6,7,8,9,11,12,13,14,15,16,17,19,$
$20,23,25,26,27,28,29,30,31,32,34,35,$
$36,37,38,39,40,41,42,43,45,47,48,49,$
50

O Quadro 8 mostra as soluções ξ - viáveis.

Quadro 8

	z	x_j = 1, zero as outras	i das restrições inviáveis	inviabilidade % b_i
1a.	16845	j = 1,4,6,7,8,11,13,14,15,16,17,19,20,22,23,24,25,26,27,29,30,31,32,34,35,36,38,39,40,41,42,43,45,47,48,49,50	1 5	2.2 8.1
2a.	16861	j = 1,4,6,8,9,11,13,14,15,16,17,18,19,20,23,25,26,27,28,29,30,31,32,34,35,37,38,39,40,41,42,43,45,47,48,49,50	1 3 5	1.5 6.7 4.3
3a.	17032	j = 4,6,8,9,11,13,14,15,16,17,19,20,21,23,25,26,27,28,29,30,31,32,34,35,37,38,39,40,41,42,43,44,45,47,48,49,50	1 2 3 5	2.6 8.0 5.8 3.3
4a.	16561	j = 4,6,8,9,11,13,14,15,16,17,18,19,20,21,25,26,27,28,29,30,31,32,34,35,36,37,38,39,40,41,42,43,45,47,48,49,50	1 2 3	1.0 2.1 5.6
5a.	16817	j = 4,6,8,9,11,13,14,15,16,17,19,20,21,23,25,26,27,28,29,30,31,32,34,35,37,38,39,40,41,42,43,45,47,48,49,50	1 2 3 5	1.6 4.6 3.4 0.6
6a.	17069	j = 1,4,6,8,9,11,13,14,15,16,17,19,20,23,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,45,47,48,49,50	1 3 5	6.6 3.4 4.0
7a.	17311	j = 2,4,6,8,9,11,13,14,15,16,17,19,20,23,25,26,27,28,29,30,31,32,34,35,37,38,39,40,41,42,43,45,47,48,49,50	1 2 3 5	6.8 8.0 3.2 5.8

Como se pode observar no Quadro 8, todas as soluções ξ - viáveis fornecem para este caso valores da função objetivo superiores à solução Ótima. Em especial a solução 7a. fornece um valor bastante superior à solução Qtima com inviabilidades inferiores a 10%. Já com inviabilidades inferiores a 5% foi atingido um valor da função objetivo próximo da solução ótima (5a. solução).

<u>Problema 11</u>. $m = 30$, $n = 60$

Os problemas 11 e 12 estão definidos pelo problema apresentado por S.Senju - Y.Toyoda [92] que utilizam uma matriz com 60 variáveis e 30 restrições. Os elementos da matriz são não-negativos e diferentes da unidade. Assim queremos testar a eficiência do algoritmo para esta classe de problemas,

O problema 11 corresponde ao problema A da referência [92] ,e a diferença dos problemas 11 e 12 somente está no segundo membro das restrições.

Senju - Toyoda utilizando um computador B 5500 obtêm como melhor solução um valor da função objetivo igual a 7700 com 19 variáveis iguaisa 1.

Utilizando o método Dual de Decomposição a melhor solução viável corresponde a:

$$z = 7739$$

$$x_j = 1 \quad \text{para } j = 2,3,5,8,13,14,16,18,19,25,27,30,31,$$
$$36,37,43,46,47,53,56,59$$

As soluções ξ - viáveis obtidas constam no Quadro 9.

Quadro 9

	z	x_j = 1, zero as outras	i das restrições inviáveis	inviabilidade % b_i
1a.	7756	j = 2,5,8,13,14,15,16,17,18, 19,20,21,25,27,30,37,45, 46,47,53,56,59	5 8 15 26	10.0 10.3 0.8 18.2
2a.	7805	j = 2,5,8,9,13,14,16,18,19, 20,21,25,27,29,30,37,39, 46,47,53,56,59	1 3 5 11 18 21 26	8.7 8.5 7.9 12.2 2.0 4.1 7.8
3a.	7850	j = 2,5,8,13,14,16,18,19,21, 25,27,30,31,36,37,46,47, 48,53,56,57,59	1 2 9 11 12 13 15 16 18 19	13.2 11.3 3.5 0.7 2.9 1.9 15.8 13.0 5.2 0.1
4a.	7767	j = 2,3,5,8,13,14,16,18,19,20, 25,27,30,31,36,37,45,46, 47,53,56,59	8 9 11 15 26	1.8 5.7 7.7 6.8 13.5
5a.	7723	j = 2,5,8,9,13,14,16,18,19,25, 27,30,31,37,43,45,46,47, 53,56,59	3 15	12.9 2.0
6a.	7795	j = 2,5,8,13,14,16,18,19,21, 25,27,30,36,37,39,45,46, 47,53,56,59	25 26	4.0 5.7

continuação .Quadro 9

	z	x_j = 1, zero as outras	i das restrições inviáveis	inviabilidade % b_i
7a.	7847	j = 5,8,9,13,14,16,18,19,21, 36,37,39,45,46,47,53,56, 57,59	2	10.6
			3	4.5
			9	2.3
			13	1.8
			15	4.8
			21	4.2
			24	0.3
			25	5.6
			26	7.2
8a.	7797	j = 2,5,8,13,14,16,18,19,21, 25,27,30,31,36,37,46,47, 53,56,57,59	2	5.0
			15	1.1
			16	4.7
9a.	7896	j = 2,5,8,13,14,16,18,19,25, 27,30,31,36,37,39,43,45, 46,47,53,56,57,59	2	11.8
			3	3.5
			5	1.1
			8	3.1
			9	2.9
			14	4.3
			15	15.7
			16	7.1
			24	0.6
			25	1.6
			26	2.8
10a.	7842	j = 2,5,8,13,14,16,18,19,21, 25,27,30,31,37,39,43,46, 47,53,56,57,59	1	10.1
			2	5.8
			3	2.5
			5	12.6
			15	1.8
			16	7.1
			24	0.2

continuação Quadro 9

	z	$x_j = 1$, zero as outras	i das restrições inviáveis	inviabilidade % b_i
11a.	7791	j = 3,5,8,13,14,16,18,19,21, 25,27,30,31,36,37,39,46, 47,53,56,57,59	2 9 21 24 25	7.1 8.7 5.1 1.2 5.3
12a.	7809	j = 5,8,9,13,14,16,18,19,20, 25,27,30,31,36,37,43,46, 47,53,56,57,59	2 3 5 11 26	17.3 17.3 0.6 5.9 1.2
13a.	7776	j = 5,8,13,14,16,18,19,20,25, 27,30,31,36,37,43,46,47, 53,56,57,59	2 3 11 26	13.4 1.1 4.8 0.5
14a.	7831	j = 5,8,13,14,16,18,19,21,22, 25,27,30,31,36,37,39,43, 46,47,53,56,59	3 5 9 11 14 18 24 25	2.5 13.7 15.1 6.8 15.1 13.7 12.4 5.5
15a.	7785	j = 5,8,13,14,16,18,19,21,25, 27,30,31,36,37,39,46,47,53, 56,57,59	2 25	6.6 5.2

Existem outras soluções ξ - viáveis e inviáveis, porém somente constam no Quadro 9 aquelas ξ - viáveis consideradas mais interessantes, Estas soluções podem ser utilizadas para uma análise de pós-otimização do problema. O tempo de computador (B 6700) gasto, para encontrar estas soluções foi de 6.74 segundos.

Problema 12. m = 30, n = 60

Corresponde ao problema B da referência [92].

Solução obtida por Senju - Toyoda

$z = 8685$

com 34 variáveis iguais a um,

Melhor solução viável obtida pelo Método Dual de Decomposição

$z = 8675$

$x_j = 1$ para j = 2,3,4,5,8,9,13,14,15,16,17,18,19,21, 25,27,29,30,34,36,37,39,41,43,44,45, 46,47,48,53,56,57,59

$x_j = 0$ para o resto de variáveis

No Quadro 10 constam as soluções ξ - viáveis.

Quadro 10

	z	x_j = 1, zero as outras	i das restrições inviáveis	inviabilidade % b_i
1a.	8732	j = 2,4,5,8,9,13,14,16,17,18, 19,21,25,26,27,28,29,30, 31,34,36,37,39,42,43,44, 45,46,47,48,53,56,59	3 15 24 25 26	8.8 2.1 6.1 9.3 5.2
2a.	8775	j = 2,3,4,5,8,9,13,14,16,17, 18,19,20,25,27,28,29,30, 34,36,37,39,41,42,43,44,45, 46,47,48,53,56,57,59	15 24 25 26	7.6 3.0 3.1 5.8

continuação Quadro 10

	z	x_j = 1, zero as outras	i das restrições inviáveis	inviabilidade % b_i
3a.	8836	j = 2,3,4,5,8,9,11,13,14,15, 16,17,18,19,20,25,27,29, 30,34,36,37,39,41,43,44, 45,46,47,48,49	2 3 4 5 8 15 24 25	7.9 7.8 1.0 0.9 0.03 4.6 10.0 5.3
4a.	8812	j = 2,4,5,8,9,13,14,15,16,17, 18,19,21,25,27,29,30,31, 34,35,36,37,39,41,42,43, 44,45,46,47,48,53,56,57,59	3 5 8 15 24 25	2.7 1.6 9.5 4.8 0.7 15.8
5a.	8726	j = 2,4,5,8,9,13,14,15,16,17, 18,19,20,21,25,27,28,29,30, 34,36,37,39,41,42,45,46,47, 48,53,56,57,59	15 25 26	1.4 5.4 8.4
6a.	8698	j = 2,4,5,8,9,13,14,15,16,17, 18,19,21,25,27,28,29,30,34, 36,37,39,41,43,44,45,46,47, 53,56,57,59	24	2.0
7a.	8690	j = 2,4,5,8,9,13,14,15,16,17, 18,19,20,21,25,27,30,34,36, 37,39,41,42,43,44,45,46,47, 48,53,56,57,59	3 8 26	1.0 2.5 2.3
8a.	8922	j = 2,4,5,8,9,11,13,14,15,16,17, 18,19,21,25,27,28,29,30,34, 36,37,39,41,42,43,44,45,46, 47,48,53,56,57,59	1 2 3 5 15 16 24 25 26	0.06 4.6 4.7 7.1 6.1 1.1 4.3 14.2 1.0

continuação Quadro 10

	z	$x_j = 1$, zero as outras	i das restrições inviáveis	inviabilidade % b_i
9a.	8738	j = 2,5,8,9,11,13,14,15,16,17, 18,19,20,21,25,27,29,30, 31,34,36,37,39,41,44,45, 46,47,48,53,56,57,59	1	0.06
			2	4.7
			3	2.0
			5	3.8
			8	5.9
			15	3.3
			25	1.9
10a.	8860	j = 2,4,5,8,9,11,13,14,15,16, 17,18,19,20,21,25,27,28, 29,30,34,36,37,39,41,44, 45,46,47,48,53,56,57,59	2	5.2
			4	0.4
			5	4.2
			15	8.1
			24	3.2
			25	6.4
			26	1.7
11a.	8679	j = 2,3,4,5,8,9,13,14,16,17, 18,19,20,25,27,28,30,34, 36,37,39,41,42,44,45,46, 47,48,49,53,56,57,59	15	3.9
			24	4.0
			26	8.2
12a.	8717	j = 2,3,4,5,8,9,13,14,15,16, 17,18,19,20,25,27,28,30, 34,36,37,39,41,43,44,45, 46,47,48,49,53,56,57,59	3	1.3
			15	4.0
			24	4.7
			26	2.7
13a.	8802	j = 2,4,5,8,9,13,14,15,16,17, 18,19,20,21,25,27,28,29, 30,34,36,37,39,41,43,44, 45,46,47,48,53,56,57,59	3	1.5
			5	6.0
			8	4.0
			15	7.9
			24	2.6
			26	2.1
14a.	8811	j = 2,3,4,5,8,9,13,14,15,16, 17,18,19,20,21,25,27,28, 29,30,34,36,37,39,41,42, 44,45,46,47,49,53,56,57, 59	15	2.7
			16	3.8
			24	12.8
			25	6.3
			26	12.4

continuação Quadro 10

	z	$x_j = 1$, zero as outras	i das restrições	inviabilidade % b_i
15a.	8838	j = 2,4,5,8,9,11,13,14,15,16, 17,18,19,20,21,25,27,29, 30,34,36,37,39,41,42,43, 44,45,46,47,53,56,57,59	2 4 5 8 24 25 26	4.0 1.6 5.7 2.2 1.0 13.6 2.6
16a.	8782	j = 2,4,5,8,9,11,13,14,15,16, 17,18,19,20,25,27,28,30, 34,36,37,39,41,42,43,44, 45,46,47,53,56,57,59	2 5 25 26	3.2 3.1 3.2 5.0
17a.	8677	j = 2,4,5,8,9,13,14,15,16,17, 18,19,25,27,29,30,34,36, 37,39,41,42,43,44,45,46, 47,49,53,56,57,59	24 25	8.2 3.6
18a.	8682	j = 2,4,5,8,9,13,14,15,16,17, 18,19,20,25,27,29,30,34, 36,37,39,41,42,44,45,46, 47,48,53,56,57,59	25	3.2
19a.	8816	j = 2,4,5,8,9,13,14,15,16,17, 18,19,20,25,27,28,29,30, 34,36,37,39,41,42,43,44, 45,46,47,48,53,56,57,59	3 15 16 24 25 26	1.2 7.8 1.1 3.1 3.3 5.9
20a.	8739	j = 2,4,5,8,9,11,13,14,15,16, 17,18,19,20,25,27,30,34, 36,37,39,41,43,44,45,46, 47,48,49,53,56,57,59	2 3 24	6.8 6.9 1.6

As soluções ξ - viáveis que constam nos quadros anteriores correspondem a valores da função objetivo superiores às soluções encontradas por Senju-Toyoda [92] e pelo método

Dual de Decomposição. Existem portanto, outras soluções ξ - viáveis e viáveis que em caso de uma análise de sensibilidade podem tornar-se interessantes.

O tempo gasto para a obtenção destas soluções foi de 8.34 segundos utilizando um computador B 6700.

Problema 13. $m = 2$, $n = 28$

Utilizando os problemas descritos por Weingartner - Ness [98], que se referem ao problema de alocação de capital para diversos projetos cujo modelo não é outra coisa que o problema Knapsack multidimensional, foram encontradas as soluções para os problemas 13 e 14.

O problema 13 corresponde ao da Tabela 1 da referência |98|.

Solução ótima de Weingartner - Ness

$z = 141.278$

$x_j = 1$ para $j = 3,5,6,7,8,10,12,13,14,19,21,23,24,26$

$x_j = 0$ para o resto

Melhor solução viável encontrada pelo Método Dual de Decomposição

$z = 140.477$

$x_j = 1$ para $j = 3,5,6,7,8,10,12,13,14,15,17,19,21,$
$23,24,27$

O Quadro 11 mostra as soluções ξ - viáveis.

Quadro 11

	z	$x_j = 1$, zero as outras	i das restrições	inviabilidade % b_i
1a.	143228	j = 3,5,6,7,8,10,12,13,14,19, 21,23,24,26,27	2	2.1
2a.	142393	j = 3,5,7,8,10,12,14,21,22, 23,24,26,27	2	3.8

Nas duas soluções as inviabilidades são desprescíveis. Como se trata de seleção de projetos, o conjunto de projetos selecionados em cada caso são bastante diferentes, dando lugar à escolha de alternativas diferentes.

Problema 14. m = 2, n = 105

Este problema corresponde à Tabela 3 da referência |98|

Solução ótima de Weingartner - Ness

$z = 1.095.445$

$x_j = 0$ para j = 13,79,80,84,87,93 até 105

$x_j = 1$ para as demais

Melhor solução viável encontrada pelo Método Dual de Decomposição

$z = 1.094.757$

$x_j = 0$ para j = 12,79,84,87,92,94,95,96,98,99,100,102, 103,104,105

$x_j = 1$ para as demais

No Quadro 12 constam as soluções ξ - viáveis.

Quadro 12

	z	$x_j = 0$, um para as outras	i das restrições	inviabilidade % b_i
1a.	1095131	j = 11,12,79,80,84,87,94 até 105	2	0.3
2a.	1094762	j = 79,87,92 até 105	1	2.4
3a.	1096796	j = 12,79,84,87,94, até 105	2	0.6

As inviabilidades apresentadas são desprecíveis, especialmente as das soluções 1a. e 3a., sendo que a 3a. fornece um maior valor da função objetivo.

Nos problemas L3 e 14 a % das inviabilidades é bastante baixa, e isto se deve ao fato de que o número de variáveis é muito maior que o número de restrições.

Os problemas 13 e 14 foram rodados em conjunto e o tempo de computação necessário para achar as soluções £oi de 7.0 segundos,

3. Conclusões

Mediante as soluções destes 14 problemas pode-se verificar que as melhores soluções viáveis encontradas estão próximas das soluções ótimas, sendo que para isto o tempo de computação é muito inferior ao tempo gasto por outras métodos e o número de iterações necessárias também é menor, além disso as soluções chamadas de ξ - viáveis fornecem alternativas para a escolha de uma solução. Assim, estas soluções podem ser utilizadas para uma análise de pós-otimização dos modelos, isto sem outro esforço de cálculo adicional.

A medida que o número de variáveis é muito maior que o número de restrições, a % das inviabilidades nas soluções - viáveis é menor ou o número destas soluções aumenta,

É conhecido por todos aqueles que lidam com modelos matemáticos, que para implementar uma solução do modelo, a solução escolhida nem sempre é a solução ótima fornecida para o modelo e se faz necessária a apresentação de outras alternativas, que de preferência deverão estar próximas da ótima.

Dentro deste critério, o método Dual de Decomposição é recomendado para seu utilizado quando nos modelos não seja estritamente necessário a obtenção da solução Ôtima.

Também este método poderá vir a ser utilizado como um ponto de partida para a obtenção da solução ótima. Ficando isto como uma idéia para um trabalho que pode ser desenvolvido posteriormente.

BIBLIOGRAFIA

1 - Arora S.R., "An Enumerative Technique for the Set Covering Problem", N.Z. Opns. Res., 5, 119-128 (1977)

2 - Balas E., "Set Covering with Cutting Planes fxom Conditional Bounds", M.S.R.R. Nº 399, Carnegie-Mellon University (1977)

3 - Balas E. and Martin C.H., "Pivot and Complement - A Heuristic for 0-1 Programming", M.S.R.R. Nº 414, Carnegie-Mellon University (1978)

4 - Balas E. and Zemel E., "Graph Substitution and Set Packing Polytopes", Networks an Inter. Journal, 7, 267-284 (1977)

5 - Balas E. and Zemel E., "All Facets of Zero-One Programming Polytopes with Positive Coefficients", M.S.R.R. Nº 374, Carnegie-Mellon University (1975)

6 - Balas E., "Facets of the Knapsack Polytope", Math. Programming, 8, 146-164 (1975)

7 - Balas E., "Disjunctive Programming,: Proprerties of the Convex Hull of Feasible Points", M.S.R.R. Nº 348, Carnegie-Mellon University (1974)

8 - Balas E., "Minimax and Duality for Linear and Nonlinear Programming - Ed. J. Abadie, Nort-Holland Pub. Co. (1970)

9 - Balas E., "Duality in Discrete Programming II: The Quadratic Case", Management Sci., 16, 14-32 (1969)

10 - Balas E., "An Additive Algorithm for Solving Linear Programs with Zero-One Variables", Opns. Res., 13, 517-546 (1965)

11 - Balas E., "Discrete Programming by the Filter Method", Opns. Res., 15, 915-957 (1967)

12 - Bell D. and Shapiro J., "A Convergent Duality Theory for Integer Programming", Opns. Res., 25, 419-434 (1977)

13 - Bell D.E., "Constructive Group Relaxations for Integer Programs", SIAM J. Appl. Math., 30, 708-719 (1976)

14 - Bradley G., "Equivalent Mixed Integer Programming Problems", Opns. Res., 21, 323-326 (1973)

15 - Bradley G. and Wahi P., "An Algorithm for Integer Linear Programming: A Combined Algebraic and Enumeration Approach", Opns. Res., 21, 45-60 (1973)

16 - Bowman V. and Glover F., "A Note on Zero-One Integer and Concave Programming", Opns. Res., 20, 182-183 (1972)

17 - Bellmore M. and Ratliff H. D., "Set Covering and Involutory Bases", Management Sci., 18, 194-206, (1971)

18 - Brooks, R.B.S. and Geoffrion, A.M., "Finding Everett's Lagrange Multipliers by Linear Programming", Opns. Res., 14, 1149-1153 (1966)

19 - Balinski, M.L., "Integer Programming: Methods, Uses, Computation", Management Sci., 12, 253-313 (1965)

20 - Cabot A.V., "An Enumeration Algorithm for Knapsack Problems", Opns. Res. 18, 306-311 (1970)

21 - Charnes A., Granot D. and Granot F., "On Intersections Cuts in Interval Integer Linear Programming", Opns. Res., 25, 252-355 (1977)

22 - Dantzig G., Linear Programming and Extensisns, Princeton Univ. Press Princeton, N.J, 1963

23 - De Maio A. and Roveda C., "An All Zero-One Algorithm for Certain Class of Transportation Problems", Opns. Res., 19, 1406-1418 (1971)

24 - Etcheberry J. "The Set-Covering Problem: A New Implicit Enumeration Algorithm", Opns. Res. 25, 760-772 (1977)

25 - Ellwein L.B., "A Flexible Enumeration Scheme for Zero-One Programming", Opns. Res., 22, 144-150 (1974)

26 - Eilon S. and Christofides N., "The Loading Problem", Management Sci., 17, 259-268 (1971)

27 - Everett III H., "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources", Opns. Res. 11, 399-417 (1963)

28 - Fisher M.L., "Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I", Opns. Res., 21, 1114-1127 (1973)

29 - Fisher M.L. and Shapiro J.F., "Constructive Duality in Integer Programming", SIAM J. Appl. Math. 27, 31-52 (1974)

30 - Faaland B., "Solution of the Value-Independent Knapsack Problem by Partitioning", Opns. Res. 21, 332-336 (1973)

31 - Fox B.L. and Landi D.M., "Searching for the Multiplier in One-Constraint Optimization Problems", Opns. Res, 18, 253-262 (1970)

32 - Garfinkel R.S., "Minimizing Wallpaper Waste, Part 1: A Class of Traveling Salesman Problems", Opns. Res. 25, 741-751 (1977)

33 - Garfinkel R.S. and Nemhauser G.L., Integer Programming John Wiley e Sons, 1972

34 - Greenberg H., "The One-Dimensional Generalized Lagrange Multiplier Problem", Opns. Res. 25, 338-345 (1977)

35 - Golden B.L., "Approaches to the Cutting Stock Problem", AIIE Transactions, 8, 265-274 (1976)

36 - Glever F. and Woolsey E., "Converting the Zero-One Polynomial Programming Problems to a Zero-One Linear Programming", Opns. Res. 22, 180-182 (1974)

37 - Gorry G.A., Northup W.D. and Shapiro J.F., "Computational Experience with a Group Theoretic Integer Programming Algorithm", Math. Programming, 4, 171-192 (1973)

38 - Gearing C.E., Swart W.W. and Var T., "Determining the Optimal Investment Policy for the Tourism Sector of a Developing Country", Management Sci., 20, 487-497 (1973)

39 - Geoffrion A.M. and Graves G.W., "Multicommodity Distribution System Besign by Benders Decomposition", Management Sci, 20, 822-844 (1974)

40 - Glover F. and Klingman D., "Concave Programming Applied to a Special Class of 0-1 Integer Programs", Opns. Res., 21, 135-140 (1973)

41 - Glover F. and Woolsey R., "Futher Reduction of Zero-One Polynomial Programming Problems", Opns. Res. 21, 156-161, (1973)

42 - Geoffrion A., "Duality in Nonlinear Programming: A simplified Applications-Oriented Development", Perspectives on Optimization, Ed. A.M. Geoffrion, Addison-Wesley Co, pp. 65-101

43 - Garfinkel R. and Nemhauser G.L., "Optimal Set Covering:: A Survey", Perspectives on Optimization, Ed. A.M. Geoffrion, Addison-Wesley Co, pp 164-193 (1972)

44 - Gorry G.A. and Shapiro J.F., "An Adaptive Group Theoretic Algorithmic for Integer Programming", Management Sci. 17, 285-306 (1971)

45 - Glover F., "A Note on Extreme-Point Solutions and a Paper by Lemke, Salkin and Spielberg", Opns. Res. 19, 1023-1025 (1971)

46 - Garfinkel R.S. and Nemhauser G.L., "Optimal Political Districting by Implicit Enumeration Techniques", Management Sci., 16, B-495 - B-508, (1970)

47 - Greenberg H. and Hegerich R.L., "A Branch Algorithm for the Knapsack Problem", Management Sci., 16, 327-332 (1970)

48 - Glover F., "Surrogate Constraints", Opns. Res. 16, 741-749, (1968)

49 - Geoffrion A.M., "Integer Programming by Implicit Enumeration and Balas Method", SIAM Rev., 7, 178-190 (1967)

50 - Gilmore P., "The Theory and Computation of Knapsack Functions, Opns. Res. 14, 1045-1074 (1966)

51 - Glover F., "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem", Opns. Res., 13, 879-919 (1965)

52 - Glover F. and Zisnts S., "Note on the Additive Algorithm of Balas", Opns. Res., 13, 546-549 (1965)

53 - Gilmore P., "Multi-Stage Cutting Stock Problems of Two and More Dimensisns", Opns. Res. 13, 94-120 (1965)

54 - Greenberg H., Integer Programming, Academic Press

55 - Hu T.C., Integer Programming and Network Flows, Addison-Weley Pub. Co. 1970.

56 - Houck D.J. and Vemuganti R.R., "An Algorithm for the Vertex Packing Problem", Opns. Res., 25, 773-787 (1977)

57 - Hu T.C. and Leonard M.L., "Optimality of a Heuristic Algorithm for a class of Knapsack Problems", Opns. Res., 24, 143-196 (1976)

58 - Held M., Wolfe P. and Crowder H.P., "Validation of Subgradient Optimization", Math. Progr., 6, 62-88 (1974)

59 - Held M. and Karp R.M., "The Traveling-Salesman Problem and Minimum Spaning Trees: Part II", Math. Progr., 1, 6-25 (1971)

60 - Hillier F.S., "Efficient Heuristic Procedures for Integer Linear Programming with an Interior", Opns. Res., 17, 600-637 (1969)

61 - Ingargiola G.P. and Korsh J.F., "A Reduction Algorithm for 0-1 Single Knapsack Problems", Management Sci., 20, 460-463 (1973)

62 - Ingargiola G.P. and Korsh J.F., "A General Algorithms for One-Dimensional Knapsack Problems", Opns. Res. 25, 752-759 (1977)

63 - Kendall K.E. and Zionts S., "Solving Integer Programming Problems by Aggregating Constraints", Opns. Res., 25, 346-351 (1977)

64 - Kianhar F., "Stronger Inequalities for 0-1 Integer Programming: Computational Refinements", Opns. Res., , 581-585, (1976)

65 - Kraft D. and Hill T., "The Journal Selection Problem in a University Library System", Management Sci., 19, 613-626, (1973)

66 - Kianfar F., "Stronger Inequalities for 0-1 Integer Programming Using Knapsack Functions", Opns. Res., 19, 1374-1392, (1971)

67 - Kolesar P.J., "A Branch and Bound Algorithm for the Knapasack Problem", Management Sci., 13, 723-735 (1967)

68 - Kaplan S., "Solution of the Lorie-Savege and Similar Integer Programrning Problems by Generalized Lagrange Multiplier Method", Opns. Res., 14, 1130-1136 (1966)

69 - Lasdon L.S., Optimization Theory for Large Systems, McMillan Series for Operations Reserch (1970)

70 - Legendre J.P. et Minoux M., "Une Application de la Nstion de Dualité en Programmation en Nombres Entieres: Sélection et Affectation Optimales d'une Flotte d'Avions", R.A. I.R.O. Recherche Opérationnelle, 11, 201-222 (1977)

71 - Lemke C.E., Salkin H.M. and Spielberg K., "Set Covering by Single Branch Enumeration with Linear Programming Sub-problems", Opns. Res., 19, 998-1022 (1971)

72 - Maculan Fº. N., Vaca Obando F.E. e Villares M.L., "Tópicos de Dualidade em programação Inteira Bivalente ", II Congresso Brasileiro de Automática, 1978

73 - Magazine M., Nemhauser G.L. and Trotter Jr. L.E., "When the Greedy Solutions Solves a Class sf Knapsack Problems", Opns. Res., 23, 207-217 (1975)

74 - Mao J. and Wallingford, "An Extension of Lawler and Bell's Method of Biscret Optimization with Examples from Capital Budgeting", Management Sci., 15, B51-B60 (1968)

75 - Maculan Fº.N., programação Linear Inteira, COPPE/UFRJ, PDD 17/78

76 - Nauss R.M., "An Algorithm for the 0-1 Integer Linear Programming Problem", Res. Rep. Univ. Missouri-St. Louis, October, 1976

77 - Nemhauser G.L. and Yu P.L., "A Problem in Bulk Service Scheduling", Opns. Res., 20, 813-819 (1972)

78 - Nemhauser G.L. and Ullmann Z., "Discrete Dynamic Programming and Capital Allocation", Management Sci., 15, 494-505 (1969)

79 - Nemhauser G.L. and Ulmann Z., "A Note on the Generalized Langrange Multiplier Sslution to an Integer Programming

Problem", Opns. Res., 16, 450-452 (1968)

80 - Padberg M., "A Note on Zero-One Programming", Opns. Res., 23, 833-837 (1975)

81 - Peterson D.E. and Laughhunn D.,,"Capital Expediture Programming and Some Alternative Approaches to Risk", Management Sci., 17, 320-336 (1971)

82 - Pierce J., "Application of Combinatorial Programming to a Class of All-One Integer Programming Problems", Management Sci., 15, 191-209 (1968)

83 - Petersen C., "Computational Experience with Variants of the Balas Algsrithm Applied to the Selection of R and D Projects", Management Sci., 13, 736-750 (1967)

84 - Ragnavachari M., "On the Zero-One Integer Programming Problem", Opns. Res., 17, 680-684 (1969)

85 - Salkin H., Integer Programming, Addison-Wesley Publishing Co., 1975

86 - Soyster A.L., Lev B. and Slivka W., "Zero-One Programming with Many Variables and Few Constraints", European J. Opns. Res., 2, 195-201 (1978)

87 - Shih W., "A Branch and Bound Procedure for a Class of Discrete Resource Allocation Problems with Several Constraints", Opns. Res. Q., 28, 439-451 (1977)

88 - Samuelsson H.M., "Integer Programming Duality for . Set-Packing-Partitioning Problems", M.S.R.R. N? 337, Carnegie-Mellon Univ. (1974)

89 - Salkin H.M. and Koncal R.D., "Set Covering by All Integer Algorithm: Computational Experience", J.A.C.M, Nº 20, 189-193 (1973)

90 - Shapiro J.F., "Generalized Langrage Multipliers in Integer Programming", Opns. Res., 19, 68-76 (1971)

91 - Shapiro J.F., "Dynamic Programming Algorithms for the Integer Programming Problem I: The Integer Programming Problem Viewed as a Knapsack Type Problern", Opns. Res., 16, 103-121 (1968)

92 - Senju S. and Toyoda Y., "An Approach to Linear Programming with Zero-One Variables", Management Sci., 15, B-196 - B-207

93 - Shapiro S. and Wagner H., "A Finite Renewal Algorithm for the Knapsack and Turnpike Models", Opns. Res., 15, 319-341 (1967)

94 - Trauth C.A. and Woolsey R.E., "Integer Linear Programming: A Study in Computational Effectiveness", Management Sci., 16, 481-494 (1969)

95 - Taha H.A., Integer Programming: Thesry, Applications, and Computations Academic Press 1975

96 - Toyoda S., "A Simplified Algorithm for Obtaining Approximate Solutions to 0-1 Programming Problems", Management Sci., 21, 1417-1427 (1975)

97 - Varaiya P.P., Notes on Optimization, Vsn Nostrand Reinhold Co., 1971

98 - Weigartner H. and Ness D., "Methods for Solution of 0-1 Knapsack Problems", Opns. Res., 15, 83-103 (1967)

99 - Wolsey L.A., "Valid Inequalities and Superadditivity for 0-1 Integer Programs", Math of Opns. Res., 2, 66-77 (1977)

100 - Zanakis S.H., "Heuristic 0-1 Linear Programming: An Experimental Comparison of Three Methods", Management Sci., 1, 91-104 (1977)

101 - Zanwill W.I., Nonlinear Programming: A Unified Approach-Hall 1969

102 - Zisnts S., Linear and Integer Programming, Prentice-Hall Inc, 1974