Agregação de Variáveis em Programação Linear

Betty Rimarachín López

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO

PARA OBTENÇÃO DO GRAU
DE
MESTRE EM MATEMÁTICA APLICADA

Área de Concentração: **Otimização**
Orientadora: **Prof ª. Dr ª. Ana Flora P. de Castro Humes**

Durante a elaboração deste trabalho,
a autora recebeu apoio do LCCA e apoio financeiro do CNPq

-São Paulo, Setembro de 1999-

Agregação de Variáveis em Programação Linear

Este exemplar corresponde à
redação final da dissertação
devidamente corrigida e defendida
por Betty Rimarachín López e aprovada
pela comissão julgadora.

São Paulo, 17 de Setembro de 1999.

Banca examinadora:

- Profa. Dra. Ana Flora Pereira de Castro Humes (Orientador) - IME-USP
- Prof. Dr. Manuel Valentim de Pera Garcia - IME-USP
- Prof. Dr. Nelson Maculan Filho - UFRJ

A minha mãe Luvinda, com carinho.

Agradecimentos

Este trabalho é parte de um ideal que não teria sido possível realiza-lo sem a ajuda, o incentivo e o companheirismo de amigos, colegas e professores. Estendo a todos a minha gratidão.

A meus pais, sem os quais não chegaria aqui.

Sou grato em especial a minha orientadora e amiga Prof^{a}. Ana Flora P. de Castro Humes, pela sugestão do tema, pelos conselhos, pela nossa amizade, pela paciência durante o curso de Mestrado e orientação desta dissertação.

Agradeço aos membros da comissão julgadora, pela observação de imprecisões e erros, na versão preliminar deste trabalho, e por suas valiosas sugestões.

Ao CNPq pelo auxilio financeiro.

Ao LCCA por ter-me permitido o aceso a seu laboratório de Computação.

Sou grato também à Prof. Carlos Humes pela crítica, conselhos e sugestões que me ajudaram a conhecer melhor a área de otimização.

Agradeço a Paulo Silva e a Marcelo Gomes de Queiroz pelas críticas e por ter me ajudado a resolver algumas duvidas durante o desenvolvimento deste trabalho.

Aos meus irmãos Leonel, Alex, Dina, Melva e Juan Carlos pelo apoio moral e incentivo.

A minhas queridas amigas Katia García A., Raquel de Jesus, Rosemary de Oliveira F. e Magem Infante, que me ajudaram em diversos momentos difíceis.

A Carlos Pantaleón, pela amizade e ajuda em programação.

A os meus compatriotas, especialmente a Gaspar Salas, Esteban Tuesta, Santos Enriquez R, Walter Cárdenas, Delhi Paiva, Nelson Vidaurre, Hernán Cuti, Xyoby Chávez e Nestor Castañeda, que tornaram mais agradável minha estadia no Brasil.

Pela oportunidade de poder melhorar a minha formação acadêmica e pelas facilidades dispensadas, agradeço à Universidade de São Paulo.

A os amigos brasileiros Sergio Munhoz, Cybele Dunder que me deram o prazer de suas companhias.

A todos, que de uma forma ou de outra ajudaram na concretização deste trabalho.

Resumo

Neste trabalho tratamos da metodologia de agregação de variáveis de um problema de programação linear e, no caso mais geral, agregação simultânea de variáveis e de restrições. Enfatizamos a discussão de diversos limitantes, superior e inferior do valor ótimo do problema original, fornecidos pelo problema agregado. Em particular, recordamos a existência de um peso ótimo para o qual a solução do problema agregado fornece a solucao ótima do problema original. Esta revisão é preparatoria para a questão central que é o problema de como melhorar os pesos de agregação com a finalidade de obter a solução quase-ótima. É proposto para isto uma estratégia baseada no método dos sub-gradientes. Fazemos ainda uma aplicação do método de agregação de variáveis para problemas particulares de programação linear inteira. Alguns testes computacionais preliminares são apresentados.

Abstract

This work deals with the methodology of variables aggregation and the case of simultaneous aggregation of variables and restrictions for a linear programming problem. We enfatize the discusion of many upper and lower bounds for the valor otimo of the original problem which is given from the aggregated problem. Particularly, we remember the existance of an optimal weight such that the solution of the aggregated problem gives the optimal solution for the original problem. This review is preparatory to the central question which is the problem concerning how to improve the aggregation weight so to obtain the quase-optima solution. To this one proposes a strategy based on a subgradient scheme. We present also an application of aggregation of variables to an particular integer programming problem. Some preliminar computational tests are presented.

Sumário

Introdução **2**

1 Agregação em Programação Linear **4**
1.1 Introdução 4
1.2 Agregação de variáveis 5
1.2.1 Construção do problema agregado 5
1.2.2 Desagregação 7
1.2.3 Limitantes a posteriori 12
1.2.4 Limitantes a priori 19
1.3 Agregação de restrições e variáveis 26

2 Agregação Iterativa de Variáveis **36**
2.1 Introdução 36
2.2 Problema auxiliar 1 37
2.3 Aplicação do metodo do gradiente condicional 57
2.4 Estratégia de agregação iterativa 62

3 Agregação e Computação de Limitantes em Programação Inteira **76**
3.1 Introdução 76
3.2 Outro limitante para Z^* 77
3.3 Limitantes para problemas de programação inteira identificando desigualdades válidas 78

4 Conclusões **89**

Apêndice **90**

A Resultados básicos de Topologia e Cálculo 91

B Implementações Computacionais 96

Bibliografía 126

Introdução

Métodos de agregação-desagregação e métodos de decomposição tem sido desenvolvidos para facilitar a solução prática de problemas de programação matemática de grande porte quando as limitações computacionais tornam impossível resolvê-los. Neste trabalho apresentaremos o método de agregação-desagregação, com ênfase na questão de limitantes.

O método de agregação de variáveis (restrições) consiste em substituir o problema inicial (PL) por outro com menor número de variáveis (restrições) chamado problema agregado. No caso de agregação de variáveis, a solução do problema agregado (PA^V) pode ser desagregada para obter soluções viáveis do PL. Por esta razão, concentramo-nos em agregação de variáveis.

Como o problema agregado normalmente tem informação menos detalhada, as soluções de PA^V não produzem necessariamente a solução ótima de (PL), porém sempre permitem a determinação de soluções viáveis e portanto limites inferiores para o valor ótimo do PL(maximização).

Uma discussão inicial de agregação em programação linear foi feita por Fisher [6] e estendida por Zipkin[23], que conseguiu achar um limitante superior para o valor ótimo do problema inicial PL(maximização), usando a solução dual do agregado. Isto ampliou o interesse, já que a obtenção deste limitante nos permite determinar a proximidade da solução desagregada à solução ótima.

Não é suficiente ter soluções viáveis para o problema inicial (PL), queremos ter soluções o mais próximas o possível do ótimo. Nesta linha é natural os temas de pesquisa em agregação estejam dirigidos a responder as seguintes questões:

1) Como gerar bons problemas agregados para o PL dado.

2) Como construir iterativamente problemas agregados que permitam uma melhor aproximação ao valor ótimo de (PL).

3) Como obter melhores limitantes a partir da solução do problema agregado.

Neste trabalho, abordaremos resultados referentes a diversos limitantes para o valor ótimo de problemas de programação linear e para um caso particular de problemas de programação linear inteira. Também abordaremos métodos de agregação iterativa onde os pesos são modificados em cada iteração, enquanto a partição utilizada para formar o problema agregado permanece sempre fixa. Apresentaremos alguns resultados computacionais.

No primeiro capítulo, apresentamos resultados gerais de agregação de variáveis e de agregação simultânea de variáveis e restrições assim como limitantes do valor ótimo do PL. Os inferiores são obtidos usando desagregação de peso fixo ou desagregação otimal. A estimação de limitantes do erro $(Z^* - Z^V)$ causado por agregação tem atraido a atenção de muitos pesquisadores, entre eles Mendelssohn [18], Taylor[21] e outros. A maioria deles procura melhorar o limitante de Zipkin[23].

O capítulo 2 trata de agregação iterativa. Inicialmente apresentamos um método teórico de agregação iterativa proposto por Litvinchev[16] e baseado na aplicação do método do gradiente condicional usado em problemas de otimização restrita. Dada a dificuldade de usar este método, apresentamos uma estratégia de agregação iterativa usando o gradiente do Lagrangeano associado ao problema agregado. Esta estratégia é proposta por Jörnstein, K. Leisten, R. e Storoy, S.[12]. Implementamos e damos resultados de alguns exemplos numéricos.

No Capítulo 3, apresentamos um resultado de Litvinchev e Rangel[17]. Eles aplicam o método de agregação de variáveis para encontrar um limitante do valor ótimo de problemas de programação linear inteira *multiknapsack*. É natural achar um limitante do valor resolvendo seu problema relaxado. Aqui não é resolvido o relaxado, mas um agregado do relaxado. Seguindo um procedimento semelhante encontramos outro limitante para o valor ótimo de um problema de programação linear inteira. Apresentamos alguns resultados de exemplos numéricos.

Capítulo 1

Agregação em Programação Linear

1.1 Introdução

Neste capítulo é apresentado o método de Agregação e Desagregação usado em problemas de grande porte de Programação Linear. O método consiste em substituir o problema original por outro, obtido fazendo combinações lineares das linhas e/ou colunas da matriz de restrições do problema de programação linear original, denotado por PL. A solução deste novo problema nos permitirá obter estimativas para o valor ótimo do problema inicial PL. Particularmente, se combinarmos só linhas (colunas) referiremo-nos a este método como método de agregação de restrições (variáveis).

Na seção 2 estudaremos a agregação de variáveis e será dividida em 4 subseções. Na primeira detalharemos a forma como é obtido o problema agregado PA^V. Uma vez resolvido o PA^V, o próximo passo é achar soluções viáveis para o problema original. Isto pode ser conseguido com a "*desagregação*" e será tratado na subseção 2. Consideraremos a *desagregação de peso fixo* e a *desagregação otimal.* Para obtermos a solução desagregada otimal, precisamos ter resolvido o problema agregado PA^V e outros pequenos problemas, que denotaremos por $DP_k(\bar{X}_k)$. A desagregação de peso fixo vai nos permitir estabelecer uma relação entre o valor ótimo (Z^*) do problema inicial e o valor ótimo (Z^V) do problema agregado PA^V. Com a desagregação otimal esta relação será melhorada, porque o valor ótimo da solução

desagregada otimal estará mais próximo do valor ótimo do PL. Na subseção 3 apresentaremos os limitantes para o valor ótimo Z^* de PL. Se estes limitantes forem obtidos depois de se ter formado e resolvido o problema agregado PA^V, serão chamados '**limitantes a posteriori**'. Alguns destes limitantes requerem só a solução ótima dual de PA^V e os outros requerem as soluções ótimas duais dos $DP_k(\bar{X}_k)$. Limitante para o erro de agregação definido como a diferença $Z^* - Z^V$ será apresentado na subseção 4, este limitante se denomina '**limitante a priori**' e para obte-lo necessitamos formar o PA^V, sem entretanto precisar resolve-lo.

Na seção 3 será apresentada a agregação simultânea de variáveis e restrições e são repetidos todos os passos feitos para o PA^V. Esta seção não é necessária para o entendimento dos próximos capítulos. Inicialmente detalharemos a forma como é obtido o problema agregado PDA. Na subseção 2, veremos o processo de desagregação de peso fixo e ao contrário do que acontece em PA^V, a solução desagregada tem o mesmo número de componentes que qualquer solução viável de PL, mas não é necessariamente viável para tal problema. Na última subseção obteremos um limitante para o valor ótimo de PL, que é uma generalização do limitante de Zipkin de agregação de variáveis, e um limitante para o erro de agregação definido como anteriormente. O último, chamado **limitante a priori**, é uma generalização do limitante a priori obtido na seção anterior.

1.2 Agregação de variáveis

1.2.1 Construção do problema agregado

Suponha que temos o seguinte problema de programação linear

$$(\mathrm{PL})\left\{\begin{array}{rrr} \mathrm{Z}^* = & & max \quad cx \\ & s.a & Ax \leq b \\ & & x \geq 0 \end{array}\right.$$

onde c é um vetor em $\mathbb{R}^n$, b um vetor em $\mathbb{R}^m$, A uma matriz em $\mathbb{R}^{m \times n}$. Assumimos que PL tenha solução ótima finita. Seja

$$(\mathrm{DL})\left\{\begin{array}{rrr} \mathrm{Z}^* = & & min \quad ub \\ & s.a & uA \geq c \\ & & u \geq 0 \end{array}\right.$$

o problema dual associado a PL.

Definição 1.2.1 *Dizemos que* $\sigma = \{S_k : k = 1 \ldots K\}$ *é uma* **partição** *de* $\{1, \ldots n\}$, *se*

- $S_i \cap S_j = \varnothing, i \neq j$
- $\cup_{i=1}^{K} S_i = \{1, \ldots n\}$

□

Para a partição σ, vamos denotar por n_k a cardinalidade de S_k . Definimos B^k como a sub-matriz de A formada pelas colunas cujos índices estão em S_k. Analogamente definimos os sub-vetores c^k e x^k. Denotaremos por A^j a j-ésima coluna de A.

Definição 1.2.2 *Para cada conjunto* S_k *da partição* σ *, diremos que o vetor* g^k *em* $\mathbb{R}^{n_k}$ *é um* **vetor de multiplicadores ou vetor de pesos** *se cada uma de suas componentes é maior ou igual a zero e a soma delas igual a um. O conjunto dos vetores* g^k, *k=1,...,K chamaremos simplesmente de* **peso** *e o denotaremos por g.*

□

Vamos construir o problema agregado PA^V. Sejam g^k, k = 1,...,K os vetores de multiplicadores. Definimos

$$\bar{A}^k = B^k g^k, \qquad \bar{c}_k = c^k g^k \qquad k = 1, \ldots, K$$

$\bar{A}^k$ é deste modo um vetor coluna obtido através de uma combinação convexa das colunas de A associadas a S_k e analogamente $\bar{c}_k$ é um escalar obtido como combinação convexa das componentes de c associadas a S_k. Definimos

$$\bar{A} = (\bar{A}^1, \ldots, \bar{A}^K) \quad e \quad \bar{c} = (\bar{c}_1, \ldots, \bar{c}_K)$$

O problema agregado PA^V é definido como:

$$(PA^V) \begin{cases} Z^V = & max \quad \bar{c}X \\ & s.a \quad \bar{A}X \leq b \\ & \qquad X \geq 0 \end{cases}$$

e seu respectivo problema dual é

$$(DA^V)\left\{\begin{array}{rrl} Z^V = & & min \quad Ub \\ & s.a & U\bar{A} \geq \bar{c} \\ & & U \geq 0. \end{array}\right.$$

Denotaremos por $\bar{X}$ e $\bar{U}$ as soluções ótimas para PA^V e seu dual DA^V, respectivamente.

Dado o problema original PL, o problema agregado é determinado pelo par (σ, g). Com σ fixo, o problema PA^V depende do peso g . Utilizaremos a notação $PA^V(g)$ quando quisermos enfatizar esta dependência.

O problema agregado PA^V pode ser obtido a partir do problema original PL através da transformação $x = RX$, onde x é variável do PL, X variável do PA^V e R matriz em $\mathbb{R}^{n\times K}$ dada por

$$R_{ij} = \left\{\begin{array}{ll} g_i^j & se \quad i \in S_j \\ 0 & \text{caso contrário.} \end{array}\right.$$

Assim o PA^V fica

$$(PA^V)\left\{\begin{array}{rrl} Z^V = & & \max \quad cRX \\ & s.a & ARX \leq b \\ & & X \geq 0 \end{array}\right.$$

1.2.2 Desagregação

Construído o PA^V, encontraremos soluções viáveis do PL obtidas a partir de soluções viáveis do PA^V . Este processo é conhecido como desagregação. Apresentaremos os seguintes tipos de desagregação:
1) desagregação de peso fixo
2) desagregação otimal.

1. **Desagregação de peso fixo**

   Suponha que X seja viável para PA^V . Construiremos o vetor $x^k \in \mathbb{R}^{n_k}$ tomando $x^k = X_k g^k$, k=1,...,K. Seja x a coleção de vetores x^k, arrumados segundo a ordem das variáveis no PL. Diremos que x é a solução obtida pela desagregação de peso fixo de X e a chamaremos de **solução desagregada de peso fixo**.

Proposição 1.2.3 *Se X é viável para PA^V e x é a solução desagregada de peso fixo, então x é viável para PL e $cx = \bar{c}X$.*

PROVA

$$\begin{aligned} Ax &= \sum_{k=1}^{K} B^k x^k = \sum_{k=1}^{K} B^k (X_k g^k) = \sum_{k=1}^{K} X_k (B^k g^k) \\ &= \sum_{k=1}^{K} X_k \bar{A}^k = \bar{A}X \leq b. \end{aligned} \tag{1.1}$$

Como $x \geq 0$, x é viável para (PL). Por outro lado

$$cx = \sum_{k=1}^{K} c^k x^k = \sum_{k=1}^{K} c^k (X_k g^k) = \sum_{k=1}^{K} X_k (c^k g^k) = \sum_{k=1}^{K} X_k \bar{c}_k = \bar{c}X. \tag{1.2}$$

■

Corolário 1.2.4 *Se $\bar{X}$ é solução ótima para PA^V e $\bar{x}$ é a solução obtida pela desagregação de peso fixo de $\bar{X}$, então $c\bar{x} = Z^V \leq Z^*$.*

PROVA
Pela proposição anterior $\bar{x}$ é viável para PL e $c\bar{x} = \bar{c}\bar{X}$. Como $\bar{X}$ é ótimo para (PA^V), então $\bar{c}\bar{X} = Z^V$, logo

$$Z^V = \bar{c}\bar{X} = c\bar{x} \leq Z^* \tag{1.3}$$

■

Definição 1.2.5 *Diremos que g é* **peso ótimo** *para o PL se com este peso, a solução obtida por desagregação de peso fixo é solução ótima para o problema original PL.*

□

Pelo Corolário 1.2.4, se g é peso ótimo para PL, então $Z^V = Z^*$.

Para qualquer partição σ , existe um peso g^* com o qual o valor ótimo de $PA^V(g^*)$ coincide com o valor ótimo de PL. Infelizmente, a determinação deste peso depende de uma solução ótima (desconhecida) de PL.

Proposição 1.2.6 *Dada uma partição $\sigma = \{S_1, \ldots, S_K\}$ de $\{1, \ldots, n\}$. Se x_0 é uma solução viável de PL e definimos o peso g_0 como*

$$g_0{}^k = \begin{cases} \textit{vetor arbitrário em} \quad \mathbb{R}^{n_k} & \textit{se} \sum_{j \in S_k} x_{0j} = 0 \\ \frac{x_0{}^k}{\sum_{j \in S_k} x_{0j}} & \textit{caso contrário} \end{cases} \tag{1.4}$$

então X_0 definido por $X_{0k} = \sum_{j \in S_k} x_{0j}$ é viável para $PA^V(g_0)$ e $cx_0 = \bar{c}X_0$

PROVA

É claro que $X_0 \geq 0$ e

$$\bar{A}X_0 = \sum_{k=1}^{K} X_{0k}\bar{A}^k = \sum_{k=1}^{K}(\sum_{j \in S_k} x_{0j})B^k g_0{}^k = \sum_{k=1}^{K} B^k x_0{}^k = Ax_0 \leq b \tag{1.5}$$

Assim X_0 é viável para $PA^V(g_0)$. Notemos que a solução desagregada de peso fixo correspondente a X_0 é x_0. Então pela proposição 1.2.3 temos

$$cx_0 = \bar{c}X_0.$$

■

Proposição 1.2.7 *Dada uma partição $\sigma = \{S_1, \ldots, S_K\}$ de $\{1, \ldots, n\}$ e uma solução ótima x^* de PL. Existe um peso g^* que é peso ótimo para o PL.*

PROVA

Pela proposição 1.2.6, escolhendo o peso como

$$g^{*k} = \begin{cases} \text{vetor arbitrário em} \quad \mathbb{R}^{n_k} & se \sum_{j \in S_k} x_j^* = 0 \\ \frac{x^{*k}}{\sum_{j \in S_k} x_j^*} & \text{caso contrário} \end{cases} \tag{1.6}$$

sabemos que o vetor $X^* = (X_1^*, \ldots, X_K^*)$ tal que $X_k^* = \sum_{j \in S_k} x_j^*$ é solução viável para $PA^V(g^*)$ e

$$Z^* = cx^* = \bar{c}X^*. \tag{1.7}$$

Por ser X^* viável para o agregado $PA^V(g^*)$ temos

$$\bar{c}X^* \leq Z^V \leq Z^*.$$

De (1.7) e a última desigualdade vemos que $\bar{c}X^* = Z^V$. Ou seja X^* é solução ótima do agregado $PA^V(g^*)$. Como x^* é sua correspondente solução desagregada de peso fixo e por hipótese ela é ótima para o PL, então g^* é um peso ótimo para o PL.

■

Observação 1.2.1

a) Com a desagregação de peso fixo, o problema PA^V gera uma solução viável e subotimal $\bar{x}$ para PL (ver 1.3).

b) Agregação e desagregação de peso fixo, formam uma restrição adicional do PL: $x^k = X_k g^k$. Para cada k, x^k está restrito ao espaço vetorial unidimensional de múltiplos de g^k.

2. **Desagregação otimal de $\bar{X}$**

Uma solução desagregada melhorada pode ser obtida através da desagregação otimal de $\bar{X}$ (solução ótima do PA^V), resolvendo K subproblemas :

$$DP_k(\bar{X}_k) \left\{ \begin{array}{ll} & max \quad c^k y^k \\ s.a & B^k y^k \leq \bar{X}_k \bar{A}^k \\ & \quad y^k \geq 0. \end{array} \right.$$

Suponha que para cada k =1,...,K, y^k é solução viável de $DP_k(\bar{X}_k)$. Seja y o conjunto de vetores y^k, arrumados segundo a ordem das variáveis em PL. Se para cada k= 1,...,K, $\overline{y}^k$ é solução ótima de $DP_k(\bar{X}_k)$, chamaremos a $\overline{y}$ **solução desagregada otimal**.

Proposição 1.2.8 *Dados PL, PA^V e os $DP_k(\bar{X}_k)$, k=1,...,K, então temos o seguinte:*
a) *A solução desagregada otimal $\bar{y}$ é viável para PL.*
b) *A solução desagregada de peso fixo $\bar{x}$ (restrita a cada sub-problema) é uma solução viável de $DP_k(\bar{X}_k)$, e*

$$Z^V \quad \leq \quad \sum_{k=1}^{K} c^k \bar{y}^k \quad \leq \quad Z^*. \tag{1.8}$$

PROVA
a)

$$A\bar{y} = \sum_{k=1}^{K} B^k \bar{y}^k \leq \sum_{k=1}^{K} \bar{X}_k \bar{A}^k = \bar{A}\bar{X} \leq b.$$

É óbvio que $\bar{y} \geq 0$. Portanto $\bar{y}$ é viável para PL.
b)

$$B^k \bar{x}^k = B^k(\bar{X}_k g^k) = \bar{X}_k(B^k g^k) = \bar{X}_k \bar{A}^k$$

Como $\bar{x}$ é viável para PL (ver 1.2.3) então $\bar{x} \geq 0$, logo $\bar{x}^k \geq 0$. Portanto $\bar{x}$ restrita a cada sub-problema é solução viável de $DP_k(\bar{X}_k)$. Então para $k = 1 \ldots, K$, tem-se

$$c^k \bar{x}^k \leq c^k \bar{y}^k.$$

Por a) $\bar{y}$ é viável para PL, então

$$\sum_{k=1}^{K} c^k \bar{y}^k = c\bar{y} \leq Z^* \tag{1.9}$$

logo pelo Corolário (1.2.4) e a penúltima desigualdade obtem-se

$$Z^V = c\bar{x} = \sum_{k=1}^{K} c^k \bar{x}^k \leq \sum_{k=1}^{K} c^k \bar{y}^k \tag{1.10}$$

de (1.9) e (1.10) segue-se

$$Z^V \leq \sum_{k=1}^{K} c^k \bar{y}^k \leq Z^* \qquad \blacksquare$$

Observação 1.2.2

a) Com a desagregação otimal de $\bar{X}$, os K-sub-problemas $DP_k(\bar{X}_k)$ geram uma solução viável $\bar{y}$ para PL, que é melhor que a solução desagregada de peso fixo $\overline{x}$ (ver 1.8 e Corolário 1.2.4).

b) Só é possível obter a solução desagregada otimal depois de ter resolvido PA^V.

1.2.3 Limitantes a posteriori

Encontraremos limitantes a posteriori para o valor ótimo (Z^*) do problema original PL, depois de formar e resolver o problema PA^V. Com o objetivo de obter uma estimativa para este valor ótimo, é conveniente encontrar limitantes inferiores e superiores de Z^*. Faremos isto baseando-nos na informação de PA^V e dos $DP_k(\bar{X}_k)$. O quadro abaixo indica os limitantes a posteriori para Z^* que obteremos

$$\textbf{Limitantes a posteriori}\begin{cases} \textit{Limitantes Inferiores} \begin{cases} \textit{usando } PA^V \\ \textit{usando } DP_k(\bar{X}_k) \end{cases} \\ \textit{Limitantes Superiores} \begin{cases} \textit{usando } PA^V \begin{cases} \textit{Zipkin} \\ \textit{Mendelsohn} \\ \textit{Taylor} \end{cases} \\ \textit{usando } DP_k(\bar{X}_k) \begin{cases} L^1 \\ L^2 \end{cases} \end{cases} \end{cases}$$

Os limitantes a posteriori: superiores e inferiores, dividem-se em limitantes que utilizam o PA^V e os que utilizam o $DP_K(\bar{X})$. O quadro acima expõe a classificação. Passamos a descreve-los.

Limitantes Inferiores

Limitantes inferiores para Z^* foram encontrados na seção anterior, quando vimos o processo de desagregação. Para cada tipo de desagregação corresponde um limitante inferior para Z^*.

1. **Limitante que usa o PA^V** O valor ótimo (Z^V) do problema PA^V é um limitante inferior para o valor ótimo de PL, porque se $\bar{X}$ é solução ótima de PA^V, então sua correspondente solução $\bar{x}$ desagregada de peso fixo é viável para PL. Isto implica que $c\bar{x} \leq Z^*$ e pela proposição 1.2.3 tem-se $c\bar{x} = \bar{c}\bar{X} = Z^V$. Logo

$$Z^V \leq Z^*.$$

2. **Limitante que usa os $DP_k(\bar{X}_k)$** A soma dos valores ótimos dos K-sub-problemas $DP_k(\bar{X}_k)$ também é um limitante inferior para o valor ótimo de PL, ou seja

$$\sum_{k=1}^{K} c^k \bar{y}^k \quad \leq \quad Z^*,$$

porque a solução desagregada otimal $\bar{y}$ é viável para o problema inicial (ver Proposição 1.2.8). A desigualdade

$$Z^V \quad \leq \quad \sum_{k=1}^{K} c^k \bar{y}^k \quad \leq \quad Z^*$$

obtida anteriormente indica que um melhor limitante inferior é obtido usando desagregação otimal.

Limitantes superiores

Para o cálculo dos limitantes superiores, vamos supor a seguinte hipótese sobre uma solução ótima x^* de PL:

Hipótese 1:

Seja $\sigma' = \{S'_k : k = 1, \ldots, K'\}$ uma partição de $\{1, \ldots, n\}$, não necessariamente igual a partição σ utilizada para construir o problema agregado. Existem $d_1, \ldots . d_n$ números positivos conhecidos e $p_1, \ldots, p_{K'}$ números não negativos conhecidos, tais que

$$\sum_{j \in S'_k} d_j x_j * \leq p_k, \quad k = 1, \ldots, K'. \tag{1.11}$$

Esta hipótese requer restrições em x^*, obviamente sem resolver PL. Estas restrições tanto podem ser extraídas do conjunto das restrições existentes

quanto obtidas manipulando as restrições do mesmo. Em alguns casos pode ser impossível obte-las.

Encontraremos 5 limitantes superiores a posteriori para Z^*. Os 3 primeiros são obtidos através da proposição 1.2.9 e utilizam a solução ótima dual do PA^V. Os 2 últimos utilizam as soluções ótimas duais de cada um dos problemas $DP_k(\bar{X}_k)$. Por isto dividimos os limitantes superiores para Z^* em 2 grupos: os que usam PA^V e os que usam $DP_k(\bar{X}_k)$.

1. **Limitantes que usam o PA^V**
   O conhecimento da solução dual do PA^V nos permite obter uma classe de limitantes superiores para Z^*. Vamos estudar 3 deles: de Zipkin, de Mendelssohn e de Taylor. Para isto, a proposição chave é:

   Proposição 1.2.9 *Se a hipótese 1 é satisfeita por alguma solução ótima x^* de (PL), então*

$$Z^* \leq wb + \varepsilon_a \tag{1.12}$$

   onde

$$\varepsilon_a = \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - wA^j}{d_j} \right]^+ p_k, \tag{1.13}$$

$$[a]^+ = max\{0, a\} \qquad e$$

$$w \geq 0 \text{ é um vetor em } \mathbb{R}^m.$$

   PROVA
   $\forall w \geq 0$

$$\begin{aligned} Z^* = cx^* &\leq cx^* + w(b - Ax^*) \\ &= wb + (c - wA)x^* \\ &= wb + \sum_{j=1}^{n} (c_j - wA^j)x_j* \\ &\leq wb + \sum_{k=1}^{K'} \sum_{j \in S'_k} \left[\frac{c_j - wA^j}{d_j} \right]^+ d_j x_j^* \end{aligned}$$

$$
\begin{aligned}
&\leq wb + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - wA^j}{d_j} \right]^+ \sum_{j \in S'_k} d_j x_j* \\
&\leq wb + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - wA^j}{d_j} \right]^+ p_k \quad (por \quad hip.1) \\
&= wb + \varepsilon_a, \qquad (1.14)
\end{aligned}
$$

logo

$$Z^* \leq wb + \varepsilon_a$$

■

Pela proposição anterior sabemos que para qualquer $w \geq 0 \quad em \quad \mathbb{R}^m$, o valor L definido como

$$L = wb + \varepsilon_a \qquad (1.15)$$

é um limitante superior para Z^*. Os 3 limitantes superiores indicados são obtidos usando valores particulares para w, a saber:

Limitante de Zipkin (L^Z) Zipkin escolhe w como sendo a solução ótima dual ($\bar{U} \geq 0$) do problema agregado, $w = \bar{U}$. Substituindo este valor em (1.15) e denotando o limitante assim obtido por L^Z, temos

$$L^Z = \bar{U}b + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - \bar{U}A^j}{d_j} \right]^+ p_k \qquad (1.16)$$

ou ainda,

$$L^Z = Z^V + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - \bar{U}A^j}{d_j} \right]^+ p_k.$$

Pela proposição 1.2.9 temos que

$$Z^* \leq L^Z = Z^V + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - \bar{U}A^j}{d_j} \right]^+ p_k. \qquad (1.17)$$

Limitante de Mendelssohn (L^M) Mendelssohn ao invés de tomar $w = \bar{U}$ como no caso anterior, considera todos os múltiplos positivos

de $\bar{U}$, ou seja, $w = \theta\bar{U}, com \quad \theta \geq 0$. Substituindo este valor em (1.15) temos um limitante em função de θ, dado por:

$$f(\theta) = \theta\bar{U}b + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - \theta\bar{U}A^j}{d_j}\right]^+ p_k \tag{1.18}$$

para o qual sabemos valer:

$$Z^* \leq f(\theta) \qquad \forall\theta \in \mathbb{R}, \theta \geq 0.$$

O limitante de Mendelssohn é obtido minimizando $f(\theta)$ em relação a θ.

$$Z^* \leq \min_{\theta \in R, \theta \geq 0} f(\theta) = L^M$$

Assim, temos:

$$\begin{aligned} L^M &= \min_{\theta \in R, \theta \geq 0} f(\theta) && (1.19) \\ &= \min_{\theta \in R, \theta \geq 0} \left(\theta\bar{U}b + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - \theta\bar{U}A^j}{d_j}\right]^+ p_k\right) \\ &= \min_{\theta \in R, \theta \geq 0} \left(\theta Z^V + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - \theta\bar{U}A^j}{d_j}\right]^+ p_k\right). && (1.20) \end{aligned}$$

Limitante de Taylor (L^T) Taylor escolhe $w = \bar{U} + \lambda b \geq 0$. Substituindo este valor em (1.15) temos

$$g(\lambda) = (\bar{U} + \lambda b)b + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - (\bar{U} + \lambda b)A^j}{d_j}\right]^+ p_k \tag{1.21}$$

Pela proposição 1.2.9

$$Z^* \leq g(\lambda) \qquad \forall\lambda \in \mathbb{R} \quad tal \quad que \quad \bar{U} + \lambda b \geq 0.$$

Logo

$$Z^* \leq \min_{\lambda \in R, \bar{U} + \lambda b \geq 0} g(\lambda),$$

obtendo assim o limitante de Taylor:

$$
\begin{aligned}
L^T &= \min_{\lambda \in R, \bar{U}+\lambda b \geq 0} g(\lambda) && (1.22)\\
&= \min_{\lambda \in R, \bar{U}+\lambda b \geq 0} \left\{ \bar{U}b + \lambda bb + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - (\bar{U} + \lambda b)A^j}{d_j} \right]^+ p_k \right\} \\
&= Z^V + \min_{\lambda \in R, \bar{U}+\lambda b \geq 0} \left\{ \lambda bb + \sum_{k=1}^{K'} \max_{j \in S'_k} \left[\frac{c_j - (\bar{U} + \lambda b)A^j}{d_j} \right]^+ p_k \right\} && (1.23)
\end{aligned}
$$

Observação 1.2.3

a) Com a finalidade de obter um melhor limitante superior para Z^*, as desigualdades de (1.11), *que aparecem na hipótese 1, deveriam ser o mais justas o possível, pois como a proposição* 1.2.9 *usa tais desigualdades, os limitantes L^Z, L^M, e L^T dependem das mesmas.*

b) Note também que se $\bar{U}$ for solução viável para o dual de PL , então

$$L^Z = Z^V \quad e \quad por \quad (1.17) \quad Z^* \leq Z^V$$

juntando com a desigualdade $Z^V \leq Z^*$ obtida no Corolário 1.2.4 *obtemos*

$$Z^V = Z^*.$$

c) Fazendo uma comparação entre os limitantes aqui apresentados L^Z, L^M e L^T, temos

$$L^M \leq L^Z$$

e

$$L^T \leq L^Z$$

porque de (1.19) *temos*

$$L^M \leq f(1) = L^Z$$

e de (1.22)

$$L^T \leq g(0) = L^Z$$

2. **Limitantes que usam os** $DP_k(\bar{X}_k)$. Zipkin descreve duas formas de gerar o limitante superior de Z^* quando se tem informação das soluções dos problemas duais dos $DP_k(\bar{X}_k)$.

Limitante L^1

Vamos continuar supondo que a hipótese 1 esteja satisfeita para alguma solução ótima x^* de PL, só que com a particularidade de que a partição σ usada para construir o problema agregado seja igual à partição σ', mencionada em (1.11).

Suponha também que (1.11) seja satisfeita com $d_j = 1$ para j = 1,...,n, ou seja, $\sum_{j \in S'_k} x_j{}^* \leq p_k$. Seja U^{k^*} a solução ótima dual de $DP_k(\bar{X}_k)$, para k=1,...,K. Então

$$\begin{aligned}
Z^* \quad = \quad cx^* \quad &\leq cx^* + \sum_{k=1}^{K} (U^{k^*}B^k - c^k)x^{*k} \\
&= cx^* - \sum_{k=1}^{K} c^k x^{*k} + \sum_{k=1}^{K} U^{k^*}B^k x^{*k} \\
&= \sum_{k=1}^{K} U^{k^*} \sum_{j \in S_k} x_j^* A^j \\
&= \sum_{k=1}^{K} \sum_{j \in S_k} x_j^* (U^{k^*} A^j) \\
&\leq \sum_{k=1}^{K} \max_{j \in S_k} (U^{k^*} A^j) \sum_{j \in S_k} x_j^* \\
&\leq \sum_{k=1}^{K} \left[\max_{j \in S_k} (U^{k^*} A^j) \right]^+ \sum_{j \in S_k} x_j^* \\
&\leq \sum_{k=1}^{K} \left[\max_{j \in S_k} (U^{k^*} A^j) \right]^+ p_k,
\end{aligned}$$

logo

$$Z^* \quad \leq \quad \sum_{k=1}^{K} \left[\max_{j \in S_k} (U^{k^*} A^j) \right]^+ p_k \tag{1.24}$$

Denotaremos por L^1 a soma $\sum_{k=1}^{K} \left[\max_{j \in S_k} (U^{k^*} A^j) \right]^+ p_k$.

Limitante L^2

Suponha que existe uma solução ótima x^* de PL que satisfaz a a condição

$$B^k x^{*k} \geq 0, \forall k = 1, \ldots, K, \tag{1.25}$$

então

$$\begin{aligned}
Z^* &= cx^* \leq cx^* + \sum_{k=1}^{K} U^{k*}(b - Ax^*) \\
&= cx^* + \sum_{k=1}^{K} U^{k*}b - \sum_{k=1}^{K} U^{k*}(\sum_{r=1}^{K} B^r x^{*r}) \\
&= cx^* + \sum_{k=1}^{K} U^{k*}b - \sum_{k=1}^{K} U^{k*}B^k x^{*k} - \sum_{k=1}^{K} U^{k*}(\sum_{r=1, r\neq k}^{K} B^r x^{*r}) \\
&\leq cx^* + \sum_{k=1}^{K} U^{k*}b - \sum_{k=1}^{K} c^k x^{*k} - \sum_{k=1}^{K} U^{k*}(\sum_{r=1, r\neq k}^{K} B^r x^{*r}) \\
&\leq \sum_{k=1}^{K} U^{k*}b - \sum_{k=1}^{K} U^{k*}(\sum_{r=1, r\neq k}^{K} B^r x^{*r}) \\
&\leq \sum_{k=1}^{K} U^{k*}b \\
&\leq K \sum_{i=1}^{m} b_i \max_k \quad U_i^{k*}.
\end{aligned}$$

Portanto

$$Z^* \leq K \sum_{i=1}^{m} b_i \max_k \quad U_i^{k*} \tag{1.26}$$

Denotaremos por L^2 o limitante de Z^* obtido aqui, isto é,

$$L^2 = K \sum_{i=1}^{m} b_i \max_k \quad U_i^{k*}.$$

1.2.4 Limitantes a priori

O erro resultante da perda de informação se usado o método de agregação é expresso pelo grau de subotimalidade da solução desagregada $\bar{x}$, quando o

valor da função objetivo do PL em $\bar{x}$ é comparada com o valor ótimo de PL. Ou seja, o erro é dado pela diferença que existe entre o valor ótimo (Z^*) de PL e o valor ótimo (Z^V) do PA^V. Vamos achar um limitante para este erro sem resolver o PA^V, por isso o nome de *limitante a priori*. Continuaremos supondo que a hipótese 1 ésteja satisfeita para alguma solução ótima x^* de PL.

Por 1.17, temos

$$Z^* \leq Z^V + \sum_{k=1}^{K'} \max_{j \in S'_k} \left\{ \left[\frac{c_j - \bar{U} A^j}{d_j} \right]^+ \right\} p_k,$$

que pode ser reescrita como

$$Z^* \leq Z^V + \sum_{k=1}^{K'} \left(\max_{j \in S'_k} \left[\frac{c_j - \bar{U} A^j}{d_j} \right] \right)^+ p_k. \tag{1.27}$$

Tomemos números $\{r_j\}$, $j = 1, \ldots, n$ tais que

$$r_j \geq c_j - \bar{U} A^j, \tag{1.28}$$

e definimos

$$\mathcal{R}^k = \max_{j \in S'_k} \left[\frac{r_j}{d_j} \right], \quad k = 1, \ldots, K', \tag{1.29}$$

De (1.27), (1.28) e (1.29), vem

$$Z^* \leq Z^V + \varepsilon_b, \tag{1.30}$$

onde

$$\varepsilon_b = \sum_{k=1}^{K'} \left[\mathcal{R}^k \right]^+ p_k. \tag{1.31}$$

O objetivo é especificar os r_j em (1.28) sem conhecimento de $\bar{U}$. Caso seja possível encontrá-los, ε_b será um limitante superior a priori de $Z^* - Z^V$. Na próxima proposição veremos uma maneira de obter estes r_j.

Para cada $j \in S_k, k = 1, \ldots, K$, consideremos o seguinte problema:

$$(P_j) \begin{cases} \min_v (c_j - vA^j) \\ s.a \quad v\bar{A}^k \geq \bar{c}^k \\ \qquad v \geq 0 \end{cases}$$

Proposição 1.2.10 *Se cada um dos problemas P_j tiver solução ótima r_j, então existe um limitante superior a priori para $Z^* - Z^V$.*

PROVA

Primeiramente temos que cada problema P_j é viável. Como $\bar{U}$ é viável para o dual de PA^V, então

$$\bar{U}\bar{A} \geq \bar{c} \quad e \quad \bar{U} \geq 0.$$

Assim, em particular $\bar{U}$ é viável para cada um dos problemas P_j. Então, minimizando sobre todos os pontos viáveis de P_j temos

$$min(vA^j) \leq \bar{U}A^j$$

logo

$$c_j - min(vA^j) \geq c_j - \bar{U}A^j,$$

fazendo

$$r_j = c_j - min(vA^j) \qquad temos$$
$$r_j \geq c_j - \bar{U}A^j.$$

Encontramos assim r_j que satisfaz (1.28) e que não depende da solução $\bar{U}$ de DA^V. Logo, de (1.31)

$$\varepsilon_b = \sum_{k=1}^{K'} \left(\max_{j \in S'_k} \left[\frac{c_j - min(vA^j)}{d_j} \right] \right)^+ p_k \tag{1.32}$$

é um limitante a priori para $Z^* - Z^V$.

■

A proposição a seguir mostra um caso particular no qual os problemas P_j têm solução ótima:

Proposição 1.2.11 *Se $A \geq 0$ e $c \geq 0$ então cada problema P_j tem solução ótima. Neste caso*

$$\varepsilon_b = \sum_{k=1}^{K'} \left(\max_{i,j} \left[\left(c_j - \bar{c}_k \frac{A_{ij}}{\bar{A}_{ik}} \right) / d_j; \quad j \in S'_k, \quad \bar{A}_{ik} > 0 \right] \right)^+ p_k \tag{1.33}$$

PROVA
Para $j \in S_k$ seja

$$\frac{A_{rj}}{\bar{A}_{rk}} = \min_i \left\{ \frac{A_{ij}}{\bar{A}_{ik}} : \quad \bar{A}_{ik} > 0 \right\}$$

então

$$A_{ij} \geq \bar{A}_{ik} \frac{A_{rj}}{\bar{A}_{rk}} \quad \forall i \in \{1, \ldots, m\}$$

Cada P_j é viável porque a solução dual do agregado satisfaz suas restrições. $\forall v$ solução viável de P_j, tem-se

$$\begin{aligned} vA^j = \sum_{i=1}^{m} v_i A_{ij} &\geq \sum_{i=1}^{m} v_i \bar{A}_{ik} \left\{ \frac{A_{rj}}{\bar{A}_{rk}} \right\} \\ &= \left\{ \frac{A_{rj}}{\bar{A}_{rk}} \right\} \sum_{i=1}^{m} v_i \bar{A}_{ik} \\ &= \left\{ \frac{A_{rj}}{\bar{A}_{rk}} \right\} v\bar{A}^k \\ &\geq \bar{c}_k \min \left\{ \frac{A_{ij}}{\bar{A}_{ik}} \quad : \bar{A}_{ik} > 0 \right\}. \end{aligned} \tag{1.34}$$

Portanto para qualquer v pertencente ao conjunto de soluções viáveis de P_j, tem-se

$$\min_v vA^j \geq \bar{c}_k \min \left\{ \frac{A_{ij}}{\bar{A}_{ik}} : \quad \bar{A}_{ik} \geq 0 \right\}. \tag{1.35}$$

Cada P_j é viável e por (1.35) P_j é limitado. Portanto cada P_j tem solução ótima. Vamos mostrar que vale a igualdade em (1.35).
a) Se $\bar{c}_k = 0$
Vamos supor que em (1.35)

$$\min_v vA^j > \bar{c}_k min \left\{ \frac{A_{ij}}{\bar{A}_{ik}} : \quad \bar{A}_{ik} \geq 0 \right\}. \tag{1.36}$$

Neste caso é claro que $v = 0$ é viável para P_j . Pela condição $A \geq 0$, temos que 0 minimiza vA^j sobre o conjunto de soluções viáveis de P_j. Então em (1.36) teremos a contradição

$$0 > 0.$$

Portanto

$$\min_v vA^j = \bar{c}_k min\left\{\frac{A_{ij}}{\bar{A}_{ik}} : \quad \bar{A}_{ik} \geq 0\right\}.$$

b) Se $\quad \bar{c}_k > 0$

Sabe-se que cada P_j tem solução ótima e que a cada $\bar{A}_{ik} > 0$ está associada a solução básica

$$\left(0, \ldots, \frac{\bar{c}_k}{\bar{A}_{ik}}, \ldots, 0\right).$$

Assim, neste caso a solução ótima de (P_j) é

$$\left(0, \ldots, \frac{\bar{c}_k}{\bar{A}_{sk}}, \ldots, 0\right).$$

sempre e quando satisfaça que

$$\left(\frac{\bar{c}_k}{\bar{A}_{ik}}\right) A_{ij} \geq \left(\frac{\bar{c}_k}{\bar{A}_{sk}}\right) A_{sj}$$

$\forall i \in \{1. \ldots, m\}$, tal que $\bar{A}_{ik} > 0$. Pelo fato de ser $\bar{c}_k > 0$, isto implica que

$$\frac{A_{ij}}{\bar{A}_{ik}} \geq \frac{A_{sj}}{\bar{A}_{sk}} \qquad \forall i \in \{1, \ldots, m\}, \quad tal \quad que \quad \overline{A}_{ik} > 0.$$

Logo, desta desigualdade, tem-se

$$\frac{A_{sj}}{\bar{A}_{sk}} = \min_i \left\{\frac{A_{ij}}{\bar{A}_{ik}} \quad : \bar{A}_{ik} > 0\right\} \tag{1.37}$$

Portanto

$$\min_v vA^j = \left\{\frac{\bar{c}_k}{\bar{A}_{sk}}\right\} A_{sj} = \bar{c}_k \min_i \left\{\frac{A_{ij}}{\bar{A}_{ik}} \quad : \bar{A}_{ik} > 0\right\} \tag{1.38}$$

Valendo a igualdade em (1.35). Substituindo (1.38) em (1.32) obtemos

$$\varepsilon_b \quad = \quad \sum_{k=1}^{K'} \left(max_{j \in S'_k} \left[\left(c_j - \bar{c}_k \min_i \left\{ \frac{A_{ij}}{\bar{A}_{ik}} \quad : \bar{A}_{ik} > 0 \right\} \right) \Big/ d_j \right] \right)^+ p_k$$

$$= \sum_{k=1}^{K'} \left(max_{j \in S'_k} \left[\left(c_j + \bar{c}_k \max_i \left\{ \frac{-A_{ij}}{\bar{A}_{ik}} \quad : \bar{A}_{ik} > 0 \right\} \right) / d_j \right] \right)^+ p_k$$

$$= \sum_{k=1}^{K'} \left(\max_{i,j} \left[\left(c_j - \bar{c}_k \frac{A_{ij}}{\bar{A}_{ik}} \right) / d_j : \bar{A}_{ik} > 0, j \in S'_k \right] \right)^+ p_k$$

■

O Lema de Farkas, que enunciamos a seguir, será utilizado na demonstração da próxima proposição, que fornece uma condição necessária e suficiente para P_j ter solução ótima.

Lema de Farkas
Dados os sistemas lineares

$$S_1 \quad \begin{cases} Ex & \leq 0 \\ e^t x & > 0 \end{cases}$$

e

$$S_2 \quad \begin{cases} E^t y & = e \\ y & \geq 0 \end{cases}$$

então, ou S_1 ou S_2 tem solução, mas nunca ambos.
PROVA (ver Dorfman[4], pag.504).

Proposição 1.2.12 *Dado $j \in S_k$. P_j tem solução ótima se e só se existe $\lambda > 0$ tal que*

$$\lambda \bar{A}^k \leq A^j$$

PROVA
Vamos escrever o problema P_j do seguinte modo

$$(P_j) \begin{cases} \min_v (c_j - vA^j) \equiv \max_v (c_j + v(-A^j)) \\ s.a \quad v\bar{A}^k \geq \bar{c}^k \\ v \geq 0. \end{cases}$$

Suponhamos que P_j tenha solução ótima. Então o seguinte sistema linear

$$(S_1) \begin{cases} -u^T \bar{A}^k & \leq 0 \\ -u & \leq 0 \\ u^T(-A^j) & > 0 \end{cases}$$

não tem solução, porque se existisse uma solução u, então a sequência

$$\{\bar{U} + nu\}_{n \in N}$$

formaria uma familia de soluções viáveis de P_j e o valor da função objetivo

$$(\bar{U} + nu)(-\bar{A}^j)$$

tenderia para infinito quando n crescesse, tornando P_j ilimitado e contradizendo a hipótese de P_j ter solução ótima. Portanto S_1 não tem solução. Podemos ver S_1 da seguinte forma

$$(S_1) \begin{cases} -(\bar{A}^k)^T u & \leq 0 \\ -Iu & \leq 0 \\ -(A^j)^T u & > 0 \end{cases}$$

Pelo *Lema de Farkas*, o seguinte sistema tem solução

$$\begin{aligned} \begin{bmatrix} -\bar{A}^k & -I \end{bmatrix} y &= -A^j \\ y &\geq 0. \end{aligned} \tag{1.39}$$

Tomando

$$y^T = [\lambda \quad y_1{}^T] \quad ; \quad \lambda \geq 0 \quad e \quad y_1 \geq 0$$

então temos em (1.39)

$$-\lambda \bar{A}^k - y_1 = -A^j \quad ; \quad \lambda \geq 0 \quad e \quad y_1 \geq 0$$

Logo, temos a desigualdade desejada

$$\lambda \bar{A}^k \leq A^j.$$

Reciprocamente, suponhamos que existe $\lambda \geq 0$ tal que

$$\lambda \bar{A}^k \leq A^j$$

A solução ótima de DA^V satisfaz as restrições de (P_j). Portanto (P_j) é viável. Tomando uma solução viável v de (P_j) temos

$$v\bar{A}^k \geq \bar{c}^k \quad e \quad v \geq 0$$

logo, por hipótese

$$vA^j \geq v(\lambda \bar{A}^k) = \lambda(v\bar{A}^k) \geq \lambda \bar{c}^k$$

Desta desigualdade temos

$$min \quad vA^j \geq \lambda \bar{c}^k$$

Portanto o problema P_j tem valor ótimo.

■

1.3 Agregação de restrições e variáveis

Vamos agora a agregar ao mesmo tempo, restrições e variaveis do problema original. Substituiremos cada conjunto de linhas e colunas por uma combinação convexa dos membros do conjunto. A análise de agregação de variáveis feita na seção anterior era simplificado pelo fato de uma solução viável do problema original PL poder ser obtida a partir da solução ótima do problema agregado. Aqui a solução agregada, quando desagregada, pode até mesmo não ser viável para o problema original. Suponha que temos o seguinte problema linear

$$(PL)\left\{\begin{array}{rrr} Z^* = & & max \quad cx \\ & s.a & Ax \leq b \\ & & x \geq 0 \end{array}\right.$$

onde como sempre $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$. Vamos supor que PL tenha solução ótima.

Precisamos de 2 partições e 2 pesos associados aos índices das linhas e das colunas da matriz A.

Seja $\sigma = \{S_k : k = 1, \ldots, K\}$ uma partição (ver definição 1.2.1) de {1,...,n} e $\rho = \{R_l : l = 1, \ldots, L\}$ uma partição de {1,...,m}, onde n_k é a cardinalidade de S_k e m_l é a cardinalidade de R_l. Como fizemos anteriormente, vamos primeiro definir os vetores peso:
Sejam

- g^k : vetor (coluna) em $\mathbb{R}^{n_k}$ de componentes não-negativas cuja soma é igual a 1.
- f^l : vetor (linha) em $\mathbb{R}^{m_l}$ de componentes não negativas cuja soma é igual a 1.
- g : sequência dos g^k, k=1,...,K.
- f : sequência dos f^l, l=1,...,L.

Os vetores g e f serão chamados de **pesos**.
Sejam agora

$$c^k = (c_j)_{j \in S_k} \qquad b^l = (b_i)_{i \in R_l} \qquad B_l^k = (A_{ij})_{i \in R_l, j \in S_k} \tag{1.40}$$

Para construir o problema duplamente agregado PDA definimos

$$\widetilde{c}_k = c^k g^k \qquad \widetilde{b}_l = f^l b^l \qquad \widetilde{A}_{lk} = f^l B_l^k g^k \tag{1.41}$$

se

$$\widetilde{c} = (\widetilde{c}_k) \quad , \quad \widetilde{b} = (\widetilde{b}_l) \quad e \quad \widetilde{A} = \widetilde{A}_{lk} \tag{1.42}$$

O problema agregado PDA se torna:

$$(PDA) \begin{cases} Z^D = & max \quad \widetilde{c}X \\ \quad s.a & \widetilde{A}X \leq \widetilde{b} \\ & X \geq 0 \end{cases}$$

onde X é um vetor com K componentes. O dual associado é dado por

$$(DDA) \begin{cases} Z^D = & min \quad U\widetilde{b} \\ \quad s.a & U\widetilde{A} \geq \widetilde{c} \\ & U \geq 0 \end{cases}$$

O PDA é deste modo determinado pelo PL, as partições σ, e ρ e os pesos g ef. Vamos assumir que PDA tenha solução ótima finita e vamos denotar por $\widetilde{X}$ e $\widetilde{U}$ as soluções ótimas do primal e do dual respectivamente.

Desagregação de peso fixo

Para qualquer par (X, U) viável para PDA e seu dual, sejam

$$x^k = X_k g^k, \quad k = 1, \ldots, K \tag{1.43}$$

$$u^l = U_l f^l, \quad l = 1, \ldots, L. \tag{1.44}$$

e (x, u) o par de vetores obtidos arrumando as componentes de x^k e u^l, segundo a ordem das colunas e linhas no problema inicial respectivamente. Como anteriormente, chamaremos a (x, u) **soluções desagregadas de peso fixo** para PL obtidas a partir de (X, U).

Observe que o problema que agrega só variáveis (restrições), é um caso particular de PDA, obtido quando se considera a partição $\rho = \{\{1\}, \ldots, \{m\}\}$ ($\sigma = \{\{1\}, \ldots, \{n\}\}$). Para o caso de variáveis (restrições), as notações $\widetilde{A}$ e PDA serão substituídas por $\overline{A}$ e PA^V ($\widehat{A}$ e PA^R) respectivamente.

Na proposição a seguir demonstraremos que o valor da função objetivo do PL na solução desagregada $\widetilde{x}$ é igual ao valor ótimo do problema agregado PDA. Também relacionaremos os valores Z^* e Z^D com os valores ótimos dos problemas particulares mencionados no parágrafo anterior.

Proposição 1.3.1 *Sejam Z^R e Z^V os valores ótimos para PA^R e PA^V respectivamente e $(\widetilde{x}, \widetilde{u})$ as soluções desagregadas de peso fixo para PL obtidas a partir das soluções ótimas $(\widetilde{X}, \widetilde{U})$ de PDA e de seu dual. Então*
a) $c\widetilde{x} = \widetilde{u}b = Z^D$
b) $Z^V \leq Z^* \leq Z^R$
c) $Z^V \leq Z^D \leq Z^R$

PROVA
a)

$$c\widetilde{x} = \sum_{k=1}^{K} c^k \widetilde{x}^k = \sum_{k=1}^{K} c^k \widetilde{X}_k g^k = \sum_{k=1}^{K} \widetilde{X}_k c^k g^k \sum_{k=1}^{K} \widetilde{X}_k \widetilde{c}_k = \widetilde{c}\widetilde{X} = Z^D$$

$$\widetilde{u}b = \sum_{l=1}^{L} \widetilde{u}^l b^l = \sum_{l=1}^{L} \widetilde{U}_l f^l b^l = \sum_{l=1}^{L} \widetilde{U}_l \widetilde{b}_l = \widetilde{U}\widetilde{b} = Z^D$$

b) Pelo Corolário 1.2.4 sabe-se que

$$Z^V \leq Z^* \tag{1.45}$$

Por outro lado, se $\triangle$ é o conjunto de soluções viáveis do PL e $\triangle^R$ é o conjunto de soluções viáveis de PA^R, então é claro que PA^R é um problema relaxado de PL, ou seja

$$\triangle \subseteq \triangle^R \tag{1.46}$$

porque se $y \in \triangle$, y satisfaz todas as restrições de PL, portanto y também satisfaz todas as restrições que são obtidas através de combinações convexas de restrições de PL, isto é $y \in \triangle^R$. Por (1.46) temos

$$\max_{x \in \triangle} c^t x \leq \max_{x \in \triangle^R} c^t x$$

logo

$$Z^* \leq Z^R.$$

De (1.45) e a desigualdade anterior temos

$$Z^V \leq Z^* \leq Z^R$$

c) Observe que o PDA pode ser visto de duas maneiras.
1- Como um problema obtido agregando só restrições do PA^V ou
2- Como um problema obtido agregando só variáveis do PA^R.
Em b) foi estabelecida a relação existente entre o valor ótimo Z^* do problema inicial PL e os valores ótimos dos problemas obtidos agregando só variáveis Z^V ou só restrições Z^R no PL. Se em b) trabalhamos com PA^V no lugar de PL, por 1) temos

$$Z^V \leq Z^D.$$

Analogamente se em b) trabalhamos com PA^R no lugar de PL, por 2) temos

$$Z^D \leq Z^R.$$

Logo, das últimas desigualdades temos

$$Z^V \leq Z^D \leq Z^R.$$

■

Infelizmente não é possível obter uma relação de ordem entre Z^D e Z^*. As soluções desagregadas de peso fixo $(\widetilde{x}, \widetilde{u})$ podem até mesmo ser inviáveis para PL e seu dual, excluindo os casos em que só variáveis ou restrições são agregadas. Quando se trata de agregação de variáveis, já foi demonstrado (prop 1.2.3) que a solução desagregada de peso fixo $\widetilde{x}$ é viável para PL. Analogamente em agregação de restrições a solução desagregada de peso fixo $\widetilde{u}$ é viável para o dual de PL.

Exemplo

Daremos um exemplo, onde as soluções desagregadas de peso fixo $(\tilde{x}, \tilde{u})$ não são viáveis para PL e seu dual, respectivamente. Seja o PL

maximize 100 x1 + 600 x2 + 1200 x3 + 2400 x4 + 500 x5 + 2000 x6
Sujeito a:
c1: 8 x1 + 12 x2 + 13 x3 + 64 x4 + 22 x5 + 41 x6 $\leq$ 10
c2: 3 x1 + 6 x2 + 4 x3 + 18 x4 + 6 x5 + 4 x6 $\leq$ 20
c3: 5 x1 + 10 x2 + 8 x3 + 32 x4 + 6 x5 + 12 x6 $\leq$ 36
c4: 5 x1 + 13 x2 + 8 x3 + 42 x4 + 6 x5 + 20 x6 $\leq$ 12

c5: 3 x1 + 2 x2 + 4 x3 + 8 x4 + 8 x5 + 4 x6 $\leq$ 24
c6: 3 x1 + 4 x2 + 8 x5 $\leq$ 18
$x_i \geq 0, i = 1, \ldots, 6$

Sejam

$$\sigma = \{\{1,2,3\},\{4,5,6\}\} = \rho$$

Os pesos

$$g^1 = (1/3, 1/3, 1/3) = f^1$$

e

$$g^2 = (1/3, 1/3, 1/3) = f^2$$

O problema PDA obtido é

maximize 1900/3 X1 + 4900/3 X2
Sujeito a:
c1: 69/9 X1 + 205/9 X2 $\leq$ 22
c2: 42/9 X1 + 96/9 X2 $\leq$ 18
$X_1 \geq 0$, $X_2 \geq 0$

As soluções ótimas primal e dual de PDA são

$$X = (2.869565, 0) \quad e \quad U = (82.608257, 0)$$

As soluções desagregadas de peso fixo obtidas a partir de (X,U) são

$$\tilde{x} = (1/3(2.869565), 1/3(2.869565), 1/3(2.869565), 0, 0, 0)$$

e

$$\tilde{u} = (1/3(82.608257), 1/3(82.608257), 1/3(82.608257), 0, 0, 0)$$

Observemos que a solução desagregada $\tilde{x}$ não é viável para o PL, porque não satisfaz a restrição 4. A solução desagregada $\tilde{u}$ não é viável para o dual do PL, porque não satisfaz a restrição dual associada a colunn 3.

Limitantes para o erro de agregação

Vamos agora encontrar limitantes para Z^*. Começaremos com limitantes que usam as soluções desagregadas de peso fixo ($\tilde{x}$, $\tilde{u}$). Como feito anteriormente, vamos chamar estes limitantes de Limitantes a posteriori, e precisaremos que por hipótese, as soluções ótimas do primal original e de seu dual

obedeçam a um conjunto adicional de restrições, a saber:

Hipótese 2:

Seja $\sigma' = \{S'_k : k = 1, \ldots, K'\}$ *uma partição de* $\{1, \ldots, n\}$, $d_1, \ldots, d_n$ *números positivos conhecidos e* $p_1, \ldots, p_{K'}$ *números não-negativos conhecidos, tais que para alguma solução ótima* x^* *do PL*

$$\sum_{j \in S'_k} d_j x_j^* \leq p_k \quad k = 1, \ldots, K' \tag{1.47}$$

Seja $\rho = \{R'_l : l = 1, \ldots, L'\}$ *uma partição de* $\{1, \ldots, m\}$, $e_1, \ldots, e_m$ *números positivos conhecidos e* $q_1, \ldots, q_{L'}$ *números não negativos conhecidos, tais que para alguma solução ótima* u^* *do dual de PL*

$$\sum_{i \in R'_l} e_i u_i^* \leq q_l \quad l = 1, \ldots, L'. \tag{1.48}$$

Frequentemente estas relações ((1.47) e (1.48)) podem ser derivadas diretamente das restrições do PL e seu dual. As partições σ' e ρ' (não são necessariamente as mesmas usadas para formar o PDA) podem ser escolhidas tomando vantagem da estrutura do PL, com o objetivo de obter as desigualdades requeridas.

Proposição 1.3.2 *Se a hipótese 2 for satisfeita para algum par* (x^*, u^*), *onde* x^* *é uma solução ótima de PL e* u^* *é solução ótima do dual de PL, então*

$$Z^D - \varepsilon_a^- \leq Z^* \leq Z^D + \varepsilon_a^+, \qquad \textit{onde}$$

$$\varepsilon_a^+ = \sum_{k=1}^{K'} \left\{ \max_{j \in S'_k} \left[\frac{c_j - \widetilde{u} A^j}{d_j} \right] \right\}^+ p_k \qquad e$$

$$\varepsilon_a^- = \sum_{l=1}^{L'} \left\{ \max_{i \in R'_l} \left[\frac{A_i \widetilde{x} - b_i}{e_i} \right] \right\}^+ q_l.$$

PROVA

$$Z^* = u^* b \geq u^* b + (c - u^* A)\widetilde{x} = c\widetilde{x} - u^*(A\widetilde{x} - b).$$

Vamos agora desenvolver a última parcela da expressão acima e utilizar (1.48). Assim temos

$$
\begin{aligned}
u^*(A\widetilde{x}-b) &= \sum_{i=1}^{m} u_i^*(A_i\widetilde{x}-b_i) \\
&\leq \sum_{l=1}^{L'}\sum_{i\in R_l'}\left[\frac{A_i\widetilde{x}-b_i}{e_i}\right]u_i^*e_i \\
&\leq \sum_{l=1}^{L'}\max_{i\in R_l'}\left[\frac{A_i\widetilde{x}-b_i}{e_i}\right]\sum_{i\in R_l'}u_i^*e_i \\
&\leq \sum_{l=1}^{L'}\left\{\max_{i\in R_l'}\left[\frac{A_i\widetilde{x}-b_i}{e_i}\right]\right\}^+\sum_{i\in R_l'}u_i^*e_i \\
&\leq \sum_{l=1}^{L'}\left\{\max_{i\in R_l'}\left[\frac{A_i\widetilde{x}-b_i}{e_i}\right]\right\}^+ q_l = \varepsilon_a^-
\end{aligned}
$$

logo

$$
Z^* = u^*b \geq c\widetilde{x} - u^*(A\widetilde{x}-b) \geq c\widetilde{x} - \varepsilon_a^- = Z^D - \varepsilon_a^-
$$

ou seja

$$
Z^* \geq Z^D - \varepsilon_a^- \tag{1.49}
$$

Analogamente demonstraremos que $Z^* \leq Z^D + \varepsilon_a^+$.

$$
Z^* = cx^* \leq cx^* + \widetilde{u}(b - Ax^*) = \widetilde{u}b + (c - \widetilde{u}A)x^*
$$

Como

$$
\begin{aligned}
(c-\widetilde{u}A)x^* &= \sum_{j=1}^{n}(c_j - \widetilde{u}A^j)x_j^* \\
&\leq \sum_{k=1}^{K'}\sum_{j\in S_k'}\left[\frac{c_j-\widetilde{u}A^j}{d_j}\right]d_jx_j^* \\
&\leq \sum_{k=1}^{K'}\max_{j\in S_k'}\left[\frac{c_j-\widetilde{u}A^j}{d_j}\right]\sum_{j\in S_k'}d_jx_j^*
\end{aligned}
$$

$$\leq \sum_{k=1}^{K'} \left\{ \max_{j \in S'_k} \left[\frac{c_j - \widetilde{u}A^j}{d_j} \right] \right\}^+ \sum_{j \in S'_k} d_j x_j^*$$

$$\leq \sum_{k=1}^{K'} \left\{ \max_{j \in S'_k} \left[\frac{c_j - \widetilde{u}A^j}{d_j} \right] \right\}^+ p_k = \varepsilon_a^+,$$

temos

$$Z^* \leq \widetilde{u}b + (c - \widetilde{u}A)x^* \leq \widetilde{u}b + \varepsilon_a^+ = Z^D + \varepsilon_a^+$$

logo

$$Z^* \leq Z^D + \varepsilon_a^+. \tag{1.50}$$

Unindo (1.49) e (1.50) temos

$$Z^D - \varepsilon_a^- \leq Z^* \leq Z^D + \varepsilon_a^+.$$

■

Limitantes a priori

A partir da proposição anterior Zipkin deriva um limitante a priori superior e inferior para o erro $Z^* - Z^D$. Continuaremos supondo que a hipótese 2 seja satisfeita por algum par de soluções ótimas (x^*, u^*) do PL e de seu dual, respectivamente. Tomemos números r_j, j=1,...,n tais que

$$r_j \geq c_j - \widetilde{u}A^j \tag{1.51}$$

e números t_i, i=1,...,m tais que

$$t_i \geq A_i\widetilde{x} - b_i. \tag{1.52}$$

onde $(\widetilde{x}, \widetilde{u})$ é solução desagregada de peso fixo (ver (1.43) e (1.44)) para PL e DL, obtida a partir de $(\widetilde{X}, \widetilde{U})$. Definimos

$$\mathcal{R}^k = max_{j \in S'_k} \left[\frac{r_j}{d_j} \right] \qquad k = 1, \ldots, K' \tag{1.53}$$

e

$$\mathcal{T}^l = max_{i \in R'_l} \left[\frac{t_i}{e_i} \right] \qquad l = 1, \ldots, L'. \tag{1.54}$$

Pela proposição anterior e utilizando (1.51 - 1.54) concluímos que

$$Z^D - \varepsilon_b^- \leq Z^* \leq Z^D + \varepsilon_b^+ \tag{1.55}$$

onde

$$\varepsilon_b^+ = \sum_{k=1}^{K'} \left[\mathcal{R}^k\right]^+ p_k \qquad e$$

$$\varepsilon_b^- = \sum_{l=1}^{L'} \left[\mathcal{T}^l\right]^+ q_l.$$

O objetivo é entretanto especificar os $\{r_j\}$ e $\{t_i\}$ de (1.51) e (1.52) sem conhecimento da solução desagregada $(\widetilde{x}, \widetilde{u}\ )$. Se for possível encontra-los, $-\varepsilon_b^-$ e ε_b^+ serão limitantes a priori inferior e superior respectivamente para $Z^* - \widetilde{Z}$. Para cada $j \in S_k$, k=1,...,K considere o seguinte problema

$$(P_j) \begin{cases} & \min_U (c_j - U\widehat{A}^j) \\ s.a & U\widetilde{A}^k \geq \widetilde{c}_k \\ & U \geq 0 \end{cases}$$

e para cada $i \in R_l$, l=1,...,L, considere este outro problema

$$(Q_i) \begin{cases} & \max_X (-b_i + \overline{A}_i X) \\ s.a & \widetilde{A}_l X \leq \widetilde{b}_l \\ & X \geq 0 \end{cases}$$

Observe que se o PDA tem solução ótima, então todos os problemas P_j e Q_i são viáveis porque se $\widetilde{X}$ e $\widetilde{U}$ são soluções ótimas de PDA e seu dual respectivamente, então

$$\widetilde{U}\widetilde{A} \geq \widetilde{c} \qquad \widetilde{U} \geq 0 \qquad e$$

$$\widetilde{A}\widetilde{X} \leq \widetilde{b} \qquad \widetilde{X} \geq 0.$$

Proposição 1.3.3 *Se cada um dos problemas P_j e Q_i tem solução ótima, então $-\varepsilon_b^-$ e ε_b^+ são respectivamente limitantes a priori, inferior e superior para $Z^* - Z^D$.*

PROVA

Usaremos as seguintes igualdades

$$\widetilde{u}A^j = \widetilde{U}\widehat{A}^j, \qquad j = 1, \ldots, n \tag{1.56}$$

$$A_i\widetilde{x} = \overline{A}_i\widetilde{X}, \qquad i = 1, \ldots, m. \tag{1.57}$$

Como $\widetilde{U}$ é viável para P_j , então por (1.57)

$$minU\widehat{A}^j \leq \widetilde{U}\widehat{A}^j = \widetilde{u}A^j$$

logo

$$c_j - minU\widehat{A}^j \geq c_j - \widetilde{u}A^j$$

fazendo

$$r_j = c_j - minU\widehat{A}^j \tag{1.58}$$

temos

$$r_j \geq c_j - \widetilde{u}A^j.$$

Analogamente como $\widetilde{X}$ é viável para Q_i, então por (1.58)

$$A_i\widetilde{x} = \overline{A}_i\widetilde{X} \leq max\overline{A}_iX$$

logo

$$-b_i + A_i\widetilde{x} \leq -b_i + max\overline{A}_iX$$

fazendo

$$t_i = -b_i + max\overline{A}_iX \tag{1.59}$$

temos

$$t_i \geq -b_i + A_i\widetilde{x}$$

Por hipótese P_j e Q_i tem solução ótima finita, então os (r_j) e (t_i) são finitos e não dependem de $\widetilde{U}$ e $\widetilde{X}$. Portanto $-\varepsilon_b^-$ e ε_b^+ são limitantes a priori inferior e superior respectivamente para $Z^* - Z^D$.

■

Capítulo 2

Agregação Iterativa de Variáveis

2.1 Introdução

Se os limitantes superior e inferior para o valor ótimo de um PL, obtidos resolvendo um problema PA^V não estão dentro de uma faixa razoável, então devemos pensar num método que nos permita mudar iterativamente o o peso para formar outro problema agregado que nos forneça limitantes mais próximos ao valor ótimo do PL. É este o objetivo quando se usa agregação iterativa. O ideal seria que a seqüência de pesos convergisse para o peso ótimo (seção 1.2), mas este ainda é um problema em aberto, como afirmam Rogers, D.F., Plante, R.D., Wong, R.T., e Evans, J.R. [20].

Nas seções 2 e 3 apresentaremos um método teórico de agregação iterativa proposto por Litvinchev[16], que consiste numa aplicação direta do método do gradiente condicional. É teórico, porque se quisermos utiliza-lo seria impossível garantir as hipóteses necessárias para sua aplicabilidade.

Na seção 2, estudamos o problema auxiliar 1, equivalente ao problema PL. Damos os detalhes que justificam a diferenciabilidade da sua função objetivo quando as condições da hipótese 4 são satisfeitas. Diferentemente de Litvinchev, vemos que a função objetivo do problema auxiliar não é continuamente diferenciável em todo G, senão num subconjunto que denotamos por $\widetilde{G}$. Na seção 3 aplicamos o método do gradiente condicional ao problema auxiliar 2, obtido a partir do problema auxiliar 1, substituindo G por outro conjunto convexo, fechado e compacto que denotaremos por G_ϵ.

Jörnstein, K., Leisten, R. e Storoy, S.[12] guiados por este método teórico, considerando o gradiente do Lagrangeano associado ao problema agregado, propõem uma estratégia de agregação iterativa, utilizando uma atualização de peso como no o método do sub-gradiente. Isto será apresentado na seção 4. Implementaremos esta estratégia e a aplicaremos em problemas "*multi-knapsack*" contínuos:

$$(\mathrm{PL})\left\{\begin{array}{rrr} Z^* = & & max \quad cx \\ & s.a & Ax \leq b \\ & & x \leq \mathbf{1_n} \\ & & x \geq 0 \end{array}\right.$$

onde $\mathbf{1_n}$ é um vetor em $\mathbb{R}^n$ com todas as componentes iguais a 1. Consideramos b não negativo, para não correr o risco de ter problema agregado inviável em alguma das iterações.

2.2 Problema auxiliar 1

Apresentaremos nesta e na próxima seção , um procedimento de agregação iterativa para encontrar soluções aproximadas de problemas de programação linear da forma

$$(\mathrm{PL})\left\{\begin{array}{rrr} Z^* = & & max \quad cx \\ & s.a & Ax \leq b \\ & & x \geq 0 \end{array}\right.$$

onde A é uma matriz em $\mathbb{R}^{m\times n}$. Assumimos que PL tenha solução ótima. Tomemos uma partição $\delta = \{S_1, \ldots, S_K\}$ dos índices das colunas $\{1, \ldots, n\}$, um peso g e formemos o problema agregado como foi feito na seção 2 do capítulo I. Assim obtemos

$$(\mathrm{PA}^V(g))\left\{\begin{array}{rrr} \bar{Z} = & & max \quad \bar{c}X \\ & s.a & \bar{A}X \leq b \\ & & X \geq 0 \end{array}\right.$$

e seu dual

$$(\mathrm{DA}^V(g))\left\{\begin{array}{rrr} \bar{Z} = & & min \quad Ub \\ & s.a & U\bar{A} \geq \bar{c} \\ & & U \geq 0 \end{array}\right.$$

Vamos supor que para qualquer escolha de peso g, o problema agregado $PA^V(g)$ seja viável. Nosso procedimento de agregação iterativa precisará de outro problema, relacionado com o PL inicial, que chamaremos de problema auxiliar 1. Antes de enuncia-lo, mencionaremos a notação a ser usada:

- Denotemos por P o conjunto de soluções viáveis do problema inicial (PL), ou seja

$$P = \{x \in \mathbb{R}^n : Ax \leq b, \quad x \geq 0\}.$$

- G o conjunto de pesos que podem ser utilizados para formar o problema agregado PA^V. Isto é

$$G = \{g \in \mathbb{R}^n / g_j \geq 0, \quad \sum_{j \in S_k} g_j = 1, \quad k = 1, \ldots, K\} \tag{2.1}$$

Associado a cada $g \in G$, denotaremos:

- $\mathcal{S}(g)$: Conjunto de soluções viáveis do problema agregado $PA^V(g)$.
- $\mathcal{P}_{oti}(g)$: Conjunto de soluções ótimas para $PA^V(g)$.
- $\mathcal{D}_{oti}(g)$: Conjunto de soluções ótimas para $DA^V(g)$.
- $\theta(g)$: valor ótimo de $PA^V(g)$.

O **Problema Auxiliar 1** que mencionamos é:

$$\begin{aligned} Z^* = max \quad & \theta(g) \\ s.a \quad & g \in G \end{aligned}$$

este problema tem como conjunto de soluções viáveis um produto cartesiano de simplexos.

Sabemos que $\forall g \in G$ o valor ótimo do problema agregado $PA^V(g)$ (denotado por $\theta(g)$) satisfaz a condição

$$\theta(g) \leq Z^* \tag{2.2}$$

e pela proposição 1.2.7 existe um peso g^* para o qual $\theta(g^*) = Z^*$. Portanto o valor ótimo do problema auxiliar é Z^*.

O objetivo desta seção é dar alguns resultados que relacionam o problema auxiliar 1 com o PL inicial e demonstrar que com algumas hipóteses adicionais a função θ é continuamente diferenciável num subconjunto de G.

Diremos que o problema inicial (PL) e o problema auxiliar 1 são **problemas equivalentes** porque a solução de um deles permite obter a solução do outro, uma vez que ambos têm o mesmo valor ótimo e se X^g é a solução ótima do $PA^V(g)$ e x^g é a correspondente solução desagregada de peso fixo, então pela proposição 1.2.3 x^g é viável para PL e

$$\theta(g) = \bar{c}X^g = cx^g. \tag{2.3}$$

Portanto, se g^* é ótimo para o problema auxiliar 1, então x^{g^*} é ótimo para PL. Reciprocamente se x^* é ótimo para (PL), então pela proposição 1.2.7 existe um peso ótimo g^* para o problema auxiliar 1. O seguinte resultado mostra que com uma condição adicional, o problema auxiliar 1 fornece um máximo global para o problema inicial.

Teorema 2.2.1 *Seja g^0 um máximo local do problema auxiliar 1 e X^0 solução ótima do problema agregado $PA^V(g^0)$, ($X^0 \in \mathcal{P}_{oti}(g^0)$) tal que $X_k^0 > 0$, $k = 1, \ldots, K$. Então a solução desagregada de peso fixo correspondente x^0 é ótima para PL.*

PROVA
Por ser g^0 um máximo local, então existe $\epsilon^0 > 0$ tal que

$$\theta(g^0) \geq \theta(g) \qquad \forall g \in B(g^0, \epsilon^0) \cap G. \tag{2.4}$$

Definimos na vizinhança $\mathcal{V}(x^0)$ a seguinte função

$$h : \mathcal{V}(x^0) \longrightarrow \mathbb{R}^n$$

onde para $i \in S_k$ temos

$$h_i(x) = \begin{cases} \frac{x_i}{\sum_{r \in S_k} x_r} & se \quad \sum_{r \in S_k} x_r \neq 0 \\ g_i^0 & se \quad \sum_{r \in S_k} x_r = 0 \end{cases}$$

Obviamente $h(x^0) = g^0$, porque

$$h_i(x^0) = \frac{x_i^0}{\sum_{r \in S_k} x_r^0} = \frac{X_k^0 g_i^0}{\sum_{r \in S_k} X_k^0 g_r^0} = \frac{g_i^0}{\sum_{r \in S_k} g_r} = g_i^0 \tag{2.5}$$

A função h é contínua em x^0, porque cada h_i é o quociente de duas funções contínuas e pela condição $X_k^0 > 0$ a função no denominador é diferente de zero em x^0.

Pela continuidade de h em x^0, existe $\delta > 0$ tal que $\forall x \in B(x^0, \delta)$ temos $h(x) \in B(g^0, \epsilon^0)$. Dado que $x^0 \in P$ e P é convexo, então

$$B(x^0, \delta) \cap P \neq \varnothing$$

Seja $\widetilde{x} \in B(x^0, \delta) \cap P$. Como h é contínua, $h(\widetilde{x}) \in B(g^0, \epsilon^0) \cap G$. Dado que g^0 é um máximo local, por (2.4) temos

$$\theta(g^0) \geq \theta(h(\widetilde{x}))$$

logo

$$cx^0 = \bar{c}X^0 = \theta(g^0) \geq \theta(h(\widetilde{x})) \tag{2.6}$$

Por outro lado, pela proposição 1.2.6, como $\widetilde{x}$ é viável para PL, então $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_K)$, definido por $\tilde{X}_k = \sum_{j \in S_k} \widetilde{x}_j$ é viável para $PA^V(h(\widetilde{x}))$. Logo

$$\bar{c}\tilde{X} \quad \leq \quad \theta(h(\widetilde{x})) \tag{2.7}$$

Pela mesma proposição temos

$$\bar{c}\tilde{X} = c\widetilde{x}. \tag{2.8}$$

De (2.6), (2.7) e (2.8) temos

$$cx^0 \geq c\widetilde{x}$$

$\forall \widetilde{x} \in B(x^0, \delta) \cap P$, indicando que x^0 é um máximo local para a função objetivo de PL e como ela é linear temos que x^0 é máximo global. Ou seja x^0 é solução ótima para PL.

■

Antes de demonstrar que a função objetivo do problema auxiliar 1 é diferenciável, desenvolveremos alguns aspectos da teoria de dualidade e a aplicaremos a $PA^V(g)$.

Dual Lagrangeano

De forma geral, dado um problema linear P, existe outro problema associado a ele chamado problema dual Lagrangeano. Se P tem solução ótima, então os valores ótimos de ambos problemas coincidem e o dual Lagrangeano tem como solução ótima os multiplicadores Lagrangeanos de P que correspondem à solução ótima.

Para cada $g \in G$, definimos a função Lagrangana associada a $PA^V(g)$ como

$$L(X,U) = \bar{c}X + U(b - \bar{A}X). \tag{2.9}$$

Consideremos agora a função dual Lagrangeana

$$q : \mathbb{R}^m_+ \to (-\infty, +\infty]$$

definida por

$$q(U) = \sup_{X \geq 0} L(X,U) \tag{2.10}$$

O **problema dual Lagrangeano** é

$$\begin{array}{ll} inf & q(U) \\ s.a & U \geq 0. \end{array}$$

A seguinte proposição será utilizada na demonstração do teorema de dualidade, quando f é linear.

Proposição 2.2.2 *Seja $f : \mathbb{R}^n \to \mathbb{R}$ convexa, continuamente diferenciável e seja $J \subset \{1, ..., r\}$. Então x^* é um máximo global para o problema*

$$\begin{array}{ll} max & f(x) \\ s.a & A_j x \leq b_j \quad j = 1, ...r. \end{array}$$

se e só se x^* é viável e existem escalares u_j^*, $j \in J$ tal que

$$u_j^* \geq 0, \quad j \in J,$$

$$u_j^* = 0, \quad \forall j \in J \quad com \quad j \notin A(x^*)$$

$$x^* = argmax_{x \in S}\{f(x) + \sum_{j \in J} u_j^*(A_j x - b_j)\}.$$

onde $S = \{A_j x \leq b_j \quad e \quad j \notin J\}$ e $A(x^*)$ representa o conjunto de restrições ativas em x^*.

PROVA (ver Bersekas[3]).

Teorema 2.2.3 **(Teorema de Dualidade)** *(a) Se o problema primal $PA^V(g)$ tem uma solução ótima, o problema dual Lagrangeano também tem solução ótima e os correspondentes valores ótimos são iguais.*
(b) $\widehat{X}$ é solução ótima para $PA^V(g)$ e $\widehat{U}$ é solução ótima do dual Lagrangeano, se e só se $\widehat{X} \in S(g)$, $\widehat{U} \geq 0$ e

$$\bar{c}\widehat{X} = L(\widehat{X}, \widehat{U}) = \sup_{X \geq 0} L(X, \widehat{U}) \tag{2.11}$$

PROVA
(a) $\forall X \in \mathcal{S}(g)$ e $\forall U \in \mathbb{R}^m_+$, temos

$$U(b - \bar{A}X) \geq 0,$$

assim

$$q(U) \geq \bar{c}X + U(b - \bar{A}X) \geq \bar{c}X. \tag{2.12}$$

Tomando o máximo em (2.12), sobre todos os $X \in S(g)$, temos

$$q(U) \geq \bar{c}\widehat{X} \quad \forall U \geq 0. \tag{2.13}$$

Pela proposição 2.2.2 existe $\widehat{U} \geq 0$ tal que

$$\widehat{U}_j \cdot (b_j - \bar{A}_j\widehat{X}) = 0 \quad \forall j \quad e$$

$$\widehat{X} = argmax_{X \geq 0} L(X, \widehat{U}).$$

Assim, usando a definição de q, temos

$$\begin{aligned} q(\widehat{U}) &= L(\widehat{X}, \widehat{U}) \\ &= \bar{c}\widehat{X} + \widehat{U}.(b - \bar{A}\widehat{X}) \\ &= \bar{c}\widehat{X} \end{aligned} \tag{2.14}$$

Combinando as equações (2.13) e (2.14), vemos que $\widehat{U}$ é solução ótima do dual Lagrangeano e que

$$q(\widehat{U}) = \bar{c}\widehat{X} \tag{2.15}$$

(b) Se $\widehat{X}$ é ótima primal e $\widehat{U}$ é ótima dual, pela parte (a), temos

$$q(\widehat{U}) = \bar{c}\widehat{X}$$

e de (2.12) vem

$$\bar{c}\widehat{X} = L(\widehat{X}, \widehat{U}) = q(\widehat{U}) = \sup_{X \geq 0} L(X, \widehat{U}).$$

Reciprocamente, a relação (2.11) pode ser escrita como

$$\bar{c}\widehat{X} = q(\widehat{U})$$

e por ser $\widehat{X}$ primal viável e $\widehat{U}$ dual viável, a equação (2.12) implica a otimalidade para PA^V e seu dual Lagrangeano respectivamente.

■

Observação 2.2.1

a) Pela demonstração do Teorema de Dualidade é claro que a solução ótima do dual $DA^V(g)$ é também solução ótima do dual Lagrangeano de $PA^V(g)$.

b) Pelo Teorema de Dualidade, para cada $g \in G$ o valor ótimo de $PA^V(g)$, denotado por $\theta(g)$ é igual ao valor ótimo do seu dual Lagrangeano. Além disso se $\widehat{X}$ e $\widehat{U}$ são soluções do $PA^V(g)$ e $DA^V(g)$ respectivamente, por 2.14 *temos*

$$\begin{aligned} \theta(g) &= \inf_{U \geq 0} \sup_{X \geq 0} L(X, U) \\ &= \inf_{U \geq 0} \sup_{X \geq 0} [\bar{c}X + U(b - \bar{A}X)] \qquad (2.16) \\ &= L(\widehat{X}, \widehat{U}) \end{aligned}$$

Observemos que em (2.16) *o Lagrangeano está em função de g, porque $\bar{c}$ e $\bar{A}$ foram definidos em função do peso. Enfatizamos isto substituindo $L(X,U)$ por $L(X,U,g)$. Portanto escreveremos*

$$\begin{aligned} \theta(g) &= \inf_{U \geq 0} \sup_{X \geq 0} L(X, U, g) \\ &= L(\widehat{X}, \widehat{U}, g) \qquad (2.17) \end{aligned}$$

Cálculo de $\nabla_g L$: Desenvolvendo para cada g o Lagrangeano associado ao Problema Agregado, temos

$$\begin{aligned}
L(X,U,g) &= \bar{c}X + U(b - \bar{A}X) \\
&= \sum_{k=1}^{K} \bar{c}_k X_k + Ub - U(\sum_{k=1}^{K} \bar{A}^k X_k) \\
&= \sum_{k=1}^{K} c^k g^k X_k + Ub - U(\sum_{k=1}^{K} B^k g^k X_k) \\
&= \sum_{k=1}^{K} \Big(\sum_{j \in S_k} c_j g_j \Big) X_k + Ub - U \sum_{k=1}^{K} \Big(\sum_{j \in S_k} A^j g_j \Big) X_k \\
&= \sum_{k=1}^{K} \sum_{j \in S_k} (c_j - UA^j) g_j X_k + Ub
\end{aligned}$$

Por conveniência de notação suponhamos que as colunas de PL tenham sido agrupadas segundo a ordem ascendente de seus índices. Se

$$\mathfrak{D}[X,U] = \Big[(c_1 - UA^1)X_1 \quad (c_2 - UA^2)X_1 \quad \ldots \quad (c_n - UA^n)X_K \Big]$$

Então

$$L(X,U,g) = \mathfrak{D}[X,U]g + Ub \tag{2.18}$$

De 2.18 temos

$$\nabla_g L/_{(X,U,g)} = \mathfrak{D}[X,U] \tag{2.19}$$

e para cada componente $j \in S_k$ temos

$$\frac{\partial L}{\partial g_j}\Big/_{(X,U,g)} = (c_j - UA^j)X_k.$$

É claro que o lagrangeano L(X,U,g) e seu gradiente (2.19) são contínuos.

Observemos que a função objetivo do problema auxiliar está definida num conjunto que não é aberto. Precisamos dar a definição de diferenciabilidade para este caso.

Definição 2.2.4 *Seja $F \subset \mathbb{R}^n$ não aberto e $\varphi : F \to \mathbb{R}$. Diremos que φ é diferenciável em F se ela pode ser estendida para uma função*

$$h : U \to \mathbb{R}$$

que é diferenciável no conjunto aberto $U \subset \mathbb{R}^n$ que contem F.

□

O seguinte resultado indica que uma função definida num conjunto não aberto é diferenciável se localmente em cada ponto x_0 do domínio ela pode ser estendida a uma função diferenciável. No próximo teorema expressamos isto formalmente.

Teorema 2.2.5 *Seja $F \subset \mathbb{R}^n$ e $\varphi : F \to \mathbb{R}$. Se para cada ponto $x_0 \in F$, existe uma vizinhança $\mathcal{V}(x_0)$ de x_0 e uma função*

$$h_{x_o} : \mathcal{V}(x_0) \to \mathbb{R}$$

diferenciável tal que

$$h_{x_o}/_{\mathcal{V}(x_0)\cap F} = \varphi.$$

Então φ é diferenciável em F.

PROVA (Ver Munkres [19])

Diferenciabilidade de θ

Demonstraremos que a função θ é diferenciável em

$$\widetilde{G} = \{g \in \mathbb{R}^n : g_j > 0 \quad e \quad \sum_{j \in S_k} g_j = 1\}$$

se as seguintes condições forem satisfeitas:

Hipótese 4

- Os conjuntos $\mathcal{P}_{oti}(g)$ e $\mathcal{D}_{oti}(g)$ são unitários $\forall g \in G$. Denotaremos por X^g e U^g as soluções ótimas únicas de $PA^V(g)$ e seu dual respectivamente.

- Os conjuntos $\{X^g : g \in G\}$ e $\{U^g : g \in G\}$ são limitados.

Pela última condição, podemos admitir que existe uma bola em $\mathbb{R}^K$

$$B_\infty[0;\alpha_1] = \left\{X \in \mathbb{R}^K : \max_{k=1,\dots,K} |X_k| \leq \alpha_1\right\}$$

que contém as soluções ótimas dos problemas agregados $PA^V(g)$ $\forall g \in G$ e também existe uma bola em $\mathbb{R}^m$

$$B_\infty[0;\alpha_2] = \left\{U \in \mathbb{R}^m : \max_{i=1,\dots,m} |U_i| \leq \alpha_2\right\}$$

que contém as soluções ótimas dos $DA^V(g)$ $\forall g \in G$.

Como as soluções ótimas de $PA^V(g)$ e seu dual têm componentes positivas, então os conjuntos compactos

$$K_1 = \left\{X \in \mathbb{R}^K_+ : \max_{k=1,\dots,K} X_k \leq \alpha_1\right\}$$

e

$$K_2 = \left\{U \in \mathbb{R}^m_+ : \max_{i=1,\dots,m} U_i \leq \alpha_2\right\}$$

contêm as soluções ótimas dos $PA^V(g)$ e $DA^V(g)$ respectivamente, $\forall g \in G$.

Seja

$$H = \mathbb{R}^n_{++}$$

Formemos um problema por meio de combinações lineares (não necessariamente convexas) das componentes do vetor de custos e das colunas da matriz de restrições do (PL) com coeficientes dados por l. Obtemos assim um novo vetor de custo $\widehat{c}$ e uma nova matriz de restrições $\widehat{A}$. Praticamente estamos construindo este novo problema da mesma maneira como formamos um problema agregado, mas com a única diferença de que l não é necessariamente um peso. Por abuso de notação escreveremos $PA^V(l)$ e

$$PA^V(l)\left\{\begin{array}{lll} \widehat{Z} = & & max \quad \widehat{c}x \\ & s.a & \widehat{A}x \leq b \\ & & x \geq 0 \end{array}\right.$$

seu dual

$$(\mathrm{DA}^V(l))\left\{\begin{array}{lll} \widehat{Z} = & & min \quad Ub \\ & s.a & U\widehat{A} \geq \widehat{c} \\ & & U \geq 0. \end{array}\right.$$

Se $g_0 \in \widetilde{G}$ e $B_2[g_0; \epsilon_0] \subset H$ demonstraremos que existe uma função

$$\theta^{g_0} : B_2(g_0; \epsilon_0) \longrightarrow \mathbb{R}$$

diferenciável, tal que

$$\theta^{g_0} / _{B_2(g_0;\epsilon_0)\cap\widetilde{G}} = \theta.$$

As proposições 2.2.7 e 2.2.8 fornecem alguns resultados que serão usados na proposição 2.2.9. Assumindo que a hipótese 4 é satisfeita demonstraremos que dados $g_0 \in \widetilde{G}$ e uma bola $B_2[g_0; \epsilon_0] \subset H$ teremos que

a) $\forall l \in B_2(g_0; \epsilon_0)$ o $PA^V(l)$ e o $DA^V(l)$ têm soluções únicas e

b) $\exists$ um conjunto compacto K^p em $\mathbb{R}^K$ que contêm as soluções ótimas dos problemas $PA^V(l)$ $\forall l \in B_2(g_0 : \epsilon_0)$ e $\exists$ outro compacto K^d em $\mathbb{R}^m$ que contêm as soluções ótimas dos problemas $DA^V(l)$ $\forall l \in B_2(g_0; \epsilon_0)$.

Definição 2.2.6 *Dado $l \in H$ diremos que $g(l) \in G$ definido como*

$$g(l)^k = \frac{l^k}{\sum_{j \in S_k} l_j} \quad para \quad k = 1, \ldots, K$$

é o **peso associado a** l.

□

A seguinte proposição mostra que dado $l \in H$ existe uma correspondência biunívoca entre o conjunto de soluções viáveis de $PA^V(l)$ e o conjunto de soluções viáveis de $PA^V(g(l))$.

Proposição 2.2.7 *Dado $l \in H$ e seja $g(l)$ o peso associado a l, então*

a) $\bar{X} = (\bar{X}_1, \ldots, \bar{X}_K)$ é viável para $PA^V(g(l))$ se e só se $\widehat{y} = (\widehat{y}_1, \ldots, \widehat{y}_K)$ definida por

$$\widehat{y}_k = \frac{\bar{X}_k}{\sum_{j \in S_k} l_j}$$

é viável para $PA^V(l)$. Além disso

$$\bar{c}\bar{X} = \widehat{c}\widehat{y}.$$

b) U é viável para $DA^V(g(l))$ se e só se U é viável para $DA^V(l)$.

PROVA
a)

$$\begin{aligned}
\bar{A}\bar{X} \quad &= \textstyle\sum_{k=1}^{K} \bar{A}_k \bar{X}_k \quad &= \displaystyle\sum_{k=1}^{K} B^k g(l)^k \bar{X}_k \\
&= \textstyle\sum_{k=1}^{K} B^k \frac{l^k}{\sum_{j\in S_k} l_j} \bar{X}_k \quad &= \displaystyle\sum_{k=1}^{K} B^k l^k \widehat{y}_k \\
&= \textstyle\sum_{k=1}^{K} \widehat{A}_k \widehat{y}_k \quad &= \widehat{A}\widehat{y}. \qquad (2.20)
\end{aligned}$$

Desta igualdade vemos que $\bar{X}$ é viável para $PA^V(g(l))$ se e só se $\widehat{y}$ é viável para $PA^V(l)$.

$$\begin{aligned}
\bar{c}\bar{X} \quad &= \textstyle\sum_{k=1}^{K} \bar{c}_k \bar{X}_k \quad &= \displaystyle\sum_{k=1}^{K} c^k g(l)^k \bar{X}_k \\
&= \textstyle\sum_{k=1}^{K} c^k \frac{l^k}{\sum_{j\in S_k} l_j} \bar{X}_k \quad &= \displaystyle\sum_{k=1}^{K} c^k l^k \widehat{y}_k \\
&= \textstyle\sum_{k=1}^{K} \widehat{c}_k \widehat{y}_k = \widehat{c}\widehat{y} \qquad (2.21)
\end{aligned}$$

b) Demonstraremos que para $k = 1, \ldots, K$

$$U\bar{A}^k \geq \bar{c}_k \quad \Longleftrightarrow \quad U\widehat{A}^k \geq \widehat{c}_k.$$

Para $k = 1, \ldots, K$ temos

$$U\bar{A}^k \geq \bar{c}_k$$

se e só se

$$U(\sum_{j\in S_k} A^j g(l)_j) \quad \geq \quad (\sum_{j\in S_k} c_j g(l)_j)$$

se e só se

$$U\Big(\sum_{j\in S_k} A^j \frac{l_j}{\sum_{r\in S_k} l_r}\Big) \quad \geq \quad \Big(\sum_{j\in S_k} c_j \frac{l_j}{\sum_{r\in S_k} l_r}\Big)$$

se e só se

$$U\Big(\sum_{j\in S_k} A^j l_j\Big) \geq \Big(\sum_{j\in S_k} c_j l_j\Big)$$

Esta última desigualdade é equivalente a

$$U\widehat{A}^k \quad \geq \quad \widehat{c}_k$$

Portanto U é viável para $DA^V(g(l))$ se e só se U é viável para $DA^V(l)$.

Proposição 2.2.8 *Dados $PA^V(l)$ e $PA^V(g(l))$,*

a) $\bar{X}$ é solução ótima de $PA^V(g(l))$ se e só se $\widehat{y} = (\widehat{y}_1, \ldots, \widehat{y}_K)$, definido por

$$\widehat{y}_k = \frac{\bar{X}_k}{\sum_{j \in S_k} l_j}$$

é solução ótima de $PA^V(l)$.

b) U é solução ótima do $DA^V(g(l))$ se e só se U é solução ótima do $DA^V(l)$.

PROVA

a) Suponha que $\bar{X}$ é solução ótima de $PA^V(g(l))$. Pela proposição 2.2.7 $\widehat{y}$ é viável para $PA^V(l)$ e

$$\bar{c}\bar{X} = \widehat{c}\widehat{y}. \tag{2.22}$$

Se $\widehat{y}$ não é ótima para $PA^V(l)$, então existe um y^* viável para $PA^V(l)$ tal que

$$\widehat{c}y^* > \widehat{c}\widehat{y}. \tag{2.23}$$

Pela proposição 2.2.7 $X^* = (\sum_{j \in S_1} l_j y_1^*, \ldots, \sum_{j \in S_K} l_j y_K^*)$ é viável para $PA^V(g(l))$ e

$$\bar{c}X^* = \widehat{c}y^*. \tag{2.24}$$

Das relações 2.22, 2.23 e 2.24 temos $\bar{c}\bar{X} < \bar{c}X^*$. Isto contradiz que $\bar{X}$ é solução ótima de $PA^V(g(l)$. Portanto $\widehat{y}$ é ótima para $PA^V(l)$.

A prova da recíproca é análoga a primeira parte.

b) Pela proposição 2.2.7 o conjunto de soluções viáveis do $DA^V(g(l))$ é igual ao conjunto de soluções viáveis do $DA^V(l)$ e a função objetivo é a mesma para ambos problemas, então U é solução ótima de $DA^V(g(l))$ se e só se U é solução ótima de $DA^V(l)$.

■

Proposição 2.2.9 *Dado $g_0 \in \widetilde{G}$ e seja $B_2[g_0; \epsilon_0] \subset H$. Se a hipótese 4 é satisfeita então*

a) $\forall l \in B_2(g_0; \epsilon_0)$ o $PA^V(l)$ e o $DA^V(l)$ têm soluções únicas e

b) $\exists$ um conjunto compacto K^p em $\mathbb{R}^K$ que contêm as soluções ótimas dos problemas $PA^V(l)$ $\forall l \in B_2(g_0;\epsilon_0)$ e $\exists$ outro compacto K^d em $\mathbb{R}^m$ que contêm as soluções ótimas dos problemas $DA^V(l)$ $\forall l \in B_2(g_0;\epsilon_0)$.

PROVA

a) Dado $l \in B_2(g_0;\epsilon_0)$ e seja $g(l)$ o peso associado a l (ver definição 2.2.6). Pela hipótese 4 sabemos que $PA^V(g(l))$ e $DA^V(g(l))$ têm soluções ótimas únicas. Pela proposição 2.2.8 deduzimos que os problemas $PA^V(l)$ e $DA^V(l)$ também têm solução única. Pela parte a) da mesma proposição sabemos que se $\bar{X} = (\bar{X}_1, \ldots, \bar{X}_K)$ é solução ótima única de $PA^V(g(l))$, então $\widehat{y} = (\widehat{y}_1, \ldots, \widehat{y}_K)$ definida por

$$\widehat{y}_k = \frac{\bar{X}_k}{\sum_{j \in S_k} l_j}$$

é solução ótima única para $PA^V(l)$. Pela parte b) ainda da mesma proposição, sabemos que se U é solução ótima única de $DA^V(g(l))$, então U é também solução ótima única de $DA^V(l)$.

b) Para $k = 1, \ldots, K$ definimos as funções

$$p_k : B_2[g_0;\epsilon_0] \longrightarrow \mathbb{R}$$

como

$$p_k(x_1, \ldots, x_n) \;=\; \sum_{j \in S_k} x_j.$$

As funções p_k são contínuas e o domínio é compacto. Seja

$$m_k = \min_{x \in B_2[g_0;\epsilon_0]} p_k(x) \quad e \quad M = \min_{k=1,\ldots,K} m_k.$$

Demonstraremos que o conjunto

$$K^p = \left\{ X \in R_+^K : \max_{k=1,\ldots,K} X_k \leq \alpha_p \right\}$$

onde $\alpha_p = \alpha_1/M$, contém as soluções ótimas dos problemas $PA^V(l)$ $\forall l \in B_2(g_0;\epsilon_0)$. Por a) $\widehat{y} = (\widehat{y}_1, \ldots, \widehat{y}_K)$ é solução de $PA^V(l)$, onde

$$\widehat{y}_k = \frac{\bar{X}_k}{\sum_{j \in S_k} l_j} \quad e \quad (\bar{X}_1, \ldots, \bar{X}_K) \in K_1$$

Portanto

$$\widehat{y}_k \leq \frac{\bar{X}_k}{M} \leq \frac{\alpha_1}{M} = \alpha_p.$$

e $\widehat{y} \in K^p$.

Pelo demonstrado ao final do ítem a), podemos escolher $K^d = K_2$.

■

Extensão Local de θ

Agora que sabemos que $\forall l \in B_2(g_0; \epsilon_0)$ o $PA^V(l)$ e o $DA^V(l)$ tem soluções ótimas únicas, usaremos a notação g para indicar qualquer elemento da $B_2(g_0; \epsilon_0)$ e X^g e U^g para denotar as únicas soluções ótimas do primal e dual respectivamente. Definiremos a extensão local de θ em $B_2(g_0; \epsilon_0)$

por

$$\theta^{g_0}(g) = \inf_{U \in K^d} \sup_{X \in K^p} L(X, U, g)$$

e pela observação 2.2.1

$$\theta^{g_0}(g) = \inf_{U \in K^d} \sup_{X \in K^p} L(X, U, g) = L(X^g, U^g, g)$$

Como $K_1 \subset K^p$ e $K_2 = K^d$, $\forall g \in B_2(g_0; \epsilon_0) \cap \widetilde{G}$ temos

$$\theta^{g_0}(g) = L(X^g, U^g, g) = \theta(g).$$

Como L é contínua e K^d e K^p são compactos, então podemos escrever

$$\theta^{g_0}(g) = \min_{U \in K^d} \max_{X \in K^p} L(X, U, g) = L(X^g, U^g, g)$$

No restante desta seção provaremos que θ^{g_0} é diferenciável em $B_2(g_0; \epsilon_0)$. Para facilitar a demonstração, usaremos a função

$$Q^{g_0} : K^d x B_2(g_0; \epsilon_0) \to \mathbb{R}$$

definida como

$$Q^{g_0}(U, g) = \max_{X \in K^p} L(X, U, g)$$

Definição 2.2.10 *Dados $g \in \mathbb{R}^n$, $\bar{g} \in B_2(g_0; \epsilon_0)$, com $U^{\bar{g}} \in \mathcal{D}_{oti}(\bar{g})$, definimos o conjunto*

$$\gamma[U^{\bar{g}}, g] = argmax_{X \in K^p} L(X, U^{\bar{g}}, g)$$

□

O gradiente $\nabla_g Q^{g_0}$ em $(U^{\bar{g}}, \bar{g})$ depende do conjunto $\gamma[U^{\bar{g}}, \bar{g}]$. Em particular na proposição 2.2.12 demonstraremos que se $\gamma[U^{\bar{g}}, \bar{g}]$ é unitário (solução única de $PA^V(\bar{g})$) então existe $\nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g})$. Primeiro veremos a seguinte proposição preliminar:

Proposição 2.2.11 *Dados $\bar{g} \in B_2(g_0; \epsilon_0)$ e $g_r \in \mathbb{R}^n$, suponha que $\gamma[U^{\bar{g}}, \bar{g}] = \{X^{\bar{g}}\}$. Se $g_r \to \bar{g}$, e $X^{(r)} \in \gamma[U^{\bar{g}}, g_r]$ para cada r, então $X^{(r)} \to X^{\bar{g}}$.*

PROVA
Vamos supor por o absurdo que $X^{(r)} \nrightarrow X^{\bar{g}}$. Então existe $\epsilon > 0$, tal que $\| X^{(r)} - X^{\bar{g}} \| > \epsilon$ para todo r que pertence a algum conjunto infinito N_1. Como K^p é compacto, sabemos que existe uma sub-sequência convergente. Então existe um $W \in K^p$ e um conjunto infinito $N_2 \subset N_1$ tal que para $r \in N_2$ temos $\lim_{r\to\infty} X^{(r)} = W$.
Por outro lado, para cada g_r com $r \in N_2$, pela definição de $\gamma[U^{\bar{g}}, g_r]$ temos

$$L(X^{(r)}, U^{\bar{g}}, g_r) \geq L(X^{\bar{g}}, U^{\bar{g}}, g_r)$$

Notando que $X^{(r)} \to W$, e $g_r \to \bar{g}$ para $r \in N_2$ e pela continuidade do Lagrangeano, temos que

$$L(W, U^{\bar{g}}, \bar{g}) \geq L(X^{\bar{g}}, U^{\bar{g}}, \bar{g}).$$

Então $W \in \gamma[U^{\bar{g}}, \bar{g}]$, o que contradiz a hipótese de que $\gamma[U^{\bar{g}}, \bar{g}]$ ser unitário.
■

Proposição 2.2.12 *Dado $\bar{g} \in B_2(g_0; \epsilon_0)$ e sejam $X^{\bar{g}} \in \mathcal{P}_{oti}(\bar{g})$ e $U^{\bar{g}} \in \mathcal{D}_{oti}(\bar{g})$, então*

$$\nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g}) = \nabla_g L(X^{\bar{g}}, U^{\bar{g}}, \bar{g}) \tag{2.25}$$

PROVA
Se

$$Q^{g_0}(U^{\bar{g}}, g_r) = \max_{X \in K^p} L(X, U^{\bar{g}}, g_r) = L(X^{(r)}, U^{\bar{g}}, g_r) \geq L(X^{\bar{g}}, U^{\bar{g}}, g_r) \tag{2.26}$$

e

$$Q^{g_0}(U^{\bar{g}}, \bar{g}) = \max_{X \in K^p} L(X, U^{\bar{g}}, \bar{g}) = L(X^{\bar{g}}, U^{\bar{g}}, \bar{g}) \geq L(X^{(r)}, U^{\bar{g}}, \bar{g}). \tag{2.27}$$

De 2.26 e 2.27, temos

$$Q^{g_0}(U^{\bar{g}}, g_r) - Q^{g_0}(U^{\bar{g}}, \bar{g}) \leq L(X^{(r)}, U^{\bar{g}}, g_r) - L(X^{(r)}, U^{\bar{g}}, \bar{g})$$

e

$$Q^{g_0}(U^{\bar{g}}, g_r) - Q^{g_0}(U^{\bar{g}}, \bar{g}) \geq L(X^{\bar{g}}, U^{\bar{g}}, g_r) - L(X^{\bar{g}}, U^{\bar{g}}, \bar{g}).$$

Usando a notação de 2.18, podemos escrever estas últimas desigualdades como

$$\mathfrak{D}(X^{\bar{g}}, U^{\bar{g}})(g_r - \bar{g}) \leq Q^{g_0}(U^{\bar{g}}, g_r) - Q^{g_0}(U^{\bar{g}}, \bar{g}) \leq \mathfrak{D}(X^{(r)}, U^{\bar{g}})(g_r - \bar{g}).$$

Logo, pela desigualdade de Schwartz,

$$\begin{aligned} 0 &\leq Q^{g_0}(U^{\bar{g}}, g_r) - Q^{g_0}(U^{\bar{g}}, \bar{g}) - \mathfrak{D}(X^{\bar{g}}, U^{\bar{g}})(g_r - \bar{g}) \\ &\leq [\mathfrak{D}(X^{(r)}, U^{\bar{g}}) - \mathfrak{D}(X^{\bar{g}}, U^{\bar{g}})](g_r - \bar{g}) \\ &\leq \| \mathfrak{D}(X^{(r)}, U^{\bar{g}}) - \mathfrak{D}(X^{\bar{g}}, U^{\bar{g}}) \| \| g_r - \bar{g} \| . \end{aligned} \tag{2.28}$$

Se dividirmos a desigualdade anterior por $\| g_r - \bar{g} \|$, obtemos

$$0 \leq \frac{Q^{g_0}(U^{\bar{g}}, g_r) - Q^{g_0}(U^{\bar{g}}, \bar{g}) - \mathfrak{D}(X^{\bar{g}}, U^{\bar{g}})(g_r - \bar{g})}{\| g_r - \bar{g} \|} \leq \| \mathfrak{D}(X^{(r)}, U^{\bar{g}}) - \mathfrak{D}(X^{\bar{g}}, U^{\bar{g}}) \|$$

aplicando limite quando $g_r \to \bar{g}$ e usando prop 2.2.11 temos

$$\lim_{g_r \to \bar{g}} \frac{Q^{g_0}(U^{\bar{g}}, g_r) - Q^{g_0}(U^{\bar{g}}, \bar{g}) - \mathfrak{D}(X^{\bar{g}}, U^{\bar{g}})(g_r - \bar{g})}{\| g_r - \bar{g} \|} = 0.$$

Portanto

$$\nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g}) = \mathfrak{D}(X^{\bar{g}}, U^{\bar{g}}) = \nabla_g L(X^{\bar{g}}, U^{\bar{g}}, \bar{g})$$

■

A seguinte proposição é chave para demonstrar a continuidade da função Q^{g_0}

Proposição 2.2.13 *Dada a função* $L : K^p \times K^d \times B_2(g_0; \epsilon_0) \to \mathbb{R}$ *e um ponto fixo* $(\widetilde{U}, \widetilde{g}) \in K^d \times B_2(g_0; \epsilon_0)$. *Para todo* $\epsilon > 0$ *existe* $\delta > 0$ *tal que se* $(U, g) \in K^d \times B_2(g_0; \epsilon_0)$ *e*

$$\| (U, g) - (\widetilde{U}, \widetilde{g}) \| < \delta \quad \textit{então} \quad |L(X, U, g) - L(X, \widetilde{U}, \widetilde{g})| < \epsilon$$

para qualquer $X \in K^p$.

PROVA

Por absurdo, vamos supor que exista $\epsilon > 0$ e seqüências de pontos (U_n, g_n) e X_n em $K^d \times B_2(g_0; \epsilon_0)$ e K^p respectivamente, tais que

$$\| (U_n, g_n) - (\widetilde{U}, \widetilde{g}) \| < 1/n \tag{2.29}$$

e

$$|L(X_n, U_n, g_n) - L(X_n, \widetilde{U}, \widetilde{g})| > \epsilon \tag{2.30}$$

Como a seqüência $X_n \in K^p$ e K^p é compacto, ela tem uma sub-seqüência convergente. Ou seja existe $X^* \in K^p$ tal que

$$\lim_{n\to\infty} X_n = X^* \quad para \quad n \in N_1 \subset N.$$

De 2.29 podemos observar que $(U_n, g_n) \to (\widetilde{U}, \widetilde{g})$. Pela continuidade de L e aplicando limite em (2.30) temos

$$|L(X^*, \widetilde{U}, \widetilde{g}) - L(X^*, \widetilde{U}, \widetilde{g})| > \epsilon$$

obtendo assim uma contradição.

■

Proposição 2.2.14 *A função $Q^{g_0} : K^d \times B_2(g_0; \epsilon_0) \to R$ é contínua.*

PROVA

Seja $(\widetilde{U}, \widetilde{g}) \in K^d \times B_2(g_0; \epsilon_0)$. Pela proposição anterior, dado $\epsilon > 0$, existe $\delta > 0$, tal que se

$$\| (U, g) - (\widetilde{U}, \widetilde{g}) \| < \delta$$

então

$$|L(X, U, g) - L(X, \widetilde{U}, \widetilde{g})| < \epsilon \tag{2.31}$$

para qualquer $X \in K^p$. Por propriedade do máximo (ver Apêndice A) sabemos que

$$\max_{X\in K^p} L(X, U, g) \leq \max_{X\in K^p} \Big(L(X, U, g) - L(X, \widetilde{U}, \widetilde{g})\Big) + \max_{X\in K^p} L(X, \widetilde{U}, \widetilde{g})$$

logo

$$\max_{X\in K^p} L(X, U, g) - \max_{X\in K^p} L(X, \widetilde{U}, \widetilde{g}) \leq \max_{X\in K^p} \Big(L(X, U, g) - L(X, \widetilde{U}, \widetilde{g})\Big)$$

Portanto, de (2.31) temos

$$\max_{X\in K^p} L(X, U, g) - \max_{X\in K^p} L(X, \widetilde{U}, \widetilde{g}) < \epsilon \tag{2.32}$$

Analogamente temos

$$\max_{X\in K^p} L(X, \widetilde{U}, \widetilde{g}) \leq \max_{X\in K^p} \Big(L(X, \widetilde{U}, \widetilde{g}) - L(X, U, g)\Big) + \max_{X\in K^p} L(X, U, g)$$

logo

$$\max_{X\in K^p} L(X,\widetilde{U},\widetilde{g}) - \max_{X\in K^p} L(X,U,g) \leq \max_{X\in K^p}\Big(L(X,\widetilde{U},\widetilde{g}) - L(X,U,g)\Big).$$

Portanto, de (2.31) temos

$$\max_{X\in K^p} L(X,\widetilde{U},\widetilde{g}) - \max_{X\in K^p} L(X,U,g) < \epsilon \tag{2.33}$$

De (2.32) e (2.33) temos

$$|\max_{X\in K^p} L(X,U,g) - \max_{X\in K^p} L(X,\widetilde{U},\widetilde{g})| < \epsilon.$$

Ou seja

$$|Q^{g_0}(U,g) - Q^{g_0}(\widetilde{U},\widetilde{g})| < \epsilon$$

■

Definição 2.2.15 *Dado $g \in \mathbb{R}^n$. Definimos o conjunto*

$$\eta[g] = argmin_{U\in K^d} Q^{g_0}(U,g)$$

□

A diferenciabilidade de θ^{g_0} em $\bar{g}$ depende dos elementos do conjunto $\eta[\bar{g}]$. Em particular se $\eta[\bar{g}]$ é unitário (solução única do dual de $PA^V(\bar{g})$), então demonstraremos na proposição 2.2.17 que θ^{g_0} é diferenciável em $\bar{g}$. A seguinte proposição se faz necessária:

Proposição 2.2.16 *Sejam $\bar{g} \in B_2(g_0;\epsilon_0)$ e $g_r \in \mathbb{R}^n$. Suponha que*

$$\eta[\bar{g}] = \{U^{\bar{g}}\}.$$

Se $g_r \to \bar{g}$ e $U^{(r)} \in \eta[g_r]$, então $U^{(r)} \to U^{\bar{g}}$.

PROVA (análoga a prova da prop (2.2.11)
Vamos supor por absurdo que $U^{(r)} \nrightarrow U^{\bar{g}}$. Então existe $\epsilon > 0$ tal que $\| U^{(r)} - U^{\bar{g}} \| > \epsilon$ para todo r que pertence a algum conjunto infinito M_1. Como K^d é compacto, existe uma sub-seqüência convergente. Então existe $Y \in K^d$ e um conjunto infinito $M_2 \subset M_1$ tal que para $r \in M_2$ temos $\lim_{r\to\infty} U^{(r)} = Y$. Pela definição de $\eta[g_r]$ temos

$$Q^{g_0}(U^{(r)},g_r) \leq Q(U^{\bar{g}},g_r).$$

Se $r \to \infty$ e $r \in M_2$, então pela continuidade de Q^{g_0} temos

$$Q^{g_0}(Y, \bar{g}) \leq Q^{g_0}(U^{\bar{g}}, \bar{g}).$$

Ou seja $Y \in \eta[\bar{g}]$. Isto é uma contradição, porque $\eta[\bar{g}]$ é unitário.

■

Teorema 2.2.17 *Dado $\bar{g} \in B_2(g_0; \epsilon_0)$ com $X^{\bar{g}}$ e $U^{\bar{g}}$ soluções ótimas de $PA^V(\bar{g})$ e seu dual respectivamente, então a função θ^{g_0} é diferenciável em $\bar{g}$ e*

$$\nabla_g \theta^{g_0}(\bar{g}) = \nabla_g L(X^{\bar{g}}, U^{\bar{g}}, \bar{g}) \tag{2.34}$$

PROVA.
De

$$\theta^{g_0}(\bar{g}) = \min_{U \in K^d} Q^{g_0}(U, \bar{g}) = Q^{g_0}(U^{\bar{g}}, \bar{g}) \leq Q^{g_0}(U^{(r)}, \bar{g}) \tag{2.35}$$

e

$$\theta^{g_0}(g_r) = \min_{U \in K^d} Q^{g_0}(U, g_r) = Q^{g_0}(U^{(r)}, g_r) \leq Q^{g_0}(U^{\bar{g}}, g_r), \tag{2.36}$$

temos a seguinte desigualdade

$$Q^{g_0}(U^{(r)}, g_r) - Q^{g_0}(U^{(r)}, \bar{g}) \leq \theta^{g_0}(g_r) - \theta^{g_0}(\bar{g}) \leq Q^{g_0}(U^{\bar{g}}, g_r) - Q^{g_0}(U^{\bar{g}}, \bar{g}). \tag{2.37}$$

Subtraindo $\nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g})(g_r - \bar{g})$ em (2.37) e separando em duas desigualdades, temos

$$\begin{aligned} &Q^{g_0}(U^{(r)}, g_r) - Q^{g_0}(U^{(r)}, \bar{g}) - \nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g})(g_r - \bar{g}) \\ &\qquad \leq \theta^{g_0}(g_r) - \theta^{g_0}(\bar{g}) - \nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g})(g_r - \bar{g}) \end{aligned}$$

e

$$\begin{aligned} &\theta^{g_0}(g_r) - \theta^{g_0}(\bar{g}) - \nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g})(g_r - \bar{g}) \\ &\qquad \leq Q^{g_0}(U^{\bar{g}}, g_r) - Q^{g_0}(U^{\bar{g}}, \bar{g}) - \nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g})(g_r - \bar{g}). \end{aligned}$$

Se dividimos ambas desigualdades por $\| g_r - \bar{g} \|$, aplicamos limite e usamos a prop 2.2.16 na primeira desigualdade, temos

$$\begin{aligned} &\lim_{g_r \to \bar{g}} \frac{Q^{g_0}(U^{\bar{g}}, g_r) - Q^{g_0}(U^{\bar{g}}, \bar{g}) - \nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g})(g_r - \bar{g})}{\| g_r - \bar{g} \|} \leq \\ &\lim_{g_r \to \bar{g}} \frac{\theta^{g_0}(g_r) - \theta^{g_0}(\bar{g}) - \nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g})(g_r - \bar{g})}{\| g_r - \bar{g} \|} \end{aligned} \tag{2.38}$$

Portanto

$$0 \leq \lim_{g_r \to \bar{g}} \frac{\theta^{g_0}(g_r) - \theta^{g_0}(\bar{g}) - \nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g})(g_r - \bar{g})}{\| g_r - \bar{g} \|} \tag{2.39}$$

Analogamente, a desigualdade no outro sentido nos leva a

$$\lim_{g_r \to \bar{g}} \frac{\theta^{g_0}(g_r) - \theta^{g_0}(\bar{g}) - \nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g})(g_r - \bar{g})}{\| g_r - \bar{g} \|} \leq 0 \tag{2.40}$$

De (2.39), (2.40) e a proposição 2.2.12 segue que

$$\nabla \theta^{g_0}(\bar{g}) = \nabla_g Q^{g_0}(U^{\bar{g}}, \bar{g}) = \nabla_g L(X^{\bar{g}}, U^{\bar{g}}, \bar{g}) \tag{2.41}$$

■

2.3 Aplicação do metodo do gradiente condicional

Nesta seção apresentaremos um método computacional para resolver o seguinte problema de otimização com restrições

$$\begin{aligned} max \quad & f(x) \\ s.a \quad & x \in X \end{aligned} \tag{2.42}$$

onde assumiremos que:

a) X é um subconjunto de $\mathbb{R}^n$ não vazio, fechado e convexo.

b) f é continuamente diferenciável sobre X.

As principais idéias dos algoritmos para resolver os problemas de otimização restrita e irrestrita são bastante similares. O método que vamos apresentar está baseado em crescimento iterativo ao longo de direções. Mas estas direções devem ter a propriedade adicional de manter viabilidade para os iterados. Tais direções são chamadas viáveis e elas são usualmente obtidas resolvendo alguns sub-problemas de otimização. Como uma regra geral, estes sub-problemas para encontrar direções tendem a ser mais simples quando o conjunto de restrições é um poliedro.

Antes de detalhar o método, daremos as condições necessárias e suficientes de otimalidade para problemas com restrições.

Condições necessárias e suficientes de Otimalidade

Chamaremos de direções viáveis no ponto x^* a todo $\Delta x = x - x^*$, tal que $x + \alpha\Delta x \in X$ para todo α suficientemente pequeno. Como em otimização irrestrita, esperamos que num máximo local x^*, as direções de primeira ordem $\nabla f(x^*)'\Delta x$ sejam não positivas, para pequenas direções viáveis Δx. Como X é convexo, as direções da forma $\Delta x = x - x^*$, onde $x \in X$ são direções viáveis, permitindo que condição necessária seja escrita como $\nabla f(x^*)'(x - x^*) \leq 0$ $\forall x \in X$. A seguinte proposição demonstra formalmente esta condição que é também uma condição suficiente para otimalidade quando f é côncava.

Proposição 2.3.1 [Condição de Otimalidade]
a) Se x^* é um máximo local de f em X, então

$$\nabla f(x^*)'(x - x^*) \leq 0, \quad \forall x \in X \tag{2.43}$$

b) Se f é côncava sobre X, então a condição da parte a) é também suficiente para que x^* seja um máximo global de f em X.

PROVA

a) Suponha que $\nabla f(x^*)'(x - x^*) > 0$ para algum $x \in X$. Pelo teorema do valor médio (ver Apêndice) para todo $\epsilon > 0$ existe um $s \in [0, 1]$ tal que

$$f(x^* + \epsilon(x - x^*)) = f(x^*) + \epsilon\nabla f(x^* + s\epsilon(x - x^*))'(x - x^*).$$

Como ∇f é contínua, temos $\forall\epsilon$ suficientemente pequeno $\epsilon > 0$, $\nabla f(x^* + s\epsilon(x - x^*))'(x - x^*) \geq 0$ e portanto $f(x^* + \epsilon(x - x^*)) > f(x^*)$. O vetor $x^* + \epsilon(x - x^*)$ é viável para todo $\epsilon \in [0, 1]$ porque X é convexo, assim obtemos uma contradição da otimalidade local de x^*.

b) Usando propriedade da côncavidade de f (ver Apêndice) temos

$$f(x) \leq f(x^*) + \nabla f(x^*)'(x - x^*) \quad \forall x \in X.$$

Se a condição $\nabla f(x^*)'(x - x^*) \leq 0, \quad \forall x \in X$ obtemos $f(x) \leq f(x^*)$ e portanto x^* maximiza f sobre X.

■

Definição 2.3.2 *Diremos que x^* é um ponto estacionário se satisfaz a condição de otimalidade (2.43)*

Na ausência de côncavidade de f, a condição em (2.43) pode também ser satisfeita por máximos locais ou pontos de inflexão.

O Método do gradiente condicional pertence à classe dos assim chamados métodos de direções viáveis que passaremos a descrever.

Métodos de direções viáveis

A inicialização do método de direções viáveis é feita com um ponto viável x^0 e gera uma sequência de pontos viáveis x^r por meio de

$$x^{r+1} = x^r + \alpha^r d^r \tag{2.44}$$

onde, se x^r não é estacionário, d^r é uma direção viável tal que

$$\nabla f(x^r)' d^r > 0 \tag{2.45}$$

e o tamanho de passo α^r é escolhido para ser positivo e tal que

$$x^r + \alpha^r d^r \in X. \tag{2.46}$$

Se x^r é estacionário, o método pára, isto é, $x^{r+1} = x^r$. Estamos interessados em métodos de direções viáveis que garantam crescimento da função objetivo, isto é, o tamanho de passo α^r é selecionadado de modo que

$$f(x^r + \alpha^r d^r) > f(x^r), \quad \forall r. \tag{2.47}$$

Em nosso caso onde X é convexo, um método de direção viável pode ser escrito na forma

$$x^{r+1} = x^r + \alpha^r(\bar{x}^r - x^r) \tag{2.48}$$

onde

$$\alpha^r \in (0, 1], \tag{2.49}$$

e se x^r não é estacionário,

$$\bar{x}^r \in X, \quad \nabla f(x^r)'(\bar{x}^r - x^r) > 0. \tag{2.50}$$

Note que se x^r não é estacionário, existe sempre uma direção viável $\bar{x}^r - x^r$ que garante o crescimento de f, ou seja satisfazendo (2.50), uma vez que de outro modo deveríamos ter $\nabla f(x^r)'(x - x^r) \leq 0 \ \forall x \in X$, implicando que x^r é estacionário.

Seleção de tamanho de passo em métodos de direções viáveis

A maioria das regras para escolher o tamanho de passo α^r em métodos do gradiente (otimização irrestrita) se aplicam também aos métodos de direções voáveis. Utilizaremos a regra de Armijo:

- **Regra de Armijo**

Aqui, fixamos escalares β e $\sigma > 0$, com $\beta \in (0,1)$ e $\sigma \in (0,1)$ e fazemos $\alpha^r = \beta^{m_r}$, onde m_r é o primeiro inteiro não negativo m para o qual

$$f(x^r) - f(x^r + \beta^m(\bar{x}^r - x^r)) \leq -\sigma\beta^m \nabla f(x^r)'(\bar{x}^r - x^r) \tag{2.51}$$

Em outras palavras, o tamanho de passo $1, \beta, \beta^2, \ldots,$ são tomadas sucessivamente até que a desigualdade anterior é satisfeita para $m = m_r$.

Método do gradiente condicional

A forma mais direta para gerar uma direção viável $\bar{x}^r - x^r$ satisfazendo a condição $\nabla f(x^r)'(\bar{x}^r - x^r) > 0$ é resolvendo o seguinte problema

$$\begin{aligned} max \quad & \nabla f(x^r)'(x - x^r) \\ s.a \quad & x \in X \end{aligned} \tag{2.52}$$

e obtemos $\bar{x}^r$ como a solução, isto é,

$$\bar{x}^r = argmax_{x \in X} \nabla f(x^r)'(x - x^r). \tag{2.53}$$

Aqui assumimos que X seja compacto para que o problema (2.52) tenha solução. Este método é conhecido como método do gradiente condicional.

A solução do problema (2.52) vai nos permitir verificar se x^r é um ponto estacionário ou não. Se o valor ótimo de (2.52) é igual a zero, então $\forall x \in X$ temos $\nabla f(x^r)(x - x^r) \leq 0$, o que indica que x^r é um ponto estacionário. No caso em que o valor ótimo de (2.52) seja positivo então utilizaremos a solução $\bar{x}^r$ para obter o próximo iterado

$$x^{r+1} = x^r + \alpha^r(\bar{x}^r - x^r)$$

onde α^r é escolhido com a regra de Armijo ou a regra de maximização limitada.

A seqüência (x^r) assim obtida converge para pontos estacionários de (2.42). A demonstração está em Bertsekas[3].

Aplicação do método do gradiente condicional

Utilizaremos o método do gradiente condicional para achar uma solução aproximada do seguinte problema de programação linear apresentado na segunda seção deste capítulo.

$$(\text{PL}) \left\{ \begin{array}{rrl} Z^* = & & max \quad cx \\ & s.a & Ax \leq b \\ & & x \geq 0 \end{array} \right.$$

Para isso vamos considerar o problema $PA^V(g)$. Definimos anteriormente o problema auxiliar 1

$$\begin{array}{rl} Z^* = max & \theta(g) \\ s.a & g \in G \end{array}$$

onde $\theta(g) \quad = \quad valor \quad ótimo \quad de \quad PA^V(g)$.

Vimos na seção anterior que o problema auxiliar 1 é equivalente a PL. Equivalente no sentido que a solução de um dos problemas permite obter a solução do outro.

Vamos supor que a hipótese 4 seja satisfeita. Então a função θ é continuamente diferenciável (ver seção anterior) num subconjunto de G, mais exatamente no conjunto

$$\widetilde{G} = \left\{ g \in \mathbb{R}^n_{++} : \quad \sum_{j \in S_k} g_j = 1 \quad \forall k = 1, \ldots, K \right\}.$$

Seja $\epsilon > 0$ suficientemente pequeno e

$$G_\epsilon = \left\{ g \in \mathbb{R}^n : g_j \geq \epsilon \quad \forall j \quad e \quad \sum_{j \in S_k} g_j = 1 \quad \forall k = 1, \ldots, K \right\}. \tag{2.54}$$

É fácil provar que G_ϵ é fechado, convexo e compacto. A função objetivo do problema auxiliar 1 é continuamente diferenciável em G_ϵ. Se mudarmos o domínio da função θ no problema auxiliar temos outro problema que chamaremos **problema auxiliar 2**.

$$\begin{array}{rl} max & \theta(g) \\ s.a & g \in G_\epsilon. \end{array}$$

Tomaremos ϵ suficientemente pequeno em G_ϵ de tal forma que $G_\epsilon \approx G$. Com a finalidade de achar uma solução aproximada de (PL), aplicaremos o método do gradiente condicional no problema auxiliar 2. Então resolveremos

$$\begin{aligned} max \quad & \nabla\theta(g^{(r)})'(g - g^{(r)}) \\ s.a \quad & g \in G_\epsilon \end{aligned} \tag{2.55}$$

Este problema pode ser dividido em K sub-problemas independentes e a solução de cada um deles é imediata. A solução do k-ésimo sub-problema, denotada por $(\bar{g}^{(r)})^k$ tem todas as coordenadas iguais a ϵ exceto uma coordenada particular, que deve ser igual a $1 - (n_k - 1)\epsilon$, onde n_k é a cardinalidade de S_k. Essa coordenada particular corresponde ao maior valor da derivada parcial em S_k, ou seja a

$$max_{j \in S_k} \frac{\partial \theta}{\partial g_j}(g^{(r)})$$

Descreveremos uma iteração do método do gradiente condicional.

Dado um $g^{(r)} \in G_\epsilon$ da iteração previa,

1) Resolva o problema agregado $PA^V(g^{(r)})$, obtendo assim as soluções $X^{g^{(r)}}$ e $U^{g^{(r)}}$.

2) Calcule $\nabla\theta(g^{(r)})$. Resolva o problema (2.55), ou seja, os K sub-problemas independentes. Seja $\bar{g}^{(r)}$ a solução de (2.55). Se o valor ótimo de (2.55) é zero, pare, caso contrário, vá para o passo 3) com $g = \bar{g}^{(r)} - g^{(r)}$.

3) Calcule o tamanho de passo α^k com a regra de Armijo e volte para o passo 1).

O passo crítico é o passo 3) já que para escolher o próximo $g^{(r+1)}$ devemos construir e resolver vários problemas agregados até encontrá-lo (ver 2.51).

2.4 Estratégia de agregação iterativa

Na seção anterior apresentamos um método iterativo teórico que permitia melhorar o peso e encontrar uma solução aproximada do problema PL, usando o gradiente do Lagrangeano. Sua utilização entretanto nos leva a dois problemas: o primeiro porque é difícil saber quando a hipótese 4 é satisfeita. Em outras palavras não sabemos quando vão ser satisfeitas as condições para que a função objetivo do problema auxiliar seja continuamente diferenciável em $\widetilde{G}$. Em segundo lugar é complicado achar o tamanho de passo adequado para garantir crescimento da função objetivo e convergência para pontos estacionários.

Baseando-se na idéia de melhoramento de pesos descrita, considerando o gradiente do Lagrangeano, apresentamos nesta seção outra atualização de pesos usado por K. Jörnstein, R. Leisten e S. Storoy [12]. Implementaremos este método e daremos finalmente os resultados de alguns exemplos numéricos. Inicialmente descreveremos a estrutura dos problemas aos quais aplicaremos o método.

Descrição dos problemas

Testamos o procedimento de agregação iterativa em problemas do tipo

$$(\mathrm{PL})\left\{\begin{array}{rl} \mathrm{Z}^* = & max \quad cx \\ & s.a \quad Ax \leq b \\ & \qquad\quad x \leq \mathbf{1_n} \\ & \qquad\quad x \geq 0 \end{array}\right.$$

onde $\mathbf{1_n}$ denota um vetor de $\mathbb{R}^n$ com todas as componentes iguais a 1. Consideramos $b \geq 0$ para garantir viabilidade de cada problema agregado. Formamos o problema agregado notando que a matriz de restrições é da forma

$$\begin{bmatrix} A \\ I_n \end{bmatrix}.$$

Assim se

$$\mathrm{D} = \begin{bmatrix} A \\ I_n \end{bmatrix}$$

e

$$\mathrm{e} = \begin{bmatrix} b \\ \mathbf{1_n} \end{bmatrix}$$

podemos escrever o PL como

$$(\mathrm{PL})\left\{\begin{array}{rl} Z^* = & max \quad cx \\ & s.a \quad Dx \leq e \\ & \qquad\quad x \geq 0. \end{array}\right.$$

Devido às restrições

$$x_i \leq 1 \quad i = 1, \ldots, n$$

no PL, quando agregamos as colunas com índices em cada conjunto S_k da partição δ, obtemos as seguintes restrições para a variável X_k

$$g^k_{j_1} X_k \leq 1$$

$$g^k_{j_2} X_k \leq 1$$

$$\vdots$$

$$g^k_{j_{n_k}} X_k \leq 1$$

que podem ser substituídas por

$$\left(\max_{j \in S_k} g^k_j\right) X_k \leq 1.$$

A seguir, vamos descrever o procedimento e posteriormente detalha-lo. O procedimento tem a seguinte estrutura geral:

- 1) **Inicialização.**
  t=0
  Tomamos uma partição δ (fixa durante todo o processo) arbitrária dos índices das variáveis $\{1, \ldots, n\}$. Inicializamos cada sub-vetor $(g^{(0)})^k$, para k=1,...,K.

- 2) **Construção e resolução do problema agregado.**
  Usamos o peso corrente para construir e resolver PA^V e achamos $X^{(t)}$ e $U^{(t)}$ (soluções ótimas na iteração t, primal e dual respectivamente).

- 3) **Desagregação.**
  Para cada grupo da partição achamos a solução desagregada de peso fixo que está definida como $(x^{(t)})^k = X^{(t)}_k (g^{(t)})^k$ (subseção 1.2.2).

- 4) **Atualização do peso.**
  Mudamos o peso do problema agregado anterior e incrementamos a iteração t:= t+1.

- 5) **Critério de parada.**
  Se algum critério de parada é satisfeito paramos. Caso contrário voltamos ao passo 2.

Vamos agora apresentar em forma mais detalhada este procedimento.

Descrição do procedimento

Como mencionamos anteriormente, a partição δ usada para formar o problema agregado permanece fixa durante todas as iterações. Os conjuntos da

partição foram formados por nossa escolha, segundo a ordem crescente dos índices das colunas. Consideramos grupos de 5 e 10 colunas. Denotaremos por R o numero de elementos em cada grupo.

- Inicialização
  caso 1: R=5.
  Para cada problema, escolhemos arbitrariamente 3 pesos diferentes para inicializar este procedimento de agregação iterativa. Estes pesos foram
  peso 1: (1/5, 1/5, 1/5, 1/5, 1/5)
  peso 2: (1/10, 3/10, 2/10, 3/10, 1/10)
  peso 3: (3/10, 1/10, 2/10, 1/10, 3/10)

  caso 2: R=10.
  Aqui inicializamos com
  peso 4: (1/10,...,1/10)

- Construção e resolução do problema agregado
  Encontramos as soluções $X^{(t)}$, $U^{(t)}$ para o $PA^V(g^{(t)})$ e seu dual respectivamente, usando as rotinas do CPLEX.

- Desagregação
  A partir da solução do problema agregado, achamos a solução desagregada de peso fixo, como indicado na subseção 1.2.2.

- Atualização do peso
  Este é o passo crítico do procedimento. Primeiro daremos a notação a ser usada.

  L = Lagrangeano associado ao problema agregado $PA^V(g^{(t)})$. ver (2.18).

$$\gamma = \frac{\partial L}{\partial g}/_{(X^{(t)},U^{(t)},g^{(t)})}$$

$$Se \quad j \in S_k, \qquad \gamma_j = (c_j - U^{(t)}A^j)X_k^{(t)}$$

  n_k = número de elementos do grupo S_k.

  r_j = limitante superior da variável x_j.

α_t : parâmetro escolhido como usualmente no método do sub-gradiente ($\alpha_t \to 0$ quando $t \to \infty$).

UB_t : melhor limitante superior do valor ótimo do PL obtido até iteração t. Em cada iteração calculamos o limitante superior como na proposição 1.2.9.

LB_t: limitante inferior gerado na iteração t, ou seja igual ao valor ótimo de $PA^V(g^{(t)})$.

A atualização do peso foi feita usando

$$g_{j,1} = g_j^{(t)} + fator * \gamma_j \tag{2.56}$$

onde o novo peso $g^{(t+1)}$ vai ser obtido por meio de "projeção". Como as componentes do peso são limitadas, no lugar de aplicar diretamente a projeção sobre o produto cartesiano de simplexos, projetamos primeiro $g_{j,1}$ no intervalo $[0,1]$

$$g_{j,2} = max(0, min(1, g_{j,1})) \tag{2.57}$$

e depois projetamos o sub-vetor $g_{j,2}^k$ sobre seu correspondente simplexo, obtendo

$$g_j^{(t+1)} = \frac{g_{j,2}}{\sum_{j \in S_k} g_{j,2}} \tag{2.58}$$

para cada $j \in S_k$.

Atualizamos o peso utilizando (2.56), onde o fator é um dos dois multiplicadores. Na primeira escolhemos

$$fator1 = \frac{\alpha_t}{\| \gamma \|^2} \frac{r_j}{n_k} (UB_t - LB_t)$$

e na segunda

$$fator2 = \frac{\alpha_t r_j}{\| \gamma \|^2} (UB_t - LB_t)$$

- Critério de Parada.
  Consideramos um número fixo de iterações, em nosso caso utilizamos 80.

Esta estratégia de agregação iterativa, não garante acréscimo da função objetivo de uma iteração para outra, nem muito menos convergência da seqüência $g^{(t)}$ para algum máximo global do problema auxiliar. Só podemos ter certeza que em cada iteração vamos encontrar pesos, com os quais formaremos diferentes problemas agregados e cuja solução ótima vai nos permitir obter soluções viáveis para o problema inicial PL. Finalmente escolheremos a melhor destas soluções.

Observações gerais

1) Para todos os exemplos consideramos um número fixo de iterações (80).
2) As variáveis foram agrupadas de acordo com a ordem crescente de seus índices. Consideramos grupos de 5 e 10 colunas.
3) O limitante superior 1 de todas as variáveis em cada grupo do (PL) foi reduzido a só um limitante superior agregado. Seeste valor não advem de uma única restrição, seu valor dual agregado é distribuido igualmente entre todas as variáveis relevantes.
5) O limitante superior do valor ótimo do PL em cada iteração foi calculado como na proposição 1.2.9 .
6) Não estamos usando a projeção sobre G, mas uma 'projeção grosseira', computacionalmente mais simples de ser implementada.

Fonte

Os exemplos utilizados para fazer os testes foram tomados de http:www.zib.de.

Resultados computacionais

Os testes foram feitos tendo em conta que para o caso 1(2), temos 3(1) pesos iniciais e duas formas de atualização do peso usando os diferentes multiplicadores do gradiente do Lagrangeano, mencionado acima. Obtendo em cada exemplo 6 resultados para o caso 1 e 2 resultados para o caso 2. Estes aparecem nas tabelas abaixo. As colunas da tabela contêm limitante inferior dividido por o valor ótimo do PL (L.inf), limitante superior dividido por o valor ótimo do PL (L.sup), peso utilizado (peso) e fator escolhido (fator1), (fator2). Nas duas últimas colunas, um asterisco indicará a utilização do fator1 ou fator2.

Exemplo 1

m = 97

n = 1989

densidade = 5.14%

L.inf	L.sup	peso	fator 1	fator 2
0.9645	1.3214	1	*	
0.9645	1.2751	1		*
0.9598	1.3844	2	*	
0.9717	1.3247	2		*
0.9619	1.3845	3	*	
0.9568	1.4128	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.9165	1.6545	4	*	
0.9270	1.3858	4		*

Exemplo 2

m = 2054

n = 10724

densidade = 0.18%

L.inf	L.sup	peso	fator 1	fator 2
0.9856	1.0903	1	*	
0.9864	1.0896	1		*
0.9522	1.1041	2	*	
0.9659	1.076	2		*
0.9527	1.082	3	*	
0.9680	1.076	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.9780	1.2347	4	*	
0.9814	1.1442	4		*

Exemplo 3

m = 146
n = 2655
densidade = 2.89%

L.inf	L.sup	peso	fator 1	fator 2
0.9634	1.364	1	*	
0.9719	1.2032	1		*
0.9567	1.3300	2	*	
0.9755	1.2998	2		*
0.9557	1.2977	3	*	
0.969	1.1970	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.8935	1.4920	4	*	
0.9250	1.5273	4		*

Exemplo 4

m = 230
n = 2025
densidade = 2.6%

L.inf	L.sup	peso	fator 1	fator 2
0.9604	1.499	1	*	
0.9742	1.3039	1		*
0.9308	1.8082	2	*	
0.9398	1.5286	2		*
0.9472	1.946	3	*	
0.9730	1.4344	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.8990	3.1414	4	*	
0.895	2.8987	4		*

Exemplo 5

m = 408

n = 7195

densidade = 1.72%

L.inf	L.sup	peso	fator 1	fator 2
0.7079	3.0875	1	*	
0.74945	2.7517	1		*
0.6884	3.2001	2	*	
0.7367	2.6818	2		*
0.6929	3.0718	3	*	
0.7339	2.9075	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.5237	5.2998	4	*	
0.5975	4.130	4		*

Exemplo 6

m = 755

n = 2756

densidade = 0.42%

L.inf	L.sup	peso	fator 1	fator 2
0.8322	1.3675	1	*	
0.8322	1.4051	1		*
0.7662	1.3559	2	*	
0.6160	1.3559	2		*
0.7412	1.3606	3	*	
0.6232	1.3606	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.6120	1.2936	4	*	
0.6120	1.3045	4		*

Exemplo 7
m = 1657
n = 10724
densidade = 0.2%

L.inf	L.sup	peso	fator 1	fator 2
0.9805	1.019	1	*	
0.9854	1.019	1		*
0.9669	1.0525	2	*	
0.9873	1.0269	2		*
0.9619	1.0197	3	*	
0.9790	1.0175	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.9438	1.072	4	*	
0.9526	1.067	4		*

Exemplo 8
m = 782
n = 8904
densidade = 1.008%

L.inf	L.sup	peso	fator 1	fator 2
0.7553	2.8480	1	*	
0.7894	2.5110	1		*
0.7263	2.9095	2	*	
0.7905	2.6532	2		*
0.7587	2.9038	3	*	
0.7923	2.6143	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.6067	5.057	4	*	
0.675	3.8074	4		*

Exemplo 9

m = 763

n = 8572

densidade = 1.033%

L.inf	L.sup	peso	fator 1	fator 2
0.7381	2.70	1	*	
0.7790	2.572	1		*
0.7399	2.7809	2	*	
0.7868	2.6617	2		*
0.7463	2.9531	3	*	
0.8022	2.6508	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.5787	4.6783	4	*	
0.6438	3.8031	4		*

Exemplo 10

m = 823

n = 8904

densidade = 0.995%

L.inf	L.sup	peso	fator 1	fator 2
0.7546	2.8535	1	*	
0.7923	2.5298	1		*
0.7380	2.7740	2	*	
0.7942	2.2979	2		*
0.7544	2.8424	3	*	
0.7773	2.3845	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.6018	4.9269	4	*	
0.6621	4.2154	4		*

Exemplo 11

m = 50

n = 6774

densidade = 18.17%

L.inf	L.sup	peso	fator 1	fator 2
0.1766	1.2248	1	*	
0.1590	1.2746	1		*
0.1847	1.156	2	*	
0.1976	1.1382	2		*
0.1766	1.2737	3	*	
0.1910	1.1757	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.1491	2.075	4	*	
0.2017	1.7551	4		*

Exemplo 12

m = 124

n = 10757

densidade = 6.82%

L.inf	L.sup	peso	fator 1	fator 2
0.1466	1.3599	1	*	
0.1777	1.2296	1		*
0.1708	1.2479	2	*	
0.1955	1.3397	2		*
0.1409	1.4022	3	*	
0.1538	1.3506	3		*

L.inf	L.sup	peso	fator 1	fator 2
0.1194	2.215	4	*	
0.1094	2.025	4		*

A tabela a seguir mostra a melhor aproximação inferior e a melhor aproximação superior do valor ótimo do PL obtida em cada exemplo, considerando os 3 pesos iniciais diferentes e as duas maneiras de atualização do peso. Aqui o número de elementos por grupo é 5.

Linhas	Colunas	R	Densidade	Max.lim.inf.	Min.lim.sup
97	1989	5	5.14	0.9645	1.2751
2054	10724	5	0.18	0.9864	1.076
146	2655	5	2.89	0.9755	1.1970
230	2025	5	2.6	0.9742	1.3039
408	7195	5	1.72	0.7495	2.6818
755	2756	5	0.42	0.8322	1.3559
1657	10724	5	0.2	0.9873	1.019
782	8904	5	1.008	0.7923	2.5110
763	8572	5	1.033	0.8022	2.572
823	8904	5	0.995	0.7942	2.2979
50	6774	5	18.17	0.1976	1.1382
124	10757	5	6.82	0.1955	1.2296

A tabela a seguir mostra a melhor aproximação inferior e a melhor aproximação superior do valor ótimo do PL obtida em cada exemplo, considerando um pesos inicial(peso 4) e as duas maneiras de atualização do peso. Aqui o número de elementos por grupo é 10.

Linhas	Colunas	R	Densidade	Max.lim.inf.	Min.lim.sup
97	1989	10	5.14	0.9270	1.3858
2054	10724	10	0.18	0.9814	1.1442
146	2655	10	2.89	0.9250	1.4920
230	2025	10	2.6	0.8990	2.8987
408	7195	10	1.72	0.5975	4.130
755	2756	10	0.42	0.6120	1.2963
1657	10724	10	0.2	0.9526	1.067
782	8904	10	1.008	0.675	3.8074
763	8572	10	1.033	0.6438	3.8031
823	8904	10	0.995	0.6621	4.2154
50	6774	10	18.17	0.2017	1.7551
124	10757	10	6.82	0.1194	2.025

Observamos nas 2 últimas tabelas que quando o grau de agregação aumenta, a aproximação é menor.

Em ambos casos o limitante superior esta mais longe do valor esperado.

Capítulo 3

Agregação e Computação de Limitantes em Programação Inteira

3.1 Introdução

Neste capítulo aplicamos o método de agregação de variáveis para encontrar limitantes superiores do valor ótimo de problemas de programação linear inteira (PI) *multiknapsack*.

Apresentamos o trabalho de Litvinchev, I.S, e Rangel S.[17] e fazemos uma pequena variante para encontrar outro limitante para o valor ótimo do PI. Damos alguns resultados com alguns exemplos numéricos.

É comum obter limitantes para os problemas acima mencionados resolvendo seus respectivos problemas relaxados, mas aqui resolveremos problemas agregados do relaxado.

Inicialmente vemos outra forma de achar um limitante superior para o valor ótimo Z^* de um problema linear, usando a solução dual agregada e um conjunto W fechado, convexo e limitado que contém a solução ótima x^*. Na seção 3 obtemos W usando desigualdades válidas para o conjunto de pontos viáveis do problema *multiknapsack*. Por um resultado de Balas é possível obter essas desigualdades válidas usando recobrimentos minimais.

3.2 Outro limitante para Z^*

Dado o problema linear original

$$(PL)\left\{\begin{array}{lll} Z^* = & & max \quad cx \\ & s.a & Ax \leq b \\ & & x \geq 0 \end{array}\right.$$

onde $A \in \mathbb{R}^{m \times n}$.

Seja W um conjunto convexo, fechado e limitado em $\mathbb{R}^n$, tal que $x^* \in W$, onde x^* é uma solução ótima de PL. Podemos obter o conjunto W por manipulação de restrições do problema original o que é mais importante, este conjunto pode ser construído depois que o problema agregado tenha sido resolvido. De fato, se conhecemos o conjunto W, podemos acrescentar a PL a condição $x \in W$, sem que isto modifique a solução ótima.

Como vimos no capítulo 1, para formar o problema agregado de PL, precisamos de uma partição $\sigma = \{S_k : k = 1, \ldots, K\}$ dos índices das colunas $\{1, \ldots, n\}$ e também de um peso $g = \{g^k : k = 1, \ldots, K\}$ (ver definições 1.2.1 e 1.2.2). O problema agregado resultante é

$$(PA^V)\left\{\begin{array}{lll} Z^V = & & max \quad \bar{c}X \\ & s.a & \bar{A}X \leq b \\ & & X \geq 0 \end{array}\right.$$

onde $X = (X_1, \ldots, X_K)$.

Suponha que $\bar{X}$ e $\bar{U}$ sejam as soluções ótimas de PA^V e seu dual respectivamente.

O problema obtido agregando colunas pode ser considerado uma restrição do problema original, no sentido que a solução agregada $\bar{X}$ pode ser desagregada (ver subseção 1.2.2) para obter uma solução viável $\bar{x}$ para PL. A forma mais simples consiste em usar desagregação de peso fixo. No Corolário 1.2.4, demonstramos que se verifica a seguinte relação

$$Z^V = c\bar{x} \leq Z^*$$

o qual nos fornece um limitante inferior para o valor ótimo Z^*. Nesta seção obteremos um limitante superior para Z^*, diferente dos já apresentados na

seção 2 do capítulo 1.

Usando multiplicadores de Lagrange no PL adicionando a condição $x \in W$ temos

$$\begin{aligned} Z^* &= max \quad \{cx : Ax \leq b, x \geq 0, x \in W\} \\ &= max_{x \in W, x \geq 0}\Big\{Inf_{u \geq 0}(cx + u(b - Ax))\Big\} \\ &\leq max_{x \in W, x \geq 0}(cx + \bar{U}(b - Ax)). \end{aligned} \tag{3.1}$$

e pelo teorema forte de dualidade, temos $\bar{U}b = \bar{c}\bar{X} = Z^V$. Logo em (3.1) temos

$$Z^* \leq Z^V + max_{x \in W, x \geq 0}\Big((c - \bar{U}A)x\Big) \tag{3.2}$$

3.3 Limitantes para problemas de programação inteira identificando desigualdades válidas

Notação

- I_n : matriz identidade de ordem n.
- $\mathbf{1_n}$: vetor em $\mathbb{R}^n$ com componentes iguais a um.

Sejam c um vetor em $\mathbb{R}^n_+$, b um vetor em $\mathbb{R}^m_+$, A uma matriz em $\mathbb{R}^{m \times n}_+$ e todos com coeficientes inteiros. O problema que será abordado nesta seção é o seguinte:

$$(PI)\left\{\begin{array}{rrl} \Gamma^* = & max & cx \\ & s.a & Ax \leq b \\ & & x \geq 0 \\ & & x_i \in \{0,1\}, \quad i = 1, \ldots, n \end{array}\right.$$

É comum obter um limitante superior para Γ^*, resolvendo o seguinte problema

$$(PR)\left\{\begin{array}{rrl} \Gamma_R = & max & cx \\ & s.a & Ax \leq b, \\ & & I_n x \leq \mathbf{1_n}, \\ & & x \geq 0. \end{array}\right.$$

onde não se toma em conta as condições das variáveis serem inteiras.

É óbvio que

$$\Gamma^* \quad \leq \quad \Gamma_R.$$

Consideremos agora o problema agregado de PR, denotado por $\overline{PR}$

$$(\overline{PR}) \begin{cases} \overline{\Gamma_R} = \quad max \quad \bar{c}X \\ \quad sujeito \quad a \quad \overline{A}X \leq b \\ \qquad\qquad 0 \leq X_k \leq \eta_k. \quad k = 1, \ldots, K \\ \\ \quad onde \quad \eta_k = \frac{1}{\max_{j \in S_k} g_j^k} \end{cases}$$

Definição 3.3.1 *Seja P um conjunto em $\mathbb{R}^n$. Diremos que a desigualdade*

$$\sum_{j=1}^{n} \alpha_j x_j \quad \leq \quad \beta$$

é válida para P se todos os pontos de P satisfazem essa desigualdade.

□

Definição 3.3.2 *Dada a restrição*

$$\sum_{j=1}^{n} \alpha_j x_j \quad \leq \quad \beta \qquad com \quad x_j \in \{0, 1\}.$$

Um **recobrimento minimal** *Q desta restrição é um subconjunto de $\{1, \ldots, n\}$ tal que*

- $\sum_{j \in Q} \alpha_j > \beta$ *e*
- $\sum_{j \in S} \alpha_j \quad \leq \quad \beta, \quad \forall S \subsetneq Q$

□

Definição 3.3.3 *A* **extensão** $E(Q)$ **de um recobrimento minimal** *é o conjunto*

$$E(Q) \quad = \quad Q \cup \left\{ j' \in \{1, \ldots, n\} - Q \quad : \alpha_{j'} \geq \alpha_j, \quad \forall j \in Q \right\}$$

□

Encontraremos um limitante superior para Γ^* resolvendo não um problema relaxado, mas um problema obtido por agregação de colunas do problema relaxado PR. A idéia principal é construir W usando desigualdades válidas para o conjunto de pontos viáveis de PI.

Uma das formas para obter essa desigualdade válida é usando um recobrimento minimal (ver definição 3.3.2). Egon Balas e Robert Jeroslow estudaram algumas propriedades do hipercubo unitário n-dimensional (ver [2]). Apresentaremos um de seus resultados. Antes indicaremos a notação a ser usada.

Notação

- K : hipercubo unitário n-dimensionai.

$$K = \{x \in \mathbb{R}^n : \quad 0 \leq x_j \leq 1, \quad j \in N\} \tag{3.3}$$

$$N = \{1, \ldots, n\}$$

- V : conjunto dos vértices de K.
  v é um vértice de K, se cada componente v_j tem valor 0 ou 1..

Teorema 3.3.4 *Seja $\alpha_j \geq 0$, para j=1,...,n. O ponto $x \in V$ satisfaz*

$$\sum_{j=1}^{n} \alpha_j x_j \leq \beta \tag{3.4}$$

se e só se satisfaz

$$\sum_{j \in E(Q)} x_j \leq |Q| - 1 \tag{3.5}$$

$\forall Q \in \Lambda$, onde Λ é a familia de todos os recobrimentos minimais da desigualdade 3.4.

PROVA
Suponha que existe $x^o \in V$, tal que algum $S_o \in \Lambda$ não obedece (3.5). Ou seja

$$\sum_{j \in E(S_o)} x_j^o > |S_o| - 1. \tag{3.6}$$

Logo definindo $N^+ = \{j \in N : x_j^o = 1\}$ e denotando

$$T_1 \quad = \quad N^+ \cap (E(S_o) - S_o) \quad e \quad T_2 \quad = \quad N^+ \cap S_o$$

temos que

$$T_1 \cap T_2 = \emptyset.$$

Seja $T = T_1 \cup T_2$, por (3.6) segue

$$|T_1| + |T_2| = |T| \geq |S_o|$$

logo

$$|T_1| \geq |S_o| - |T_2| = |S_o - T_2| \tag{3.7}$$

Por outro lado da definição de $E(S_o)$ e T_1 temos

$$(j \in T_1, h \in S_o) \Longrightarrow (j \in E(S_o) - S_o, h \in S_o) \Longrightarrow (\alpha_j \geq \alpha_h) \tag{3.8}$$

e em conseqüência

$$\begin{aligned}
\sum_{j \in T} \alpha_j &= \sum_{j \in T_1} \alpha_j + \sum_{j \in T_2} \alpha_j \\
&\geq \sum_{j \in S_o - T_2} \alpha_j \quad + \quad \sum_{j \in T_2} \alpha_j \quad (por \quad 3.7 \quad e \quad 3.8) \\
&= \sum_{j \in S_o} \alpha_j \\
&> \beta \quad (porque \quad S_o \in \Lambda)
\end{aligned} \tag{3.9}$$

Portanto $\sum_{j \in T} \alpha_j x_j^o > \beta$ e

$$\sum_{j=1}^{n} \alpha_j x_j^o \quad > \quad \beta$$

não obedecendo (3.4).

Reciprocamente vamos supor que $x^o \in V$ não satisfaça (3.4). Ou seja

$$\sum_{j=1}^{n} \alpha_j x_j^o = \sum_{j \in N^+} \alpha_j > \beta \tag{3.10}$$

onde N^+ é definido acima. Pela desigualdade (3.10) podemos afirmar que existe um conjunto $S \subset N^+$ que é um recobrimento minimal de (3.4) e podemos encontrar uma extensão $E(S)$ deste recobrimento minimal. Logo

$$\sum_{j \in E(S)} x_j^o \geq \sum_{j \in S} x_j^o = |S| > |S| - 1$$

Portanto (3.5) não é satisfeita.

■

Observação 3.3.1 *Suponha que temos a i-ésima restrição do problema PI*

$$\sum_{j=1}^{n} A_{ij} x_j \leq b_i. \tag{3.11}$$

Se x pertence ao conjunto de soluções viáveis do PI, então $x \in V$ e em particular satisfaz a restrição (3.11). *Logo pelo teorema anterior, x também satisfaz*

$$\sum_{j \in E(Q)} x_j \leq |Q| - 1 \tag{3.12}$$

para todo recobrimento minimal Q da desigualdade (3.11).

O seguinte teorema vai nos permitir obter um recobrimento minimal estendido de (3.11) tal que (3.12) corte um determinado ponto $y \in \mathbb{R}^n$.

Teorema 3.3.5 *Dado um ponto $y \in K$ e o seguinte problema*

$$(P) \begin{cases} \epsilon = & \min \quad \sum_{j=1}^{n} (1 - y_j) s_j \\ & s.a \quad \sum_{j=1}^{n} \alpha_j s_j > \beta \\ & \qquad s_j \in \{0, 1\}, \quad j = 1, \ldots, n. \end{cases}$$

Se $\epsilon < 1$, existe um recobrimento minimal S da seguinte desigualdade

$$\sum_{j=1}^{n} \alpha_j x_j \leq \beta \tag{3.13}$$

tal que

$$\sum_{j \in E(S)} x_j \leq |S| - 1 \tag{3.14}$$

corte o ponto y.

PROVA

Seja $s^* = (s_1^*, \ldots, s_n^*)$ a solução ótima de P. Seja

$$S^+ = \{j \in N : s_j^* = 1\}.$$

Da viabilidade de s^* para o problema P temos

$$\beta < \sum_{j \in N} \alpha_j s_j^* = \sum_{j \in S^+} \alpha_j s_j^* = \sum_{j \in S^+} \alpha_j.$$

Desta desigualdade é claro que existe um conjunto que é um recobrimento minimal para a desigualdade (3.13). Ou seja existe um conjunto $S \subset S^+$, tal que

$$\sum_{j \in S} \alpha_j > \beta \quad e \quad \sum_{j \in R} \alpha_j \leq \beta, \quad \forall R \subsetneq S$$

Agora demonstraremos que (3.14) corta y. Do valor ótimo de P, temos

$$\sum_{j \in S} (1 - y_j) s_j^* \leq \sum_{j=1}^{n} (1 - y_j) s_j^* = \epsilon < 1$$

logo

$$\begin{aligned} \sum_{j \in S} y_j s_j^* &> \sum_{j \in S} s_j^* - 1 \\ &= |S| - 1 \end{aligned} \tag{3.15}$$

Por outro lado

$$\sum_{j \in S} y_j s_j^* = \sum_{j \in S} y_j \leq \sum_{j \in E(S)} y_j \tag{3.16}$$

Logo de (3.15) e (3.16) temos

$$\sum_{j \in E(S)} y_j > |S| - 1.$$

Portanto y não satisfaz (3.14).

■.

Escolha de W

Sejam $\bar{X}$ e $\bar{U}$ soluções primal e dual de $(\overline{PR})$. Se no problema (P) do Teorema 3.3.5 tomemos y igual a $\bar{x}$ (solução obtida por desagregação de peso fixo de $\bar{X}$ e a restrição de (P) como

$$\sum_{j=1}^{n} A_{ij} s_j > b_i,$$

então pelo teorema 3.3.5, se o valor ótimo de (P) é menor que 1, existe um recobrimento minimal S da i-ésima restrição de (PI), tal que a desigualdade

$$\sum_{j \in E(S)} x_j \leq |S| - 1$$

corta $\bar{x}$. O ideal seria que $\bar{x}$ estivesse suficientemente próxima da solução ótima de PR, pois assim, provavelmente, a desigualdade anterior também cortaria a solução ótima do PR. O acréscimo desta restrição ao problema PR permitiria uma diminução de seu valor ótimo. Analogamente o valor ótimo de $\overline{PR}_w$ sería menor que o valor ótimo de $\overline{PR}$ o que acarretará uma possível melhoria do limitante superior.

Vamos definir W como

$$W = \{x \in K : \sum_{j \in E(S)} x_j \leq |S| - 1\} \tag{3.17}$$

Este conjunto tem as seguintes propriedades:
a) É limitado, fechado e convexo.
b) Todas as soluções viáveis para (PI) pertencem a W (observação 3.3.1).
c) A solução desagregada $\bar{x}$ não esta em W (Teorema 3.3.5).

Com o objetivo de usar o resultado mencionado na primeira seção, vamos a reescrever os problemas PI, PR e $\overline{PR}$, acrescentando a condição de que $x \in W$. A notação para estes novos problemas será respectivamente PI_w, PR_w e $\overline{PR}_w$. A desigualdade usada para definir o conjunto W, aparecerá nos problemas PI_w e PR_w como

$$A_{m+1} x \leq b_{m+1}$$

e no problema $\overline{PR}_w$ aparecerá como

$$\overline{A}_{m+1}X \leq b_{m+1}.$$

Seja

$$PI_w \begin{cases} \Gamma^* = & max \quad cx \\ & s.a \quad Ax \leq b \\ & A_{m+1}x \leq b_{m+1} \\ & I_n x \leq \mathbf{1_n} \\ & x_i \geq 0, x_i \in \mathbb{Z} \quad i = 1, \ldots, n \end{cases}$$

$$PR_w \begin{cases} \Gamma_{Rw} = & max \quad cx \\ & s.a \quad Ax \leq b \\ & A_{m+1}x \leq b_{m+1} \\ & I_n x \leq \mathbf{1_n} \\ & x_i \geq 0, \quad i = 1, \ldots, n \end{cases}$$

$$\overline{PR}_w \begin{cases} \overline{\Gamma_{Rw}} = & max \quad \bar{c}X \\ & s.a \quad \overline{A}X \leq b \\ & \overline{A}_{m+1}X \leq b_{m+1} \\ & 0 \leq X_k \leq \eta_k. \quad k = 1, \ldots, K \\ & onde \quad \eta_k = \frac{1}{\max_{j \in S_k} g_j^k} \end{cases}$$

Observação 3.3.2 *Pela observação* 3.3.1 e a escolha de W, a solução ótima de PI é igual a solução ótima de PI_w.

Apresentaremos em seguida dois limitantes para Γ^*, seguindo a idéia usada para obter (3.2). Seja $u \in \mathbb{R}^{m+1}$, $u = (u^1, u^2)$, então no problema PR_w temos

$$\begin{aligned} \Gamma_{Rw} &= max \quad \{cx : Ax \leq b, A_{m+1}x \leq b_{m+1}, x \in W\} \\ &= \max_{x \in W} \left\{ Inf_{u \geq 0}(cx + u^1(b - Ax) + u^2(b_{m+1} - A_{m+1}x)) \right\} \end{aligned} \tag{3.18}$$

- **Limitante 1:** Para obter este limitante, assumimos que o problema $\overline{PR}_w$ este resolvido, portanto sua solução dual é conhecida. Seja $v =$

(v^1, v^2) sub-vetor da solução dual, correspondente as primeiras $m+1$ restrições de $\overline{PR}_w$. Se tomamos em (3.18) $u = (v^1, v^2)$, então temos

$$\begin{aligned}
\Gamma_{R_w} &= \max_{x \in W} \left\{ cx + v^1(b - Ax) + v^2(b_{m+1} - A_{m+1}x) \right\} \\
&= v^1 b + v^2 b_{m+1} + \max_{x \in W} \left\{ cx - v^1 Ax - v^2 A_{m+1} x \right\} \\
&\leq \overline{\Gamma_{R_w}} + \max_{x \in W} \left\{ (c - v^1 A - v^2 A_{m+1})x \right\}
\end{aligned} \tag{3.19}$$

Portanto, como é claro que

$$\Gamma^* \leq \Gamma_{R_w}$$

então desta desigualdade e 3.19 temos

$$\Gamma^* \leq \overline{\Gamma_{R_w}} + \max_{x \in W} \left\{ (c - v^1 A - v^2 A_{m+1})x \right\} \tag{3.20}$$

- **Limitante 2:** A obtenção deste limitante não pressupõe a solução de $\overline{PR}_W$ mas supõe que tenhamos resolvido o problema agregado $\overline{PR}$. Seja U sub-vetor da solução dual de $\overline{PR}$, correspondente as primeiras m restrições. Se tomamos em (3.18) $u = (U, 0)$, então temos

$$\begin{aligned}
\Gamma_{R_w} &\leq \max_{x \in W} \left\{ cx + U(b - Ax) \right\} \\
&\leq Ub + \max_{x \in W} \left\{ (c - UA)x \right\} \\
&\leq \overline{\Gamma_R} + \max_{x \in W} \left\{ (c - UA)x \right\} \qquad (3.21) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.22)
\end{aligned}$$

Portanto como

$$\Gamma^* \leq \Gamma_{R_w},$$

pela desigualdade de acima temos

$$\Gamma^* \leq \overline{\Gamma_R} + \max_{x \in W} \{(c - UA)x\}. \tag{3.23}$$

O limitante 2 foi obtido por I.Litvinchev e S.Rangel e o limitante 1 proposto por nós.

Fonte Os exemplos utilizados para fazer os testes foram tomados de http://www.zib.de.

Antes de dar os resultados computacionais, convém ressaltar os seguintes fatos:

a) Em 40% dos exemplos tomados não foi possível aplicar o método, porque não conseguiamos fazer um corte que incluisse todas as soluções viáveis inteiras do PI, mas que deixasse fora a solução aproximada do problema relaxado, obtida por agregação.

b) O limitante superior para o valor ótimo do PI obtido resolvendo o problema relaxado foi melhor que os limitantes 1 e 2.

Resultados Computacionais

Na tabela de embaixo, R índica o número de colunnas por grupo na partição σ utilizada para agregar o (PI). Lim1 representa o limitante 1 dividido por o valor ótimo de (PI) e Lim2 representa o limitante 2 dividido por o valor ótimo de (PI).

Linhas	Colunas	R	Lim 1	Lim 2
4	28	5	1.149	1.17
4	34	5	1.122	1.121
5	50	5	1.221	1.238
5	30	5	1.182	1.2751
5	30	5	1.14	1.257
5	50	5	1.309	1.3725
5	50	5	1.268	1.286
5	50	5	1.3012	1.365
5	50	5	1.013	1.372
5	60	5	1.335	1.33
5	60	5	1.1957	1.11
5	60	5	1.1789	1.1789
5	70	5	1.3221	1.322

Observando a tabela podemos ver que para os exemplos considerados, na maioria dos casos o limitante 1 foi um pouco melhor que o limitante 2.

Capítulo 4

Conclusões

O limitante superior para o valor ótimo do PL precisa ser melhorado. Nos casos que foram testados, este valor mostrou-se relativamente distante do valor esperado, isto é, do valor ótimo do problema inicial. Sabemos teoricamente que melhores limitantes podem ser obtidos quando se usa o limitante de Mendelsohn ou limitante de Taylor, mas preferimos usar um limitante como o de Zipkin porque sua utilização é menos complexa.

Devem ser usados alguns critérios para formar os grupos (*cluster analysis*) que leve em conta algumas similaridade entre as colunas. Igualmente devemos ter critérios para inicializar o peso. No caso de agregação de restrições, existe um trabalho de Shetty e Taylor [22] que usa 'proximidade angular' para formar os grupos da partição inicial e outro critério para o peso. Esta ídea pode ser aproveitada aqui.

Para a maioria dos exemplos de programação linear inteira testados aqui, o limitante superior que propusemos foi melhor que o limitante obtido por Litvinchev e Rangel. É interessante notar que o valor ótimo do problema relaxado sem agregação forneceu nestes mesmos exemplos um valor mais próximo do valor real ótimo do problema.

Para o caso de agregação iterativa, podemos considerar várias mudanças que podem ser analisadas. Por exemplo melhorar a projeção "grosseira" que foi usada. Analisar o efeito de diferentes graus de agregação. Considerar em (2.56) diferentes multiplicadores do gradiente do Lagrangeano. Ver a relação que existe entre a densidade da matriz de restrições e a aproximação obtida

quando se aplica a estratégia de agregação iterativa.

Com o pequeno número de exemplos que utilizamos no capítulo 2 observamos que a maior grau de agregação corresponde menor aproximação.

Apêndice A

Resultados básicos de Topologia e Cálculo

Neste Apêndice mencionaremos definições e resultados básicos de Topologia e Cálculo que foram usados nos 4 Capítulos deste trabalho.

Definição A.0.6 *Uma norma* $\| \, . \, \|$ *em* $\mathbb{R}^n$ *é uma função que faz corresponder um escalar a todo* $x \in \mathbb{R}^n$ *e que tem as seguintes propriedades:*
$\| x \| \geq 0$ *para todo* $x \in \mathbb{R}^n$.
$\| cx \| = |c| \, \| x \|$ *para todo* $x \in \mathbb{R}^n$.
$\| x \| = 0$ *se e só se* $x=0$.
$\| x + y \| \leq \| x \| + \| y \|$ *para todo* $x, y \in \mathbb{R}^n$.

Exemplos de normas em $\mathbb{R}^n$

1) $\| x \|_1 = \sum_{i=1}^{n} x_i$

2) $\| x \|_2 = (x_1^2 + \ldots + x_n^2)^{1/2}$

3) $\| x \|_\infty = \max_{i=1,\ldots,n} x_i$

Definição A.0.7 *Dado* $A \subset \mathbb{R}^n$. *Diremos que* $a \in A$ *é ponto interior de* A, *se existe uma bola aberta* $B(a;r)$ *tal que* $B(a;r) \subset A$.
Denotaremos por Int(A) ao conjunto de todos os pontos interiores de A.

Definição A.0.8 *Dado o conjunto* $A \subset \mathbb{R}^n$. *Diremos que* A *é aberto se* $\forall a \in A$, *existe uma bola* $B(a;r)$ *que está contida em* A.

Definição A.0.9 *Dado o conjunto* $A \subset \mathbb{R}^n$. *Diremos que* A *é fechado se o conjunto* $\mathbb{R}^n - A$ *é aberto.*

Definição A.0.10 *Um subconjunto* $X \subset \mathbb{R}^n$ *diz-se limitado quando existe um número real* $c > 0$, *tal que* $\| x \| \leq c \quad \forall x \in X$.

Definição A.0.11 *Seja* $f : A \to \mathbb{R}$. *Diremos que* $x^* \in A$ *é um máximo de* f *em* A *se*

$$f(x) \leq f(x^*)$$

$\forall x \in A$

Propriedades do máximo
Dadas as funções f e g definidas em A e com valores em $\mathbb{R}$ e seja α um escalar positivo. Se f e g têm máximo em A, então:

1) $\max_{x \in A} \alpha f(x) = \alpha \max_{x \in A} f(x)$

2) $\max_{x \in A}(f(x) + g(x)) \leq max_{x \in A} f(x) + max_{x \in A} g(x)$

Proposição A.0.12 (Teorema de Bolzano-Weierstrass)
Toda seqüência limitada em $\mathbb{R}^n$ *possui uma sub-seqüência convergente.*

Definição A.0.13 *Seja* A *um subconjunto aberto de* $\mathbb{R}^m$.
Diremos que a função $f : A \to \mathbb{R}^n$ *é contínua no ponto* $a \in A$ *se para toda bola* $B(f(a); \epsilon)$ *existe outra bola* $B(a; \delta)$ *tal que*

$$f(B(a; \delta)) \quad \subset \quad B(f(a); \epsilon).$$

Diremos que ela é continua em A, *se é contínua em todo ponto* $a \in A$.

Definição A.0.14 *Diremos que um conjunto* $K \subset \mathbb{R}^n$ *é compacto quando ele for limitado e fechado.*

Assim, por exemplo, são compactas todas as bolas fechadas do espaço $\mathbb{R}^n$, mas o espaço $\mathbb{R}^n$ inteiro não é compacto.

Uma propriedade importante de compactos em conexão com problemas de otimização será enunciada no seguinte Teorema.

Proposição A.0.15 (**Teorema de Weierstrass**)
Toda função real contínua $f : K \to R$, definida num compacto $K \subset \mathbb{R}^n$, atínge seu máximo e seu mínimo em K, isto é, existem pontos $x_0, x_1 \in K$, tais que $f(x_0) \leq f(x) \leq f(x_1)$ para qualquer $x \in K$.

Em virtude do Teorema de Bolzano-Weierstrass, temos o seguinte resultado.

Proposição A.0.16 *Um conjunto $K \subset \mathbb{R}^n$ é compacto se, e somente se, toda seqüência $(x_k)_{k \in N} \subset K$ possui uma sub-seqüência que converge para um ponto de K.*

Derivadas

Seja $f : \mathbb{R}^n \to R$ uma função, fixemos algum $\mathbf{x} \in \mathbb{R}^n$ e consideremos a seguinte expressão

$$\lim_{\alpha \to 0} \frac{f(\mathbf{x} + \alpha \mathbf{e_i}) - \mathbf{f}(\mathbf{x})}{\alpha}$$

onde $\mathbf{e_i}$ é o vetor unitário canônico (1 na i-ésima componente e zero nas demais). Se o limite existe, ele é chamado a i-ésima derivada parcial de f no ponto $\mathbf{x}$ e é denotado por $(\partial f / \partial \mathbf{x})$ ou $\nabla_i f(\mathbf{x})$. Assumindo que todas as derivadas parciais existam, o gradiente de f em $\mathbf{x}$ é definido como o vetor

$$\nabla f(\bar{\mathbf{x}}) \quad = \quad \Big(\frac{\partial f(\bar{\mathbf{x}})}{\partial x_1}, \ldots, \frac{\partial f(\bar{\mathbf{x}})}{\partial x_n}\Big)$$

Para qualquer $\mathbf{y} \in \mathbb{R}^n$, definimos a derivada direcionai de f na direção $\mathbf{y}$, como

$$f'(\mathbf{x}; \mathbf{y}) = \lim_{\alpha \to 0+} \frac{f(\mathbf{x} + \alpha \mathbf{y}) - f(\mathbf{x})}{\alpha},$$

supondo que o limite exista.

Definição A.0.17 *A função f é Frechet diferenciável em $\mathbf{x}$ se existe um vetor $\mathbf{d}$ satisfazendo a seguinte equação*

$$\lim_{y \to 0} \frac{f(\mathbf{x} + \mathbf{y}) - f(\mathbf{x}) - \mathbf{y} \cdot \mathbf{d}}{\| \mathbf{y} \|} = 0 \tag{A.1}$$

Se um tal vetor $\mathbf{d}$ existe, podemos ver que todas as derivadas parciais $(\partial f / \partial x_i)(\mathbf{x})$ existem e que $\mathbf{d} = \nabla \mathbf{f}(\mathbf{x})$.

Note que a definição referente a diferenciabilidade de f no ponto $\mathbf{x}$ só envolve os valores de f em uma vizinhança de $\mathbf{x}$. De este modo estas definições podem ser usadas para funções que não estão definidas em todo o espaço $\mathbb{R}^n$, mas que estejam definidas em uma vizinhança do ponto onde a derivada é calculada.

Teorema A.0.18 (Teorema do valor médio) *Se $f : \mathbb{R} \to \mathbb{R}$ é continuamente diferenciável sobre um intervalo I, então $\forall x, y \in I$ existe algum $\epsilon \in [x, y]$ tal que*

$$f(y) - f(x) = \nabla f(\epsilon)(y - x).$$

Funções convexas

Agora veremos algumas propriedades de continuidade e diferenciabilidade de funções convexas.

Definição A.0.19 *Diremos que o conjunto $C \subset \mathbb{R}^n$ é convexo se para todo par de pontos x e y em C e $\lambda \in [0, 1]$ temos*

$$\lambda x + (1 - \lambda y) \in C.$$

Definição A.0.20 *Dado C um subconjunto convexo de $\mathbb{R}^n$. Uma função $f : C \to R$ é convexa se $\forall\ x, y \in C$ e $\forall \alpha \in [0, 1]$, temos*

$$f(\alpha x + (1 - \alpha)y) \quad \leq \quad \alpha f(x) + (1 - \alpha)f(y).$$

A função f é côncava se -f é convexa.

Proposição A.0.21 *Para qualquer coleção $\{C_i / i \in I\}$ de conjuntos convexos, a intersecção C_i é convexo.*

Exemplos

1) Toda função linear em $\mathbb{R}^n$ com valores reais é convexa.
2) A função norma é convexa.
3) Toda combinação linear de funções convexas, com escalares positivos é convexa.
4) Se I é um conjunto de índices, $C \subset \mathbb{R}^n$ é um conjunto convexo e $f_i : C \to \mathbb{R}$ é convexa para cada $i \in I$, então a função $h : C \to R$ definida por

$$h(x) \quad = \quad \sup_{i \in I} f_i(x)$$

é também convexa.

Para funções diferenciáveis, existe uma caracterização alternativa de convexidade, dada na seguinte proposição.

Proposição A.0.22 *Seja $C \subset \mathbb{R}^n$ um conjunto convexo e $f : C \to \mathbb{R}$ diferenciável sobre C.*
a) A função f é convexa se e só se

$$f(z) \geq f(x) + (z - x)' \bigtriangledown f(x) \quad \forall x, z \in C$$

b) Analogamente f é côncava se e só se

$$f(z) \leq f(x) + (z - x)' \bigtriangledown f(x) \quad \forall x, z \in C$$

Apêndice B

Implementações Computacionais

```
/*************************************************************************/
/*       lee o problema inicial con formato MPS e me da os vetores        */
/*       ind_A,   val_A  ( representacao de A por listas ligadas )    e    */
/*       ind_c,   val_c  ( representacao de c por listas ligadas )         */
/*************************************************************************/
void lee ( arq1)
FILE  *arq1;

{
  char  c1[M1 + 1], c2[M2 + 1], c3[M3 + 1], c4[M4 + 1], c5[M5 + 1], car[6];
  int  cont,   cont_A,  cont_c, i, j, k, w  ;
  cont = 0;
  cont_A  = 0;
  cont_c  = 0;
  k = -1;



  for ( i=0; i < VC;  i++ )
     {
       ind_c[i] =  0;
       val_c[i] =  0;
     }
```

```
for ( i=0;  i < VA; i++ )
     val_A[i] = 0;


for ( i=0;  i < IA; i++ )
     ind_A[i] = 0;

for( i=0; i<M; i++ )
   b[i] = 0;

fgets ( c1, M1, arq1);
while (  !feof(arq1) & ( strcmp(c1, "COLUMNS") )  )
      fgets (c1, M1, arq1 );

if ( feof(arq1) )
   {
     printf ( "error 1 " );
     exit(1);
   }

fgets ( c1, M1 + 1, arq1);
strcpy ( car, c1);

while ( !feof(arq1)  &&  ( strstr(car, "R") == NULL )  )

   {
     fgets (c2, M2 + 1, arq1 );
     fgets (c3, M3 + 1, arq1 );
     j = atoi(&c1[5]) - 1 ;
     if ( j != k )
       cont++;


     if ( strstr ( c2, "obj" ) != NULL )
       {
         ind_c[cont_c] = j ;
         val_c[cont_c] = strtod( c3, (char **) NULL );
         cont_c++;
```

```
  }

else
  {

    val_A[cont_A] = ( strtod( c3, (char **) NULL ) );
    ind_A[cont_A] = atoi(&c2[1]) - 1 ;

    cont_A++ ;

    if ( j != k )
      {
        w = VA + j;
        ind_A[w] = cont_A - 1 ;
        k = j;
      }

  }

fgets( c4, M4 + 1, arq1 );

  if ( ( strstr( c4, "c" ) == NULL ))
     {
      strcpy ( c1, c4 );
      if ( strstr ( c1, "R" ) == NULL )
        {
          fgets ( car, 5, arq1 );
        }

     }
  else
     {
      fgets ( c5, M5 + 1, arq1 );

      val_A[cont_A] = ( strtod( c5, (char **) NULL ));
      ind_A[cont_A] = atoi(&c4[3]) - 1 ;
      cont_A++;
```

```
            if ( j != k )

              {
                w = VA + j;
                ind_A[w] = cont_A - 1 ;
                k = j;
              }

            fgets( c1, M1 + 1, arq1 );
           }

         strcpy ( car, c1 );
    }

if ( feof(arq1) )
  {
   printf( "error\n" );
   exit(1);
  }

fgets( c1, M1 + 1, arq1 );
strcpy( car, c1);

while( !feof(arq1) && ( strstr( car, "E" )  == NULL ) )
  {
    fgets( c2, M2 + 1, arq1);
    fgets( c3, M3 + 1, arq1 );

    i = atoi( &c2[1] ) - 1;
    b[i] =   strtod( c3, (char **) NULL );

    fgets( c4, M4 + 1, arq1 );

    if(  ( strstr( c4, "c" ) == NULL ) )
       {
         strcpy( c1, c4 );
         fgets( car, 5, arq1 );
```

```
        }
      else
        {
          fgets( c5, M5 + 1, arq1 );
          i = atoi( &c4[3] ) - 1 ;
          b[i] =    strtod( c5, (char **) NULL )  ;

          fgets( c1, M1 + 1, arq1 );
        }
     strcpy ( car, c1) ;

   }

}

/*********************************************************************/
/*                                                                   */
/*                      ordena um vetor em forma decrescente  e      */
/*          me da os indices das componentes segundo este ordem      */
/*                                                                   */
/*********************************************************************/



 void  sort( L, Y,  A, O  )
 int L, Y;
 int  O[], A[];

{

       int  i, j,  f, temp, tem, e ;

       i = L;
       j = Y;

       if ( (Y-L)%2 == 0 )
         {
          f = (Y+L)/2;
```

```
    e = A[f];
   }
else
   {
    f = (Y+L+1)/2;
    e = A[f];
   }
while ( i<=j )
   {
    while ( A[i] > e )
         {
           i = i + 1;
         }
    while ( e > A[j] )
         {
           j = j - 1;
         }
    if ( i <= j )
      {
        tem  = O[i];
        O[i] = O[j];
        O[j] = tem;
        temp = A[i];
        A[i] = A[j];
        A[j] = temp;
        i = i + 1;
        j = j - 1;
      }

   }

if ( i > j )

   {

    if  ( L < j )
      {
        sort( L, j, A, O );
```

```
            }

          if ( i < Y )
            {
              sort( i, Y, A, O) ;
            }
        }


}

/*************************************************************************/
/*              encontra a particao delta', usada para                   */
/*   calcular o limitante superior do  valor  ótimo do problema ini-     */
/*                              cial (PL)                                */
/*************************************************************************/
void particao2(void)
{

int i,j,k, kk, alfa, omega, cont_2[M], contador ;

  for( i=0; i<N; i++ )
    {
      posi_lin[i] = 0;
      D[i] = 0;
    }

  for( j=0; j<M; j++ )
    {
      cont_1[j] = 0;
      cont_2[j] = 0;
    }



  for ( i=0; i< VA; i++ )
    {
      k = ind_A[i];
```

```
    cont_1[k]++;
    if ( val_A[i] >= 0 )
      cont_2[k]++;
  }

for ( i =0; i < M; i++ )
  {
     if ( cont_1[i] != cont_2[i] )
       cont_1[i] = 0;
  }


for ( i=0; i< M; i++)
  O[i] = i;

sort(0, M-1, cont_1, O );

for( j=0; j<M; j++ )
  {
    contador = 0;
    if ( cont_1[j] != 0 )
      {
       for(k=0; k<N; k++ )
        {
          if ( k != N-1)
            alfa = ind_A[VA + k + 1];
          else
            alfa = VA;

          for( i= ind_A[VA + k]; i < alfa; i++ )
            {
              if( (ind_A[i] == O[j]) & ( D[k] == 0 )  )
                contador++;
            }
        }

      if(  contador == ( cont_1[j] )   )
        {
```

```
            for( kk=0; kk<N; kk++ )
             {
               if ( kk != N-1 )
                 alfa = ind_A[VA + kk + 1];
               else

                 alfa = VA;

               for( i= ind_A[VA + kk]; i< alfa; i++ )
                 {
                  omega = ind_A[i];
                  if ( omega == O[j] )
                    {
                      D[kk] = val_A[i];
                      posi_lin[kk] = O[j] + 1;
                    }

                 }
             }
         }

       }


    }



}

/*************************************************************************/
/*                      calcula o peso inicial  1 e 4                    */
/*************************************************************************/

void peso_inicial1(void)
{
  int   j;
```

```
  for( j=0; j < (T-1)*R; j++ )
    {
       g[j] = (float)1/R;
    }
  for ( j = (T-1)*R; j < N; j++ )
    {
      if ( S != 0)
           g[j] = (float)1/S ;
      else
           g[j] = (float)1/R;

    }

}
/*************************************************************************/
/*                    calcula o peso inicial 2                           */
/*************************************************************************/

void peso_inicial2(void)
{
  int  j;

  for( j=0; j < (T-1)*R; j++ )
    {
      if ( (j%R == 0) || ( j%R == 4 ))
        g[j] = (float)1/10;
      else
        {

           if ( (j%R == 1) || ( j%R == 3) )
               g[j] = (float)3/10;
           else
             g[j] = (float)2/10;
        }

    }
  for ( j = (T-1)*R; j < N; j++ )
```

```
    {
      if ( S != 0)
           g[j] = (float)1/S ;
      else
           g[j] = (float)1/R;

    }
  for (j=0; j< 20; j++ )
      printf( "g[%i] = %f\n", j, g[j] );


}

/*********************************************************************/
/*                  calcula o peso inicial  3                        */
/*********************************************************************/

void peso_inicial3(void)
{
  int  j;

  for( j=0; j < (T-1)*R; j++ )
    {
      if ( (j%R == 0) || ( j%R == 4 ))
        g[j] = (float)3/10;
      else
        {

           if ( (j%R == 1) || ( j%R == 3) )
               g[j] = (float)1/10;
           else
             g[j] = (float)2/10;
        }

    }
  for ( j = (T-1)*R; j < N; j++ )
    {
      if ( S != 0)
```

```
            g[j] = (float)1/S ;
        else
            g[j] = (float)1/R;

    }
  for (j=0; j< 20; j++ )
      printf( "g[%i] = %f\n", j, g[j] );

}

/*************************************************************************/
/*    Este programa encontra os limitantes superiores para as variáveis   */
/*************************************************************************/
 void agr_lim_var(void)
{

  int  i,j, k;
  float max;

  for(j=0; j<N; j++)
    posi_g[j] = 0;

  for(i=0; i<T; i++)
    cont_g[i] = 0;


  for ( k=0; k< T-1; k++ )
      {
        agr_var[k] = 0;
        max = 0;

       for( j=k*R; j< (k+1)*R; j++)
           {
             if ( g[j] > max )
               {
                 max = g[j];

               }
```

```
            if ( j == (k+1)*R - 1 )
             agr_var[k] = max;

          }

       for(j=k*R; j< (k+1)*R; j++)
         {
            if ( g[j] == max )
              {
                posi_g[j] = 1;
                cont_g[k]++;
              }
         }
      }

   max = 0;

   for ( j = (T-1)*R; j<N; j++ )
      {
       if ( g[j] > max )
            max = g[j];

       if (j == N-1 )
         agr_var[T-1] = max;
      }
   for( j= (T-1)*R; j<N; j++ )
     {
       if ( g[j] ==  max )
         {
           posi_g[j] = 1;
           cont_g[T-1]++;
         }
     }
}


/*****************************************************************************/
```

```
/*  faz as combinacoes lineares das colunas de A e tambem as combinacoes  */
/*  lineares das componentes de c. Obtemos assim a nova matriz Ag e o novo*/
/*                                  vetor cg.                              */
/*************************************************************************/

void combina(void)
{
 int i, j, k, s, w;
 float novo_val_A[VA];



 for ( j=0; j < T; j++ )
    cg[j] = 0;

 for ( i=0;  i < M + T; i++ )
   {
     for ( j = 0; j < T; j++ )
       Ag[i][j] = 0;
   }


 for ( k = 0; k < T; k++ )
   {
    for ( j = 0; j < VC; j++ )

      {  w = ind_c[j] ;
         if(   ( w >= k*R ) &&  ( w < ( k + 1 )*R )   )
         {
           cg[k] =  cg[k] +  (val_c[j])*(g[w]);
         }


      }
   }

 for ( j =0; j < N- 1; j++ )
   {
```

```
    for ( k = ind_A[VA + j];  k < ind_A[VA + j + 1]; k++ )
      {
        novo_val_A[k] = val_A[k]*(g[j]);
      }

  }

for ( k = ind_A[VA + N - 1];  k < VA ;  k++ )
  {

    novo_val_A[k] =  val_A[k]*(g[N-1]);

  }

for ( k = 0; k < T-1; k++ )
  {
    for ( j = ind_A[ VA + k*R] ;  j < ind_A[ VA + (k + 1)*R];  j++ )
      {

       s = ind_A[j];
       Ag[s][k] = Ag[s][k] + novo_val_A[j];

      }

    }

for ( j = ind_A[ VA + (T-1)*R];  j < VA;  j++ )
  {
    k = ind_A[j];
    Ag[k][T-1] = Ag[k][T-1]  +  novo_val_A[j];
  }

for(i=0; i<T; i++)
  Ag[M+i][i] = agr_var[i];

}
```

```
/*************************************************************************/
/*                    escreve o problema agregado em formato rev         */
/*************************************************************************/

 void esc_PA_rew(void)
 {

  int  i, j, k;


  fprintf( arq2, "%s\n", "NAME    " );
  fprintf( arq2, "%s\n", "OBJSENSE");
  fprintf( arq2, "%s\n", "  MAX" );
  fprintf( arq2, "%s\n", "ROWS" );
  fprintf( arq2, "%s\n", " N  obj");

  for( i=1; i <= M + T; i++ )
      fprintf ( arq2, "%s%i\n", " L  c", i );

  fprintf( arq2, "%s\n", "COLUMNS" );

  for( j=0; j<T; j++ )

    {
      k=0;

      if(  cg[j] != 0 )
         {
          fprintf( arq2, "%s", "    x" );
          fprintf( arq2, "%-8i", j+1);
          fprintf( arq2,  "%s", "obj  ");
          fprintf( arq2,  "%15f", cg[j] );
          k = k + 1;
         }


     for ( i= 0;  i < M + T; i++ )
        {
```

```
        if ( Ag[i][j]  != 0 )
           {
             if ( k%2 == 0 )
                {
                 fprintf( arq2, "%s", "    x" );
                 fprintf( arq2, "%-8i", j + 1);
                 fprintf( arq2, "%s", "c" );
                 fprintf( arq2, "%-4i",i+1 );
                 fprintf( arq2,  "%15f", Ag[i][j] );
                 k = k + 1;
                }
             else
                {
                 fprintf( arq2, "%s", "   c");
                 fprintf( arq2, "%-5i", i+1 );
                 fprintf( arq2, "%14f\n", Ag[i][j] );
                 k = k + 1;
                 }
           }
        if   ( ( i== M + T -1) && ( k%2 != 0 )  )
            fprintf( arq2, "%s\n", " " );
      }


   }

fprintf( arq2, "%s\n", "RHS" );
k = 0;

for ( i = 0; i < M + T; i++ )
   {
    if ( i < M )
      {

        if  (b[i] != 0 )
         {
           if(( k%2 == 0 )  )
             {
```

```
            fprintf( arq2, "%s",  "    rhs      c" );
            fprintf( arq2, "%-4i", i + 1 );
            fprintf( arq2, "%15f", b[i] );
            k = k+1;
           }
         else
           {
            fprintf( arq2, "%s", "   c");
            fprintf( arq2, "%-5i", i + 1);
            fprintf( arq2, "%14f\n", b[i] );
            k = k+1;
           }
       }
   }
 else
   {

         if(( k%2 == 0 )  )
           {
            fprintf( arq2, "%s",  "    rhs      c" );
            fprintf( arq2, "%-4i", i + 1 );
            fprintf( arq2, "%15i", 1 );
            k = k+1;
           }
         else
           {
            fprintf( arq2, "%s", "   c");
            fprintf( arq2, "%-5i", i + 1);
            fprintf( arq2, "%14i\n",  1 );
            k = k+1;
           }
   }


   if ( ( i == M+T-1 )  &&  ( k%2 != 0 ) )
      fprintf( arq2, "%s\n", " " );
}
```

```
 /* fprintf( arq2, "%s\n", "BOUNDS" );

 for ( j=0; j<T; j++ )
   {
     fprintf( arq2, "%s", " UP bnd      x");
     fprintf( arq2, "%-4i", j + 1);
     fprintf( arq2, "%14f\n", lim_var_ag[j]);
   }  */

     fprintf( arq2, "%s\n", "ENDATA" );



}

/*************************************************************************/
/*          resuelve el problema agregado usando o CPLEX                 */
/*                                                                       */
/*************************************************************************/

 void OptPa_leePa(arq2)
 {
 char *name = "  ";
 int   mac;
 int   mar;
 int   mae;
 int   objsen;
 double  *objx = NULL;
 double  *rhsx = NULL;
 char    *senx = NULL;
 double  *rngval = NULL;
 int     *nrowind = NULL;
 int     *matbeg  = NULL;
 int     *matcnt = NULL;
 int     *matind = NULL;
 double  *matval = NULL;
 double  *bdl = NULL;
 double  *bdu = NULL;
```

```
int     *etype = NULL;
int     *enzbeg = NULL;
int     *enzcnt = NULL;
int     *enzind = NULL;
double  *enzval = NULL;
char    *dataname = NULL;
char    *objname  = NULL;
char    *rhsname  = NULL;
char    *rngname  = NULL;
char    *bndname  = NULL;
char    **cname = NULL;
char    *cstore = NULL;
char    **rname = NULL;
char    *rstore = NULL;
char    *estore = NULL;
char    **ename = NULL;
int     macsz = 0;
int     marsz = 0;
int     matsz = 0;
int     maesz = 0;
int     enzsz = 0;
unsigned   cstorsz = 0;
unsigned   rstorsz = 0;
unsigned   estorsz = 0;
int   lpstat;
double   *slack = NULL;
double   *dj = NULL;
struct cpxlp        *lp = NULL;
struct cpxchannel   *userchan = NULL;
FILE   *solfile  = NULL;
char   message_string[LINE_LEN + 1 ];
int  status;
int  i ;
int   curmac,  curmar;



setscr_ind (1);
```

```
status = mpsread( name, &mac, &mar, &mae, &objsen, &objx, &rhsx, &senx,
                  &matbeg, &matcnt, &matind, &matval, &bdl, &bdu, &rngval,
                  &nrowind, &etype, &enzbeg, &enzcnt, &enzind, &enzval ,
                  &dataname, &objname, &rhsname, &rngname, &bndname, &cname,
                  &cstore, &rname, &rstore, &ename, &estore, &macsz, &marsz,

if ( status == 0 )
  {
   lp = (struct cpxlp *)

   loadprob( name, mac, mar, mae, objsen, objx, rhsx, senx, matbeg,
                  matcnt, matind, matval, bdl, bdu, rngval, nrowind, etype,
                   enzbeg,  enzcnt,  enzind,  enzval,  dataname, objname,
                  rhsname,  rngname, bndname, cname, cstore, rname, rstore ,
                  ename, estore, macsz, marsz, matsz, maesz, enzsz, cstorsz,
                  rstorsz,  estorsz ) ;


  status = optimize (lp);
   if ( status == 0 )
     {

      curmac = getmac (lp);
      curmar = getmar (lp);
      X = (double *) malloc ( (unsigned) curmac*sizeof(double) );
      pi = (double *) malloc ( (unsigned) curmar*sizeof(double) );
      slack = (double *) malloc ( (unsigned) curmar*sizeof(double) );
      dj  = (double *) malloc ( (unsigned) curmac*sizeof(double) );

       status = solution (lp, &lpstat, &obj, X, pi, slack, dj);
       if ( status == 0 )
         {

          printf( "\n Resultados do problema agregado :\n ");
          printf( "Solution status = %d\n", lpstat );
          printf( "Objective value  = %f\n", obj );
         }
```

```
      }

    }


        freeprob ( &lp);


 if ( slack )  free ( (char *) slack );
 if ( dj )     free ( (char *)  dj );
}

/*************************************************************************/
/*        produto    do vetor pi pela matriz A , estando a matriz        */
/*           A  na forma de listas ligadas ( ind_A  e  val_A )           */
/*************************************************************************/

{
 int i, j, k, s, alfa ;



 for( k=0; k<N; k++ )
    {
      produto[k] = 0;
      if ( k != N-1 )
        alfa = ind_A[VA + k + 1];
      else
        alfa = VA;


      for( j= ind_A[VA + k]; j< alfa; j++ )
        {
          s = ind_A[j];
          produto[k] = produto[k] + pi[s]*val_A[j];

        }
    }
```

```
 for( i=0; i<T-1; i++ )
   {
     for( k=i*R; k < (i+1)*R; k++ )
       {
         if( posi_g[k] == 1)
           produto[k] = produto[k] + pi[M+i]/cont_g[i] ;
       }
   }

 for( k = (T-1)*R; k<N; k++ )
   {
     if ( posi_g[k] == 1)
       {
         produto[k] = produto[k] + pi[M+T-1]/cont_g[T-1] ;
       }
   }

for( i=0; i<T-1; i++ )
   {
     for( k=i*R; k < (i+1)*R; k++ )
       {
         if( posi_g[k] == 1)
           {
             produto1[k] = produto[k] + pi[M+i];
             break;
           }

       }
   }

 for( k = (T-1)*R; k<N; k++ )
   {
     if ( posi_g[k] == 1)
       {
         produto1[k] = produto[k] + pi[M+T-1];
         break;
```

```
        }
    }
}

/*************************************************************************/
/*              calcula o limitante superior do valor otimo do pro-     */
/*                              blema inicial                           */
/*************************************************************************/

float limitante_superior()
{
  int i, j, l;
  float  beta, soma =0;
  float MAXIMO[M];


  prod_pi_A();



  for ( i=0; i<M; i++ )
    {
      MAXIMO[i] = 0;
      l=0;
      for ( j=0; j<N; j++ )
        {
         if( posi_lin[j] == i+1)
           {
             while ( ind_c[l] < j)
               l++;

             if (ind_c[l] == j)
               {
                 beta = MAX(0, (val_c[l] - produto[j])/D[j] );
                 MAXIMO[i] = MAX( MAXIMO[i], beta );
                 l++;

               }
```

```
            else
              {
                beta = MAX(0, ( - produto[j]/D[j]) );
                MAXIMO[i] = MAX( MAXIMO[i], beta );

              }
          }
        if ( j== N-1)
          {
           MAXIMO[i] = MAXIMO[i]*b[i];
          }
      }
    soma = soma + MAXIMO[i];

  }
printf("soma = %f\n", soma);

l=0;

for ( j=0; j<N; j++ )
  {

    if( posi_lin[j] == 0 )
      {
        while ( ind_c[l] < j)
          l++;
        if ( ind_c[l] == j )
          {
            soma = soma + MAX( 0, val_c[l] - produto[j] );
            l++;

          }
        else
          {
            soma = soma + MAX( 0,  - produto[j] );

          }
      }
```

```
    }

  lim_sup = obj + soma;
  printf( "lim_sup = %f\n", lim_sup );
  return(lim_sup);
}


/*********************************************************************/
/*          calcula o peso que vai ser usado para construir o        */
/*                          novo problema agregado                   */
/*********************************************************************/

void  novo_peso(num_iter)


{

 int  i, j, k, l, beta ,  num  ;
 float  dif, norma, soma, PARAM;
 float gamma[N] ;

 dif = LIM_SUP - LIM_INF;
 printf ("dif = %f\n", dif );

 printf("numero de iterac = %i\n", num_iter );
 PARAM = (float)(1 + Z)/( num_iter + Z);
 printf( "PARAMETRO = %f\n", PARAM );

 l = 0;

 norma = 0;

 for ( k=0; k<T; k++ )
   {
    if ( k != T-1 )
      beta = (k+1)*R;
    else
```

```
      beta = N;

    for ( j = k*R; j< beta; j++ )
       {
        if ( ind_c[l] == j)
          {         gamma[j] = ( val_c[l] - produto1[j])*X[k];
            l++;
          }
        else
            gamma[j] =  -produto1[j]*X[k];
        norma = norma + gamma[j]*gamma[j];

       }
  }

for ( k=0; k<T; k++ )
  {
   if ( k != T-1 )
     {
       beta = (k+1)*R;
       num = R;
     }
   else
     {
       beta = N;
       if ( S != 0 )
         num  = S;

       else
          num = R;
     }
   for ( i = k*R; i< beta; i++ )
      g[i] = g[i] + (PARAM*gamma[i]*dif)/(norma*num) ;

  }


for ( j=0;  j<N; j++ )
```

```
    g[j] = MAX( 0.000001,  MIN( 1, g[j] ) );


 for ( k=0;  k<T;  k++ )
    {
      if( k != T-1 )
       {
         beta = (k+1)*R;
       }
     else
       {
         beta = N;
       }
     soma = 0;
     for ( j= k*R;  j<beta;  j++ )
       {

         soma = soma + g[j];
         if ( j== beta - 1 )
            {

              for ( i=k*R;  i < beta;  i++ )
                   g[i] = g[i]/soma;
            }
       }
    }


}



void  main(argc, argv)
int argc;
char *argv[];
```

```
{

  char  nome1[12], nome2[12]  ;

  if ( argc !=  3)
     {
       printf("Especifique nome do arquivo *.rev e nome do arquivo de  saida

       exit(1);
     }

   strcpy( nome1, argv[1] );
       if (  (arq1 = fopen(nome1, "r" )  ) == NULL  )
         {
            printf ( "Nao posso abrir o arquivo inicial %s\n ", " nome1"  );
            exit(1);
         }
   lee ( arq1);

   if (arq1 != NULL )
     fclose (arq1);
     arq1 = NULL;

   particao2 ();

   peso_inicial1 ();

   num_iter = 1;

   do
     {
       agr_lim_var ();

       combina ();

       strcpy ( nome2, argv[2] );
```

```
        if (  ( arq2 = fopen(nome2, "wt" )  ) == NULL )
          {
            printf ( "Nao posso criar o arquivo %s\n ",  nome2  );
            exit(1);
          }

      esc_PA_rew();

      if (arq2 != NULL )
         fclose(arq2);


      OptPa_leePa(arq2);
      if ( num_iter == 1)          melhor_valor = obj;
      else
        melhor_valor = MAX( melhor_valor, obj);
      printf ("melhor_valor = %f\n", melhor_valor );

      LIM_INF = obj;
      printf( "LIM_INF = %f\n", LIM_INF );

      limitante_superior ();
      if ( num_iter == 1 )
          LIM_SUP = lim_sup;
      else
          LIM_SUP = MIN( lim_sup, LIM_SUP);
      printf("LIM_SUP = %f\n", LIM_SUP );

      if ( X)    free ( (char *) X);
      if ( pi)   free ( (char *) pi);
      novo_peso (num_iter);
      num_iter = num_iter + 1 ;
    }

    while (  num_iter < 81  );
}
```

Referências Bibliográficas

[1] Balas E.
Facets for the Knapsack Polytope.
Mathematical Programming 8 (1975), 146-164.

[2] Bazaraa M., Sheraly H. e Shetty C. M.
Nonlinear Programming Theory and Algorithms.
John Wiley - Sons, Inc. 1993.

[3] Bertsekas, D. P.
Nonlinear Programming
Athena Scientific. 1995.

[4] Dorfman R., Samuelson P. e Solow R
Linear Programming and Economic Analysis
McGraw-Hill, 1958.

[5] Evans J.R.
Aggregation in the Generalized Transportation Problem
Computers Operations Research 6 (1979), 199-204.

[6] Fisher, W.D.
Clustering and Aggregation in Economics.
Johns Hopkins University Press. 1969.

[7] García, L.
An Algorithm of feasible directions for the aggregation. problem
Optimization 20(2) (1989), 193-208.

[8] Geoffrion, A,M.
Elements of Large-Scale Mathematical Programming, Part I: Concepts.
Management Science 16 (1970), 652-675.

[9] Hallefjord A. e Storoy S.
Aggregation and disaggregation in integer programming problems.
Operations Research 38 (1990), 619-623.

[10] Humes Jr. C. e Humes, A.F.P.C.
Programação Linear-Um primeiro Curso.
In: IX Congresso Nacional de Matemática Aplicada e Computacional.
SBMAC, Brasília, 1986.

[11] Humes Jr. C.
Problemas de Grande Porte em Programação Matemática
In: Reunião de Matemática Aplicada,1, Río de Janeiro, 1978.

[12] Jörnstein, K., Leisten, R. e Storoy, S.
Gradient schemes in iterative aggregation procedures for variable-aggregated LP-problems.
Optimization 30 (1994), 251-268.

[13] Kallio, M.
Computing Bounds for the Optimal Value in Linear Programming.
Naval Res. Logist. Quart. 24 (1977), 301-308.

[14] Leisten, R.
A posteriori error bounds in linear programming aggregation.
Computers Operations Research 24 (1997), 1-16.

[15] Lima, E.L.
Curso de Análise. Volume 2. Segunda edição.
Gráfica Editora Hamburg Ltda. 1985.

[16] Litvinchev, I, S.
Decomposition-Aggregation method for convex programming problems.
Optimization 22 (1991), 47-56.

[17] Litvinchev, I.S, e Rangel S.
Localization of the optimal solution and a posteriori bounds for aggregation.
Computers Operations Research 26 (1999), 967-988.

[18] Mendelssohn, R.
Improved Bounds for Aggregated Linear Programs.
Operations Research 28 (1980), 1450-1453.

[19] Munkres J. R.
Analysis on Manifolds.
Addison-Wesley, 1991.

[20] Rogers, D.F., Plante, R.D., Wong, R.T., e Evans, J.R.
Aggregation and disaggregation techniques and methodology in optimization.
Operations Research 39 (1991), 553-582.

[21] Taylor, R. W.
Aggregate Programming in large scale linear systems .
Ph.D. Dissertation. 1983.

[22] Shetty, C.M. e Taylor, R.W.
Solving large scale linear programs by aggregation.
Computers Operations Research. 14 (1987), 385-393.

[23] Zipkin, P.
Bounds on the effect of aggregating variables in linear programs.
Operations Research 28(2) (1980), 403-418.

[24] Zipkin, P.
Bounds for row-aggregation in linear Programming.
Operations Research 28(4) (1980), 903-916.