UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

UMA OPÇÃO DE ANÁLISE DE PÓS-OTIMALIDADE PARA O ALGORITMO PROJECT

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA

ANTÔNIO SÉRGIO COELHO

FLORIANÓPOLIS

SANTA CATARINA - BRASIL

JULHO - 1983

UMA OPÇÃO DE ANÁLISE DE PÓS-OTIMALIDADE PARA O ALGORITMO PROJECT

ANTÔNIO SERGIO COELHO

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE:

"MESTRE EM ENGENHARIA"

ESPECIALIDADE ENGENHARIA DE PRODUÇÃO E APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO.

PROFº Antônio Diomário de Queiroz, Dr.
Coordenador do Programa
de Pós-Graduação em Engª de Produção

BANCA EXAMINADORA:

PROFº Wilhelm Rödder, Ph.D.
Presidente

PROFº Ariolando Tavares Araruna, Dr.

PROFº Paulo Renécio Nascimento, M.Sc.

PROFº Carlos A. Nunes Cosenza, Dr.

À esposa
Izete
e a meus pais
Antônio José
e
Maria

R E S U M O

O presente trabalho tem como objetivo desenvolver a teoria e os passos algorítmicos necessários para fazer uma análise de Pós-Otimalidade de um problema resolvido pelo Algoritmo PROJECT.

Primeiramente, será feita uma justificativa do trabalho, seguida de uma suscinta retrospectiva sobre alguns métodos de soluções para problemas lineares de otimização.

Logo após, será apresentado o algoritmo PROJECT, no qual foi baseado todo o desenvolvimento feito neste trabalho.

Finalmente, será desenvolvida a Análise de Pós-Otimalidade com a teoria e implementação algorítmica, para os principais parâmetros do Problema de Programação Linear.

ABSTRACT

The Objective of the present work is to develope the theory and algorithms that are necessary in post-optamality analiysis of problems resolved with the algorithm project.

First, the present work will be justified, followed by a sucinct retrospect concerning some methods for optimization of Linear Problems.

Afterwards, the algorithm project will be presented, in which was be founded the development all in this work.

Finally, the analysis of Post-optimality will be developed in theory and algorithm for the principle parameters of the LP-Problem.

S U M Á R I O

pág.

CAPÍTULO I

CAPÍTULO II

CAPÍTULO III

CAPÍTULO IV

C A P Í T U L O I

1. Introdução

1.1. Origem do Trabalho

O primeiro algoritmo para resolver o Problema de Programação Linear (PPL) foi o Simplex desenvolvido pelo matemático americano George Dantzig (veja/DA/cap. 2). Este algoritmo, após vários aprimoramentos, conseguiu se tornar bastante eficiente para resolver o tipo de problema a que se propôs, apesar de pesquisar a solução ótima andando de vértice em vértice.

O presente trabalho foi desenvolvido tendo como base o algoritmo PROJECT (veja /RB1/,/RB2/), que também tem o propósito de resolver o PPL genérico. Usando uma nova filosofia de procura da solução ótima.

O algoritmo PROJECT foi desenvolvido por Rödder e Blauth (veja /RB1/,/RB2/), usando a idéia de procurar o ótimo na direção do gradiente da função objetivo, projetado sobre o espaço nulo da matriz de restrições, sugerida anteriormente (veja /HN/, /LE/). Rödder reestudou esta teoria e desenvolveu um algoritmo para resolver o PPL, o qual mostrou uma boa eficiência numérica, o que incentivou a continuação das pesquisas para torná-lo ainda melhor.

Entre os vários ramos de pesquisa que ainda se encontravam abertos para o algoritmo PROJECT estava a Análise de Pós-Otimalidade, um dos principais requisitos para um algoritmo tornar-se competitivo.

Levando-se em consideração o fato mencionado acima, decidiu-se desenvolver uma opção de Análise de Pós-Otimalidade para o algoritmo PROJECT, surgindo desta forma o desenvolvimento teórico que será apresentado no decorrer deste trabalho.

1.2. Objetivo do Trabalho

O presente trabalho propõe-se a desenvolver a teoria e os passos algorítmicos necessários para permitir uma Análise de Pós-Otimalidade, no algoritmo PROJECT, levando em consideração os seguintes itens:

- Análise de sensibilidade sobre o vetor de recursos b;
- Análise de sensibilidade sobre o vetor de custos c;
- Adição de uma variável;
- Retirada de uma variável;
- Adição de uma restrição;
- Retirada de uma restrição.

Com estas opções o algoritmo PROJECT dispõe das principais ferramentas para resolver problemas práticos de Programação Linear.

1.3. Importância do Trabalho

Os problemas de Programação Linear não são totalmente estáveis, pode fazer-se necessário uma variação em alguns dos seus parâmetros para expressar melhor a situação real. Se o problema que se pretende resolver é de grande porte, torna-se inviável a solução de um novo problema para cada novo valor dos parâmetros.

Desta forma, fica claro que é necessário encontrar uma maneira eficiente de achar o novo ótimo sem resolver totalmente o novo problema, isto é, desenvolver uma Análise de Pós-Otimalidade.

As pesquisas sobre a Análise de Pós-Otimalidade, devido à sua importância, vêm sendo desenvolvidas por vários autores, isto torna-se evidente em um livro publicado por GAL (veja /GA/) sobre este assunto, no qual ele menciona mais de seiscentas referências ligadas ao mesmo.

Como o PROJECT se propõe a resolver o tipo de problema já mencionado, é óbvio que o desenvolvimento de uma opção de Análise de Pós-Otimalidade para este algoritmo é uma contribuição muito importante.

A importância do trabalho não se limita aos fatos mencionados, pois um estudo profundo da filosofia do algoritmo implicou no aparecimento de resultados matemáticos (veja Capítulo IV) e uma maior familiarização com as propriedades já existentes que, se forem devidamente aproveitados, podem resultar em uma melhoria na eficiência numérica do PROJECT.

1.4. Estrutura do Trabalho

O Capítulo II tem a função de mostrar alguns métodos não-Simplex para resolver o PPL ou suas estruturas especiais os quais são mais eficientes que o Simplex ou simplesmente contribuições matemáticas na área de Programação Linear.

No Capítulo III será apresentado o algoritmo PROJECT, com sua idéia básica, a teoria matemática e os passos algorítmicos para a resolução do PPL, tornando desta forma familiar a nomenclatura e os conceitos utilizados no próximo capítulo.

No Capítulo IV serão apresentados a teoria matemática, os conceitos básicos e os passos algorítmicos necessários para fazer-se uma Análise de Pós-Otimalidade dos itens já mencionados na seção 1.2., compondo a parte central do trabalho.

No Capítulo V tem-se uma conclusão a respeito do trabalho e as recomendações para futuras pesquisas nesta área.

CAPÍTULO II

2. Métodos de Resolução de Problemas de Programação Linear.

2.1. Visão Geral

A Técnica da Programação Linear aplica-se a problemas cuja natureza satisfazem hipóteses básicas bem definidas, tais como a linearidade e o emprego de parâmetros determinísticos conhecidos.

Para tais problemas, são construídos modelos matemáticos que os representam. Estes modelos, a semelhança de outros desenvolvidos por diferentes técnicas da Pesquisa Operacional, apresentam uma função objetivo e um conjunto de restrições que devem satisfazer às hipóteses básicas da técnica em questão:

Para resolver os modelos criados a partir dos problemas, por esta técnica, tem-se desenvolvido diferentes métodos de solução. Estes métodos são genéricos quando são possíveis de aplicar a qualquer destes modelos. Os algoritmos Simplex e de Khachian (ver referência ao final desta seção), por exemplo, se encaixam nesta categoria. Por outro lado, tem-se observado que estruturas especiais destes modelos permitem a utilização de métodos específicos que são muito eficientes apenas para estes casos, não se aplicando em geral. São, portanto, algoritmos de uso restrito, porém, de larga eficiência onde se aplicam. Os algoritmos dos modelos de transporte e os de fluxo em redes (ver referência ao final desta seção) são exemplos desta última situação.

Para se ter uma noção mais precisa da distinção que existe entre as duas classes de métodos, citam-se aqui, as principais idéias do método de Khachin, exemplo de método aplicado ao caso geral, e dos algoritmos de transporte e o de Ford-Fulkerson, exemplos de casos de aplicação específica.

O algoritmo de Khachian tornou-se conhecido em 1979. Sua filosofia básica é a de utilizar as restrições primais e duais do modelo, mais uma restrição construída a partir das funções objetivo primal e dual com o fim de encontrar um ponto do R^n, o que equivale, como se pode mostrar através de convenientes formulações matemáticas, a procurar um ponto viável em uma região ao redor do ponto. Isto torna-se possível pela utilização de computadores que trabalhem com determinadas precisões numéricas. Quando encontrado, ao final de um sistema iterativo que determina uma seqüência de elipsóides, o ponto viável fornece a solução ótima. Este algoritmo apresenta uma inovação notável: é o primeiro algoritmo com limite de complexidade numérica polinomial a resolver um Problema de Programação Linear. Além disso, trouxe uma idéia nova na forma de fazê-lo, totalmente distinta da metodologia do Simplex. Entretanto, tem-se observado pouca eficiência na sua aplicação numérica sem que, porém, se conheça restrições teóricas ao seu uso. Esta já não é a situação do algoritmo utilizado no problema dos transportes, de uso específico para modelos de Programação Linear com estrutura bem característica, e que, aproveitando-se deste tipo de estrutura, torna sua aplicação muito eficiente.

O problema em questão diz respeito à determinação de quantidades de recursos que devem ser transportados de um conjunto de origens para um conjunto de destinos, através de caminhos selecionados de forma a minimizar o custo total da operação. Este problema pode envolver aspectos mais complexos, como o da existência de destinos intermediários. Entretanto, por questão de simplicidade, se restringirá a presente análise ao caso geral, onde se considera a igualdade entre a capacidade de absorção nos destinos e o montante de recursos gerados nas origens.

Este problema permite a montagem de um modelo que atende às hipóteses básicas da Programação Linear, portanto, pode ser resolvido através do uso de um algoritmo genérico. Pode-se verificar, entretanto, que o uso do algoritmo específico para o problema é mais eficiente, embora este utilize a filosofia básica do Simplex. De fato, pode-se acelerar consideravelmente a resolução do modelo se forem adotadas técnicas especiais de determinação da solução básica viável inicial. Estas técnicas, conhecidas como a Regra do Noroeste e o Processo do Custo Mínimo, compõem o algoritmo específico em questão que, ainda permite a simplificação do teste de otimalidade da solução encontrada, utilizando-se variáveis duais cuja determinação é feita de forma rápida, facilitada pelo próprio algoritmo. Estes aspectos respondem pela maior eficiência do algorítmo específico frente ao genérico.

De forma similar, observa-se que o algoritmo de Ford-Fulkerson é mais eficiente que os algoritmos genéricos na resolução dos chamados problemas de fluxo em redes.

Estes problemas referem-se à análise do fluxo que pode ser enviado através de uma rede, ou seja, um conjunto de nós (pontos de ligação) e outro de arcos (elos de ligação entre dois nós) estruturados de forma que haja orientação no sentido dos arcos que, por sua vez, tais quais os nós, apresentam capacidades limitadas.

O objetivo do problema pode variar de acordo com os apec tos particulares que se deseja considerar na rede, tais como a determinação do fluxo viável através da rede ou valor máximo possível deste fluxo ou ainda a sucessão de caminhos de mínimo custo. O algoritmo de Ford-Fulkerson, aqui citado como exemplo, permite a obtenção do fluxo máximo viável que pode ser enviado através de uma rede.

Pode-se verificar que este problema satisfaz às hipóteses básicas da programação linear e, por isso, permite o emprego de algoritmos genéricos para sua resolução. Entretanto, utilizando-se do fato que o modelo que representa o problema tem estrutura especial, Ford-Fulkerson desenvolveram um algoritmo que determina o fluxo máximo. Para tanto, a partir de um fluxo arbitrário inicial, geralmente zero, e através de um processo iterativo utiliza-se um método indutivo de rotulação que permite incrementos discretos no fluxo, efetivados a cada iteração. Este processo é repetido até que se esgotem as possibilidades de rotularem-se os nós, a partir de um nó não-final, quando então se obtém o fluxo máximo.

Tem-se observado que este algoritmo é de uso fácil e simples. Entretanto, a razão principal de sua maior eficiência quando comparado com algoritmos genéricos, consiste no fato que o número de restrições, nestes últimos, aumentaria de forma muito rápida quando fosse incluído na rede novos arcos ou nós. Isto significa que, para redes complexas, é muito grande o número de restrições do modelo linear que representa o problema. Este incremento de restrições afeta com muito menor intensidade o algoritmo de Ford-Fulkerson.

Esta análise superficial pretende apenas mostrar a diversidade da natureza dos algoritmos, hoje usados na técnica da

Programação Linear. Informações mais precisas podem ser obtidas em fontes bibliográficas como as seguintes:

- Khachian: /KH/, /WO/, PA/, GO/ e /BR/.
- Transportes: /SI/ e /DA/.
- Ford.Fulkerson: /SI/ e /DA/.

2.2. O Algoritmo PROJECT no Universo da Programação Linear

Dentro da visão geral da Técnica da Programação Linear, exposta na seção anterior, pode-se caracterizar o algoritmo PROJECT como um método de solução geral, ou seja, aplicável a modelos de Programação Linear que apresentem estruturas quaisquer.

Isto generaliza sua aplicação, ampliando, teoricamente, seu uso a todos os modelos que satisfaçam as hipóteses básicas da Programação Linear. Entretanto, torna sua eficiência vulnerável a problemas lineares, cujos modelos apresentem estruturas particulares - isto é, algoritmos específicos, como os citados Ford-Fulkerson ou de transportes - que podem ser mais eficientes na resolução de tipos particulares de modelos.

Apresenta-se, nos próximos capítulos, uma visão geral do PROJECT.

C A P Í T U L O III

3. O Algoritmo PROJECT

3.1. A Filosofia

O algoritmo PROJECT é um algoritmo para resolver o problema clássico de Programação Linear. Sua idéia básica consiste em pesquisar o ótimo na direção do gradiente da função objetivo, projetado sobre o espaço nulo da màtriz de restrições.

Aumenta-se o valor da função objetivo até que uma ou mais variáveis livres[4] atinjam o zero. Neste ponto serão ativadas[5] todas as variáveis que atingiram o zero, ou seja, para cada uma destas variáveis x_j será adicionada uma restrição da forma $e_j^T x=0$ e o gradiente projetado é calculado novamente satisfazendo estas restrições adicionais. Este processo é repetido até que todas as componentes do gradiente projetado sejam nulas. Faz-se agora um teste de otimalidade, em caso de não ser ótimo, relaxam-se as restrições adicionais onde as variáveis duais correspondentes têm sinal "errado" (veja /RB1/, /RB2/), calcula-se novamente o gradiente projetado e repete-se o processo.

3.2. Matemática Básica

Dada uma matriz B de dimensões p x n e rank p, sabe-se que a projeção ortogonal de um vetor $c \in R^n$ sobre o espaço nulo de B é (veja /RB1/):

$$(I - B^T (BB^T)^{-1} B) c . \qquad (3.1)$$

---

[4] São as variáveis para as quais não existe uma restrição da forma $e_j^T x = 0$.

[5] São as variáveis para as quais adiciona-se uma restrição da forma $e_j^T x = 0$.

Para simplificar a notação far - se - á $P^B = I - B^T(BB^T)^{-1}B$, portanto $P^B c$ é a projeção acima citada. Denomina-se $(BB^T)^{-1}$ a projeção inversa correspondente. Nota-se que para resolver um sistema Bx = d de dimensões pxn e rank p, conhecendo uma solução viável x^0, a equação seguinte dá uma representação paramétrica para um conjunto de soluções viáveis x de Bx=d e $z=c^Tx$:

$$x = x^0 + P^B c\ (z - c^T x^0)/c^T P^B c, \text{ somente se } P^B c \neq 0. \quad (3.2)$$

(3.2) mostra que pode-se calcular uma solução x de um PPL para qualquer valor de z, ignorando as restrições de não-negatividade. Considerando a não-negatividade em (3.2), pode-se então determinar os $j \in J \subset \{1, 2, \ldots, n\}$, para os quais x_j chega primeiro a zero com o aumento de z.

Determinado J em (3.2), calcula-se agora a projeção de c sobre o espaço nulo da matriz B mais as restrições adicionais $e_j^T\ x=0$, $(j \in J)$. O espaço nulo de B mais as restrições adicionais $e_j^T\ x=0$ $(j \in J)$, pode ser escrito simplesmente como

$$\{x: x \in R^n, \begin{bmatrix} B \\ e_J^T \end{bmatrix} x = 0\}, \text{ onde } e_J = \{e_{j_1}, \ldots, e_{j_k}\}, \{j_1, \ldots, j_k\} = J$$

e e_{j_i} representa o j_i-ésimo vetor unitário do R^n.

Apenas para tornar mais evidente a aplicação do algoritmo, será usada a nomenclatura do PPL padrão.

Max. $c^T x$

s.a

$$Ax = b \qquad (3.3)$$

$$x \geq 0,$$

onde A é uma matriz mxn de rank m.

O lema e teorema a seguir fornecem as ferramentas necessárias para o cálculo da projeção de c sobre o espaço nulo da matriz $\begin{bmatrix} A \\ e_J^T \end{bmatrix}$, onde J é o conjunto dos índices das variáveis ativas.

<u>Lema 3.1</u>: Seja A uma matriz mxn, $D = AA^T$, $J=\{j_1,\ldots,j_k\} \subset \{1,\ldots,n\}$, $e_J=\{e_{j_1},\ldots,e_{j_k}\}$ e a_{j_i} a j_i-ésima coluna de A.

i) $\begin{bmatrix} A \\ e_J^T \end{bmatrix} \begin{bmatrix} A^T; & e_J \end{bmatrix} = \begin{bmatrix} D & a_J \\ a_J^T & I_k \end{bmatrix}$ onde I_k é uma matriz identidade kxk.

ii) Se todas as inversas existirem, tem-se:

$$\begin{bmatrix} D & a_J \\ a_J^T & I_k \end{bmatrix}^{-1} = \begin{bmatrix} (D-a_Ja_J^T)^{-1} & -(D-a_Ja_J^T)^{-1}a_J \\ -a_J(D-a_Ja_J^T)^{-1} & I_k+a_J^T(D-a_Ja_J^T)^{-1}a_J \end{bmatrix}.$$

iii) Para a, b ε R^m e todas as inversas existindo, tem-se:

$$(D-(+)ba^T)^{-1} = D^{-1}+(-)y\bar{y}^T.\ 1/(1-(+)a^Ty), \qquad (*)$$

com $y=D^{-1}b$ e $\bar{y}=D^{-1}a$.

O item i) mostra como pode ser calculada a projeção inversa após adicinarem-se as restrições $e_J^T\ x = 0$, $j\varepsilon J$; o ii) é um caso especial de inversão de uma matriz por partição (veja /HL/, p.36); e o iii) é uma forma simplificada de calcular a inversa $(D-(+)ba^T)^{-1}$ quando tem-se D^{-1}.

<u>Corolário 3.1</u>: Seja A uma matriz mxn, $D= AA^T$ e a ε R^m uma coluna de A então:

$$(D-(+)aa^T)^{-1} = D^{-1}+(-)yy^T.1/(1-(+)a^Ty), \text{ com } y=D^{-1}a.$$

O corolário 3.1. é um caso particular do Lema 3.1, item iii).

Teorema 3.1: Seja A, D, J, e_j e a_{j_i} definidos como no Lema 3.1, então

$$\begin{bmatrix} D & a_J \\ a_J^T & I_k \end{bmatrix}^{-1}$$ pode ser calculado da seguinte forma:

$$(D_{j_1})^{-1} = (D - a_{j_1} a_{j_1}^T)^{-1} = D^{-1} + y_1 y_1^T . 1/(1 - a_{j_1}^T y_1),$$

com $y_1 = D^{-1} a_{j_1}$,

$$(D_{j_1, j_2})^{-1} = (D_{j_1} - a_{j_2} a_{j_2}^T)^{-1} = (D_{j_1})^{-1} + y_2 y_2^T . 1/(1 - a_{j_2}^T y_2),$$

com $y_2 = (D_{j_1})^{-1} a_{j_2}$,

.

.

.

$$(D_J)^{-1} = (D_{J_1, \ldots, j_k} - a_{j_k} a_{j_k}^T)^{-1} = (D_{j_1, \ldots, j_{k-1}})^{-1} + y_k j_k^T .$$

$$1/(1 - a_{j_k}^T y_k),$$

com $y_k = (D_{j_1, \ldots, j_{k-1}})^{-1} a_{j_k}$. Então tem-se:

$$\begin{bmatrix} D & a_J \\ a_J^T & I_k \end{bmatrix}^{-1} = \begin{bmatrix} D_J^{-1} & -D_J^{-1} a_J \\ -a_J^T D_J^{-1} & I_k + a_J^T D_J^{-1} a_J \end{bmatrix}.$$

OBS: Para um conjunto $J' \subset J$ de k' elementos e dado D_J^{-1} a inversa $\begin{bmatrix} D & a_{J'} \\ a_{J'} & I_{k'} \end{bmatrix}^{-1}$ pode ser calculada em um processo similar, fazendo pequenas mudanças nas respectivas equações recursivas, ou seja, trocando $(D - a_j a_j^T)^{-1}$ por $(D + a_j a_j^T)^{-1}$ (veja corolário 3.1).

O lema a seguir dará as condições de otimalidade para uma solução de (3.3), as quais são uma aplicação das condições de kuhn - Tucker (veja /RB1/):

Lema 3.2: Seja $x = x_0^o$ uma solução viável de (3.3). Seja $J \subset \{1,\ldots,n\}$ tal que $x_j^o = 0$ para todo $j\varepsilon J$ e

$$c^T - c^T[A^T,e_J]\left[\begin{bmatrix} A \\ e_J^T \end{bmatrix}[A^T,e_J]\right]^{-1}\begin{bmatrix} A \\ e_J^T \end{bmatrix} = 0. \qquad (3.4)$$

Então é necessário e suficiente para a otimalidade de $x=x^o$ que

$$c^T[A^T,e_J]\begin{bmatrix} -D_J^{-1}a_J \\ I_k + a_J D_J^{-1} a_J \end{bmatrix} \leqq 0.$$

O teorema a seguir é a chave para a eficiência do algoritmo PROJECT (veja /RB2/).

Teorema 3.2: Todas as definições como no Lema 3.2. Seja (3.4) satisfeita, mas exista $J^* \neq \phi$, $J^* \subset J$ tal que:

$$u_j = c^T[A^T,e_J]\begin{bmatrix} -D_J^{-1}a_J \\ I_k + a_J^T D_J^{-1} a_J \end{bmatrix}_j > 0 \text{ para } j \ \varepsilon \ J^*,$$

onde $\begin{bmatrix} -D_J^{-1}a_J \\ I_k + a_J D_J^{-1} a_J \end{bmatrix}_j$ é a j-ésima coluna da matriz $\begin{bmatrix} -D_J^{-1}a_J \\ I_k + a_J^T D_J^{-1} a_J \end{bmatrix}$,

então para $J' = J - J^*$ tem-se:

$$\left[P\begin{bmatrix} A \\ e_{J'}^T \end{bmatrix}\right]^T e_j > 0 \text{ para } j\varepsilon J^*$$

Em outras palavras, se as variáveis duais u_j das restrições adicionais $e_j^T x=0$ $(j\varepsilon J^*)$ têm o sinal errado e se estas restrições são relaxadas, então a projeção de c sobre o espaço nulo da matriz $\begin{bmatrix} A \\ e_{J'}^T \end{bmatrix}$ aponta para o interior do poliedro convexo com os respectivos $x_j \geq 0$, $j \in J^*$. Isto também vale para relaxar todas as restrições $e_j^T x=0$ ($j \in J^*$) de uma só vez evitando desta forma que o algoritmo ande ao longo de uma aresta.

3.3. Considerações Algorítmicas

Para tornar mais simplificada a representação do gradiente projetado e das variáveis duais será usada a transformação conhecida de (3.3):

Max. z

S. a

$$\begin{bmatrix} 1 & -c^T \\ 0 & A \end{bmatrix} \begin{bmatrix} z \\ x' \end{bmatrix} = \begin{bmatrix} 0 \\ b' \end{bmatrix} \tag{3.5}$$

Reindexando $x \geq 0$ e fazendo $x:=$[6] $\begin{bmatrix} z \\ x' \end{bmatrix}$, $\bar{A} = \begin{bmatrix} 1 & -c^T \\ 0 & A \end{bmatrix}$, $b:=\begin{bmatrix} 0 \\ b' \end{bmatrix}$, $n:=n+1$

e $m:=m+1$ (3.5) fica equivalente a

$$\text{Max. } (1,0,\ldots,0)x,\ \bar{A}x = b,\ (x_2,\ldots,x_n) \geq 0 \tag{3.6}$$

Note que na forma (3.6) o gradiente da função objetivo é um vetor unitário, e isto é de grande vantagem para o cálculo da projeção do gradiente sobre o espaço nulo da matriz de restrições que fica o seguinte:

$$ca: = (1,0,\ldots,0) - (1,0,\ldots,0)\ \bar{A}^T(\bar{A}\bar{A}^T)^{-1}\bar{A},$$

chamando $\bar{D}^{-1} = (\bar{A}\bar{A}^T)^{-1}$ o bloco central de (3.6) e substituindo $\bar{A}^T$ por $\begin{bmatrix} 1 & 0 \\ -c & A^T \end{bmatrix}$ tem-se:

6 O símbolo := indica que a variável anterior assumirá o valor posterior a ele.

$$ca := (1,0,\ldots,0) - (1,0,\ldots,0)\begin{bmatrix}1 & 0\\ c & A\end{bmatrix}\bar{D}^{-1}\bar{A}$$

$$ca := (1,0,\ldots,0) - \bar{D}^{-1}(1,.)\bar{A}$$

3.4. Os Passos Algorítmicos do PROJECT

Nesta seção serão apresentados os passos do algoritmo PROJECT, mas deve-se considerar que em todos os passos o algoritmo PROJECT usa o PPL na forma (3.6), e para seguir os passos deve-se usar as fórmulas apresentadas nas seções anteriores.

1. Gere uma solução viável (se necessário use variáveis artificiais).
2. Calcule a projeção de c sobre o espaço nulo de $\bar{A}$.
3. Ande nesta direção até que uma ou mais das restrições de não negatividade $x_j \geq 0$ fiquem ativas (se isto não acontecer o problema é ilimitado).
4. Calcule a projeção de c sobre o espaço nulo de todas as restrições ativas. Se esta projeção é diferente do valor nulo, vá para 3. Senão,
5. faça um teste de otimalidade verificando todas as variáveis duais das restrições ativas $e_j^T x = 0$ ($x_j = 0$) onde e_j é o j-ésimo vetor unitário do R^n. Se todas estas variáveis têm o sinal certo, a solução é ótima. Senão,
6. relaxe todas as restrições $e_j^T x = 0$, para as quais as variáveis duais têm o sinal errado. Volte para 4.

Exemplo 1: Seja dado o problema a seguir:

$$\begin{aligned}
&\text{Max } z = x_1 + x_3\\
&\text{s.a.}\\
&\quad x_1 + x_2 = 1\\
&\quad x_1 + x_2 + x_3 \geq 2 \qquad \text{(E.1)}\\
&\quad x_1, x_2, x_3 \geq 0
\end{aligned}$$

Para transformar (E.1) em um PPL na forma padrão será introduzida a variável de folga x_4 na inequação, restando o seguinte:

$$\begin{aligned}
&\text{Max } z = x_1 + x_3\\
&\text{s.a}\\
&\quad x_1 + x_2 = 1\\
&\quad x_1 + x_2 + x_3 + x_4 = 2 \qquad \text{(E.2)}\\
&\quad x_1, x_2, x_3, x_4 \geq 0
\end{aligned}$$

Como o algoritmo PROJECT é aplicado para o PPL na forma apresentada na seção 3.3, o problema (E.2) ficará:

$$\begin{aligned}
&\text{Max } z \\
&\text{s. a.} \\
&\qquad z - x_1 - x_3 = 0 \\
&\qquad x_1 + x_2 = 1 \qquad\qquad (E.3) \\
&\qquad x_1 + x_2 + x_3 + x_4 = 2 \\
&\qquad x_1 , x_2, x_3, x_4 \geq 0
\end{aligned}$$

Passo 1:

Para obter-se uma solução inicial viável de (E.3) resolve-se o seguinte problema:

$$\begin{aligned}
&\text{Max } \bar{z} \\
&\text{s.a} \quad z + x_5 = 0 \\
&\qquad x_1 + x_2 + x_5 = 1 \qquad\qquad (E.4) \\
&\qquad x_1 + x_2 + x_3 + x_4 = 2 \\
&\qquad x_1, x_2, x_3, x_4, x_5 \geq 0
\end{aligned}$$

Passo 1.1: Solução inicial $\bar{z}=-1$, $x_4 = 2$, $x_5 = 1$

Passo 1.2:

$$\bar{A}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$\bar{D}=\bar{A}\bar{A}^T=\begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 2 & 4 \end{bmatrix}$$

$$\bar{D}^{-1} = \begin{bmatrix} 2/3 & -1/3 & 1/6 \\ -1/3 & 2/3 & -1/3 \\ 1/6 & -1/3 & 5/12 \end{bmatrix}$$

$\bar{D}^{-1}(1,.) = (2/3 \ -1/3 \ \ 1/6)$

$$\partial a := (1,0,0,0,0,0) - (2/3,-1/3,1/6)\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad \text{(Veja (3.6))}$$

$\partial a = (1/3, 1/6, 1/6, -1/6, -1/6, 1/3)$

Passo 1.3

$$x = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 1/3 \\ 1/6 \\ 1/6 \\ -1/6 \\ -1/6 \\ -1/3 \end{bmatrix} \cdot \Delta z / 1/3 \qquad \text{(veja (3.2))}$$

$\Delta z = 0$

$x := (-1, 0, 0, 0, 2, 1)$

A restrição $x_4 \geq 0$ torna-se ativa.

Passo 1.4

$$y = \bar{D}^{-1} a_4 = \begin{bmatrix} 1/6 \\ -1/3 \\ 5/12 \end{bmatrix}$$

$$\bar{D}^{-1} = \begin{bmatrix} 2/3 & -1/3 & 1/6 \\ -1/3 & 2/3 & -1/3 \\ 1/6 & -1/3 & 5/12 \end{bmatrix} + \begin{bmatrix} 1/6 \\ -1/3 \\ 5/12 \end{bmatrix} (1/6, -1/3, 5/12) . 1/(1 - 5/12)$$

(Veja corolário 3.1)

$$\bar{D}^{-1} = \begin{bmatrix} 5/7 & -3/7 & 2/7 \\ -3/7 & 6/7 & -4/7 \\ 2/7 & -4/7 & 5/7 \end{bmatrix}$$

$$\bar{D}^{-1} a_4 = \begin{bmatrix} 2/7 \\ -4/7 \\ 5/7 \end{bmatrix}$$ (Veja Teorema 3.1)

$$\bar{D}^{-1}(1,.) = (5/7, -3/7, 2/7, -2/7)$$

$$ca = (1, 0, 0, 0, 0, 0) - (5/7, -3/7, 2/7, -2/7) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

$$\hat{c}a = (2/7, 1/7, 1/7, 0, -2/7, -2/7)$$

Passo 1.3

$$x = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 2/7 \\ 1/7 \\ 1/7 \\ 0 \\ -2/7 \\ -2/7 \end{bmatrix} \Delta z / 2/7$$

$\Delta z = 1$

$x = (0, 1/2, 1/2, 0, 1, 0)$

A restrição $x_5 \geq 0$ torna-se ativa.

Passo 1.4

$$y = \bar{D}^{-1} a_6 = \begin{bmatrix} 2/7 \\ 3/7 \\ -2/7 \end{bmatrix}$$

$$\bar{D}^{-1} = \begin{bmatrix} 5/7 & -3/7 & 2/7 \\ -3/7 & 6/7 & -4/7 \\ 2/7 & -4/7 & 5/7 \end{bmatrix} + \begin{bmatrix} 2/7 \\ 3/7 \\ -2/7 \end{bmatrix} (2/7,\ 3/7,\ -2/7).\ 1/(1 - 5/7)$$

$$\bar{D}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3/2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\bar{D}^{-1} a_j = \begin{bmatrix} 0 & 1 \\ -1 & 3/2 \\ 1 & -1 \end{bmatrix}$$

$$ca = (1,0,0,0,0,0) - (1,0,0,0,-1) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ca = (0,0,0,0,0,0)$$

Passo 1.5

$$a_J^T \bar{D}^{-1} a_J = \begin{bmatrix} -1 & -1 \\ -1 & 5/2 \end{bmatrix} \qquad I_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$-(1,0,0,0,0,0) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & -3/2 \\ -1 & 1 \\ 2 & -1 \\ -1 & 7/2 \end{bmatrix} = (0,1) \quad \text{(Veja Lema 3.2)}$$

Logo x= (0 , 1/2 , 1/2 , 0 , 1,0)é uma solução ótima de (E.4), ou seja, x=(1/2 , 1/2 , 0 , 1,0)é uma solução viável de (E.3).

Passo 2

$$\bar{A}= \begin{bmatrix} 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\bar{A}\bar{A}^T = \begin{bmatrix} 3 & -1 & -2 \\ -1 & 2 & 2 \\ -2 & 2 & 4 \end{bmatrix}$$

$$\bar{D}^{-1} = \begin{bmatrix} 1/2 & 0 & 1/4 \\ 0 & 1 & -1/2 \\ 1/4 & -1/2 & 5/8 \end{bmatrix}$$

$$ca= (1,0,0,0,0) - (1/2 \quad 0 \quad 1/4) \begin{bmatrix} 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$ca= (1/2,\ 1/4,\ -1/4,\ 1/4,\ -1/4)$$

Passo 3

$$x= \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1/2 \\ 1/4 \\ -1/4 \\ 1/4 \\ -1/4 \end{bmatrix} \Delta z/\,1/2$$

$\Delta z = 1$

$x = (3/2,\ 1,\ 0,\ 1/2,\ 1/2)$

A restrição $x_3 \geq 0$ Torna-se ativa.

<u>Passo 4</u>

$$y = \bar{D}^{-1} a_3 = \begin{bmatrix} 1/4 \\ 1/2 \\ 1/8 \end{bmatrix}$$

$$\bar{D}^{-1} = \begin{bmatrix} 1/2 & 0 & 1/4 \\ 0 & 1 & -1/2 \\ 1/4 & -1/2 & 5/8 \end{bmatrix} + \begin{bmatrix} 1/4 \\ 1/2 \\ 1/8 \end{bmatrix} (1/4\,,\ 1/2\,,\ 1/8)\,.\ 1/_{5/8}$$

$$\bar{D}^{-1} = \begin{bmatrix} 2/3 & 1/3 & 1/3 \\ 1/3 & 5/3 & -1/3 \\ 1/3 & -1/3 & 2/3 \end{bmatrix}, \qquad \bar{D}^{-1} a_J = \begin{bmatrix} 2/3 \\ 4/3 \\ 1/3 \end{bmatrix}$$

$$ca = (1,0,0,0,0) - (2/3,\ 1/3,\ 1/3,\ -2/3) \begin{bmatrix} 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$ca = (1/3,\ 0,\ 0,\ 1/3,\ -1/3)$$

<u>Passo 3</u>

$$x = \begin{bmatrix} 3/2 \\ 1 \\ 0 \\ 1/2 \\ 1/3 \end{bmatrix} + \begin{bmatrix} 1/3 \\ 0 \\ 0 \\ 1/3 \\ -1/3 \end{bmatrix} \Delta z.3$$

$\Delta z = 1/2$

$x = (2,\ 1,\ 0\ ,\ 1,\ 0)$

A restrição $x_5 \geq 0$ torna-se ativa.

Passo 4

$$j = \bar{D}^{-1} a_5 = \begin{bmatrix} 1/3 \\ -1/3 \\ 2/3 \end{bmatrix}$$

$$\bar{D}^{-1} = \begin{bmatrix} 2/3 & 1/3 & 1/3 \\ 1/3 & 5/3 & -1/3 \\ 1/3 & -1/3 & 2/3 \end{bmatrix} + \begin{bmatrix} 1/3 \\ -1/3 \\ 2/3 \end{bmatrix} (1/3,\ -1/3,\ 2/3) \cdot 1/_{2/3}$$

$$\bar{D}^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix}$$

$$\bar{D}^{-1} a_J = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 2 \end{bmatrix}$$

$$ca = (1,0,0,0,0) - (1,0,1,-1,-1) \begin{bmatrix} 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ca = (0,0,0,0,0)$$

Passo 5

$$I_k + a_J \bar{D}^{-1} \cdot a_J = \begin{bmatrix} -1 & 1 \\ 1 & 3 \end{bmatrix}$$

$$(1,0,0,0,0)\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 2 \\ -1 & 1 \\ 1 & 3 \end{bmatrix} = (0,0)$$

Logo $x=(2,1,0,1,0)$ é uma solução ótima de (E.3).

CAPÍTULO IV

4. A Análise de Pós-Otimalidade

4.1. Base Matemática

Nesta seção será apresentada a teoria matemática necessária para o desenvolvimento das demais seções deste capítulo.

Definição: Seja a matriz H(mxn) de rank m, então: $H^T(HH^T)^{-1}$ é a inversa generalizada da matriz H, (veja /BO/).

Lema 4.1: Seja H como na definição acima e $N(H)=\{y: Hy=0\}$, então $x= H^T(HH^T)^{-1}b + y$ é uma solução de $Hx=b$ sse $y \varepsilon N(H)$.

Prova: Condições de suficiência: $x=H^T (HH^T)^{-1}b + y$ é uma solução de $Hx=b$, então temos:

$$Hx = HH^T(HH^T)^{-1} b + Hy$$

$$Hx = b + Hy$$

por hipótese tem-se que x é uma solução de $Hx=b$, então $Hy=0$.

Condições de necessidade: $y \varepsilon N(H)$, então

$$Hx = b + Hy$$

por hipótese tem-se que $y \varepsilon N(H)$, então $Hx=b$.

Uma solução como a do Lema 4.1. $x=H^T(HH^T)^{-1} b + y$ será chamada de solução pseudo-básica, do sistema $Hx=b$.

Retornando ao PPL (3.3) e particionando a matriz A da forma $[A_{\bar{J}}, A_J]$, onde $\bar{J}$ é o conjunto dos índices das variáveis livres e J o conjunto dos índices das variáveis ativas, $\bar{J} \cup J=\{1,...,n\}$ usando esta notação pode-se formular os lemas 4.2, 4.3, 4.4 que possibilitarão o uso de uma nomenclatura mais simplificada no decorrer deste trabalho.

<u>Lema 4.2</u>: Seja $A= [A_{\bar{J}}\ ,\ A_J]$, $D= AA^T$ e $D_J=D - A_JA_J^T$, então $D_J=A_{\bar{J}}A_{\bar{J}}^T$.

<u>Prova</u>:

$$D - A_JA_J^T = [A_{\bar{J}}\ ,\ A_J] \begin{bmatrix} A_{\bar{J}}^T \\ A_J^T \end{bmatrix} - A_JA_J^T$$

$$= A_{\bar{J}}A_{\bar{J}}^T + A_JA_J^T - A_JA_J^T = A_{\bar{J}}A_{\bar{J}}^T\ .$$

<u>Lema 4.3</u>: Seja $c^T-c^T\ [A^T,e_J] \left[\begin{bmatrix} A \\ e_J^T \end{bmatrix} [A^T,e_J]\right]^{-1} \begin{bmatrix} A \\ e_J^T \end{bmatrix}$, a projeção de c sobre o espaço nulo da matriz $\begin{bmatrix} A \\ e_J^T \end{bmatrix}$, onde $A=[A_{\bar{J}},A_J]$ e

$c^T= \left[c_{\bar{J}}^T\ ,\ c_J^T\right]$, então

$$c^T - c^T\ [A^T,e_J] \left[\begin{bmatrix} A \\ e_J^T \end{bmatrix} [A^T,e_J]\right]^{-1} \begin{bmatrix} A \\ e_J^T \end{bmatrix} = [c_{\bar{J}}^T - c_{\bar{J}}^T A_{\bar{J}}^T (A_{\bar{J}}A_{\bar{J}}^T)^{-1} A_{\bar{J}}, 0, \dots, 0].$$

<u>Prova</u>: Aplicando os Lemas 3.1 e 4.2 tem-se:

$$c^T-c^T\ [A^T,e_J] \left[\begin{bmatrix} A \\ e_J^T \end{bmatrix} [A^T,e_J\ ]\right]^{-1} \begin{bmatrix} A \\ e_J^T \end{bmatrix} = [c_{\bar{J}}^T, c_J^T] - [c_{\bar{J}}^T, c_J^T] \begin{bmatrix} A_{\bar{J}}^T & 0 \\ A_J^T & e_J \end{bmatrix}.$$

$$\begin{bmatrix} (A_{\bar{J}}A_{\bar{J}}^T)^{-1} & -(A_{\bar{J}}A_{\bar{J}}^T)^{-1}\ A_J \\ -A_J^T(A_{\bar{J}}A_{\bar{J}}^T)^{-1} & I+A_J^T\ (A-A_J^T)^{-1}A_J \end{bmatrix} \begin{bmatrix} A_{\bar{J}} & A_J \\ 0 & e_J^T \end{bmatrix} =$$

$$\left[c_{\bar{J}}^T\ ,\ c_J^T\right] - \left[c_{\bar{J}}^T\ ,\ c_J^T\right] \begin{bmatrix} A_{\bar{J}}^T D_J^{-1} & -A_{\bar{J}}^T D_J^{-1} A_J \\ 0 & e_J \end{bmatrix} \begin{bmatrix} A_{\bar{J}} & A_J \\ 0 & e_J^T \end{bmatrix}$$

$$\left[c_{\bar{J}}^T, c_J^T\right] - \left[c_{\bar{J}}^T, c_J^T\right] \begin{bmatrix} A_{\bar{J}}^T D_J^{-1} A_{\bar{J}} & 0 \\ 0 & I_k \end{bmatrix} =$$

$$\left[c_{\bar{J}}^T, c_J^T\right] - \left[c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_{\bar{J}}, c_J^T\right] = \left[c_{\bar{J}}^T - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_{\bar{J}}, 0 \ldots, 0\right]$$

O Lema 4.3 nos garante que para calcular a projeção de c sobre o espaço nulo da matriz $\begin{bmatrix} A \\ e_J^T \end{bmatrix}$, basta calcular $c_{\bar{J}}^T - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_{\bar{J}}$, pois as demais componentes serão sempre nulas.

<u>Lema 4.4</u>: Seja $A = [A_{\bar{J}}, A_J]$, $c^T = [c_{\bar{J}}^T, c_J^T]$ e $D_J = A_{\bar{J}} A_{\bar{J}}^T$, então

$$c^T [A^T , e_J] \begin{bmatrix} -D_J^{-1} A_J \\ I_k + A_J^T D_J^{-1} A_J \end{bmatrix} = c_J^T - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_J .$$

<u>Prova</u>: Aplicando o Lema 3.2 tem-se;

$$c^T [A^T, e_J] \begin{bmatrix} -D_J^{-1} A_J \\ I_k + A_J^T D_J^{-1} A_J \end{bmatrix} = [c_{\bar{J}}^T, c_J^T] \begin{bmatrix} A_{\bar{J}}^T & 0 \\ A_J^T & e_J \end{bmatrix} \begin{bmatrix} -D_J^{-1} A_J \\ I_k + A_J^T D_J^{-1} A_J \end{bmatrix}$$

$$= [c_{\bar{J}}^T, c_J^T] \begin{bmatrix} -A_{\bar{J}}^T D_J^{-1} A_J \\ e_J \end{bmatrix} = -c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_J + c_J^T$$

$$= c_J^T - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_J .$$

O Lema 4.4 mostra que para verificar a condição de otimalidade do Lema 3.2 é o mesmo que verificar se

$$c_J^T - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_J \leqq 0 .$$

Com o objetivo de facilitar a aplicação do algoritmo PROJECT usa-se PPL na forma (3.6), portanto, uma alteração no vetor custo c mudará a matriz de restrições e por conseguinte, o bloco central[7]. Os lemas e teorema a seguir darão as condições necessárias para obter-se o novo bloco central após uma alteração do vetor custo c.

Lema 4.5: Para algum $\bar{J} \subset \{1,\dots,n\}$ e problema (3.6) o bloco central é:

$$\bar{D}_J^{-1} = \left[\begin{bmatrix} 1 & -c_{\bar{J}}^T \\ 0 & A_{\bar{J}}^T \end{bmatrix}\begin{bmatrix} 1 & 0^T \\ -c_{\bar{J}} & A_{\bar{J}}^T \end{bmatrix}\right]^{-1} = \begin{bmatrix} 1+c_{\bar{J}}^T c_{\bar{J}} & -c_{\bar{J}}^T A_{\bar{J}}^T \\ -A_{\bar{J}} c_{\bar{J}} & D_J \end{bmatrix}^{-1} \qquad (4.1)$$

Lema 4.6: Seja $\begin{bmatrix} r & d^T \\ d & G \end{bmatrix}$ a partição correspondente de (4.1), então tem-se:

$$D_J^{-1} = G\left[I - (A_{\bar{J}} c_{\bar{J}}) d^T\right] . \; 1/(1 + (A_{\bar{J}} c_{\bar{J}})^T d).$$

Prova: Aplicando as fórmulas de cálculo da inversa de uma matriz por partição obtém-se:

$$G = D_J^{-1} - D_J^{-1}(-A_{\bar{J}} c_{\bar{J}}) . d^T$$

$$G = D_J^{-1}\left[I - (-A_{\bar{J}} c_{\bar{J}}) . d^T\right]$$

Multiplicando a equação acima pela direita com a inversa $\left[I - (-A_{\bar{J}} c_{\bar{J}}) d^T\right]^{-1}$ tem-se:

$$D_J^{-1} = G\left[I - (-A_{\bar{J}} c_{\bar{J}}) d^T\right]^{-1}$$

7 Bloco central é a inversa $(\bar{A}\bar{A}^T)^{-1}$.

aplicando o Lema 3.1, item iii) tem-se:

$$D_J^{-1} = G[I - (A_{\bar{J}}c_{\bar{J}})d^T].\ 1/(1 + (A_{\bar{J}}c_{\bar{J}})^T d).$$

Teorema 4.1: Dada a matriz inversa $D_J^{-1} = (A_{\bar{J}}A_{\bar{J}}^T)^{-1}$ e o novo vetor custo da função objetivo, $c \in R^n$ então a matriz

$$\begin{bmatrix} \tilde{r} & \tilde{d} \\ \tilde{d}^T & \tilde{G} \end{bmatrix} = \begin{bmatrix} 1+\tilde{c}_{\bar{J}}^T\tilde{c}_{\bar{J}} & -\tilde{c}_{\bar{J}}^T A_{\bar{J}}^T \\ -A_{\bar{J}}\tilde{c}_{\bar{J}} & D_J \end{bmatrix}^{-1}$$

pode ser calculada como segue:

$$\tilde{r} = (1 + \tilde{c}_{\bar{J}}^T\tilde{c}_{\bar{J}} + \tilde{c}_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_{\bar{J}}\tilde{c}_{\bar{J}})^{-1},$$

$$\tilde{d}^T = -\tilde{r}.\tilde{c}_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1},$$

$$\tilde{G} = D_J^{-1} + D_J^{-1} A_{\bar{J}}\tilde{c}_{\bar{J}}\tilde{d}^T.$$

O Teorema 4.1 é uma simples aplicação de partição de matrizes.

4.2. Análise de Sensibilidade no Vetor de Recursos b

4.2.1. Variável a Ser Ativada

Pode-se observar que durante o decorrer do algoritmo PROJECT, o vetor de recursos só é usado para encontrar uma solução viável inicial. Como a solução varia apenas na direção do gradiente da função objetivo projetado sobre o espaço nulo da matriz de restrições A, então as soluções permanecem viáveis a respeito de Ax=b. Também no final do algoritmo não tem-se a solução ótima explicitamente em função de b, como por exemplo no Simplex. O que torna necessário encontrar uma representação da solução ótima em função de b, para poder verificar a variação da solução ótima quando muda-se b para $b + e_i \Delta b_i$ para algum $i \in \{1,\ldots,m\}$, e_i o i-ésimo vetor unitário de R^m e variando $\Delta b_i \in R$.

<u>Teorema 4.2</u>: Seja $[x_{\bar{J}}^T = x_{\bar{J}}^{oT},\ x_J^T = 0^T]$ a solução ótima de (3.3) e seja $[A_{\bar{J}}, A_J]$ a partição correspondente de A, então obtém-se:

i) $x_{\bar{J}} = y_{\bar{J}} + A_{\bar{J}}^T D_{\bar{J}}^{-1}\, b$ para algum $y_{\bar{J}} \in N(A_{\bar{J}})$.

ii) $\bar{x}_{\bar{J}}\ (\Delta b_i) = x_{\bar{J}} + A_{\bar{J}}^T D_{\bar{J}}^{-1}\, e_i\ \Delta b_i$. (4.2)

iii) Chamando $v^i = A_{\bar{J}}^T D_{\bar{J}}^{-1} e_i$, o máximo Δb_i que mantém a não-negatividade de $\bar{x}_{\bar{J}}(\Delta b_i)$ pode ser calculado por:

$$\Delta b_i^0 = -x_{j_0}/v_{j_0}^i = \operatorname*{Min}_{j\varepsilon J} \{-x_j/v_j^i \ :\ v_j^i < 0\}. \qquad (4.3)$$

<u>Prova</u>:

i) Segue imediatamente do Lema 4.1

ii) $\bar{x}_{\bar{J}}\ (\Delta b_i) = y_{\bar{J}} + A_{\bar{J}}^T D_{\bar{J}}^{-1}(b+e_i \Delta b_i)$,

$\bar{x}_{\bar{J}}\ (\Delta b_i) = y_{\bar{J}} + A_{\bar{J}}^T D_{\bar{J}}^{-1} b + A_{\bar{J}}^T D_{\bar{J}}^{-1} e_i \Delta b_i$,

de i) tem-se:

$\bar{x}_{\bar{J}}\ (\Delta b_i) = x_{\bar{J}} + A_{\bar{J}}^T D_{\bar{J}}^{-1} e_i \Delta b_i$

iii) Óbvio por ii)

O teorema 4.2 fornece a(s) componente(s) de $x_{\bar{J}}$ que chega a zero para $\Delta b_i = \Delta b_i^0$ e será ativada. Na seção 4.2.2 serão identificadas as componentes de x_J que devem ser relaxadas para continuar aumentando Δb_i além de Δb_i^0 .

4.2.2. Variável a Ser Relaxada

O dual do problema (4.3) é:

$$\text{Min } \begin{bmatrix} u_m^T, u_n^T \end{bmatrix} \begin{bmatrix} b \\ 0 \end{bmatrix}, \quad \begin{bmatrix} u_m^T, u_n^T \end{bmatrix} \begin{bmatrix} A \\ I \end{bmatrix} = c^T, \; u_n \leqq 0 \qquad (4.4)$$

onde $u_m \in R^m$, $u_n \in R^n$ e $I(nxn)$.

Usando a mesma partição $\bar{J}, J$ como na solução ótima de (3.3) tem-se para a solução ótima do dual a seguinte partição:

$$\begin{bmatrix} \bar{u}_m^T, u_J^T, \bar{u}_{\bar{J}}^T \end{bmatrix} \begin{bmatrix} A_{\bar{J}} & A_J \\ 0 & I \\ I & 0 \end{bmatrix} = \begin{bmatrix} c_{\bar{J}}^T, c_J^T \end{bmatrix}, \qquad (4.5)$$

Em (4.5): $\bar{u}_m$ são as variáveis duais do sistema $Ax=b$,

$\bar{u}_J$ são as variáveis duais das restrições $e_J^T x=0$,

$\bar{u}_{\bar{J}}$ são as variáveis duais das restrições $e_{\bar{J}}^T x=0$, as quais estão no momento relaxadas (isto é $\bar{u}_{\bar{J}}=0$).

Pode-se separar (4.5) da seguinte forma:

$$\begin{bmatrix} \bar{u}_m^T, \bar{u}_J^T \end{bmatrix} \begin{bmatrix} A_{\bar{J}} & A_J \\ 0 & I \end{bmatrix} + \bar{u}_{\bar{J}}^T \begin{bmatrix} I, 0 \end{bmatrix} = \begin{bmatrix} c_{\bar{J}}^T, c_J^T \end{bmatrix}, \qquad (4.6)$$

e multiplicando (4.6) pela direita por $\begin{bmatrix} A_{\bar{J}}^T D_J^{-1} & -A_{\bar{J}}^T D_J^{-1} A_J \\ 0 & I \end{bmatrix}$,

ter-se-á:

$$\begin{bmatrix} \bar{u}_m^T, \bar{u}_J^T \end{bmatrix} = \begin{bmatrix} c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1}, \; c_J^T - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_J \end{bmatrix} + \bar{u}_{\bar{J}}^T \begin{bmatrix} -A_{\bar{J}}^T D_J^{-1} A_J \end{bmatrix}. \quad (4.7)$$

(4.7) expressa as variáveis $[\bar{u}_m, \bar{u}_J]$ em função das variáveis $\bar{u}_{\bar{J}}$.

Na seção 4.2.1. foi identificada a(s) variável(is) x_{j_0}, $j_0 \varepsilon \bar{J}$ a qual deve ser ativada para manter a viabilidade da solução, nesta seção a equação (4.7) fornece a(s) variável(is) $x_j{*}$, $j^* \varepsilon j$ que deve ser relaxada mantendo a otimalidade da solução, ou seja, a viabilidade do problema(4.4).

<u>Teorema 4.3:</u> Seja $j_0 \varepsilon \bar{J}$ um índice no qual (4.3) assume o mínimo, u_{j*}, $j^* \varepsilon J$ a(s) variável(is) que primeiro chega a zero quando aumentar u_{j_0} em (4.7).

Chamando $u_J^T := c_J^T - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_J$ e

$$w_0^j := A_{\bar{J}}^T D_J^{-1} a_{j_0} \quad ,$$

O mínimo u_{j_0} que mantém a não-positividade em (4.7) é dado por:

$$u_{j_0}^0 := -u_{j*}/w_{J*}^{j_0} = \min_{j \varepsilon J} \{u_j/w_j^{j_0} : w_j^{j_0} < 0\}. \qquad (4.8)$$

A prova do teorema 4.3 segue imediatamente de (4.7), lembrando que u_m não tem restrição de sinal e u_j é não positivo.

Depois de calculado o Δb_i^0 por (4.3) e $u_{j_0}^0$ por (4.8) a nova solução primal e dual calculada por (4.2) e (4.7) respectivamente, satisfazem as condições de Kuhn-Tucker.

Após relaxar $e_{j*}^T x=0$ e ativar $e_{j_0}^T x=0$ continua-se aumentando o Δb_i além de Δb_i^0. Como no caso clássico este processo termina quando:

- não existe nenhum $v_j^i < 0$ em (4.3).
- ou não existe nenhum $w_j^{j_0} < 0$ em (4.8).

A seguir será apresentado um algoritmo que faz esta análise de sensibilidade.

4.2.3. Implementação Algorítmica da Análise de Sensibilidade para o Vetor de Recursos b

Usando os símbolos das seções 4.2.1 e 4.2.2 formular-se-ão os passos da análise de sensibilidade do vetor b como segue:

1. Seja x_0 a solução ótima atual do problema (3.3)
2. Salve $bs:=b_i$, signum:=+.
3. $b_i:=bs$, $x:=x_0$, signum:=-signum.
4. $\Delta b_i := \underset{j \varepsilon \bar{J}}{\mathrm{Min}} \{x_j/v_j^i \ .\ \text{signum}: v_j^i \ .\ \text{signum} > 0\}$,

   memorize j_0 para o qual $-x_{j_0} / v_{j_0}^i = \Delta b_i$.

   $\Delta b_i := -\Delta b_i \ .\ \text{signum}$

   Se para todo $j \varepsilon \bar{J}$ $v_j^i \ .\ \text{signum} \leqq 0$, vá para 10.

5. $x_{\bar{J}} := x_{\bar{J}} + v^i \ .\ \Delta b_i$.
6. $\Delta u := -\underset{j \varepsilon J}{\mathrm{Min}} \{u_j/w_j^{j_0} : w_j^{j_0} < 0\}$,

   memorize j^* para o qual $-u_{j^*}/w_{j^*}^{j_0} = \Delta u$.

   Se para todo $j \varepsilon \bar{J}$ $w_j^{j_0} \geqq 0$, vá para 11

7. Relaxe a(s) restrição(ões) $e_{j^*}^T x=0$, $j^* \varepsilon J$,

   Ative a(s) restrição(ões) $e_{j_0}^T x=0$, $j_0 \varepsilon \bar{J}$,

   $J:= J - \{j^*\} \cup \{j_0\}$, $\bar{J}:=\bar{J} - \{j_0\} \cup \{j^*\}$,

   Atualize o valor da função objetivo.

8. Imprima - O valor da função objetivo
- A solução ótima atual
- $b_i + \Delta b_i$
- Variáveis duais

9. $b_i := b_i + \Delta b_i$, vá para 4.

10. Δb_i pode ser aumentado até infinito, o incremento relativo da função objetivo é igual a variável dual da i-ésima restrição. Vá para 12.

11. Depois de b_i não existe mais solução viável para o problema primal.

12. Se signum = -, vá para 3
Fim.

4.3. Análise de Sensibilidade no Vetor Custo c

4.3.1. Variação de c_j para uma Variável Ativa.

Estudar-se-á agora o caso em que c_{j_0} é mudado para $c_{j_0} + \Delta c_{j_0}$, onde deve-se distinguir duas situações: $j_0 \varepsilon J$ ou $\bar{J}$. Primeiro será estudada a influência na otimalidade do PPL (3.3) quando mudar c_{j_0} para $c_{j_0} + \Delta c_{j_0}$ onde $j_0 \varepsilon J$, ou seja, a variável x_{j_0} é uma variável ativa. Posteriormente, a situação em que $j_0 \varepsilon \bar{J}$.

Lema 4.7: Seja J, $\bar{J}$ a partição de $\{1,...,n\}$ correspondente à solução ótima do PPL (3.3) e $j_0 \varepsilon J$. Então tem-se:

A projeção do gradiente $c^T + e_{j_0}^T \Delta c_{j_0}$, onde e_{j_0} é o j_0-ésimo vetor unitário do R^n, sobre o espaço nulo de $\begin{bmatrix} A \\ e_J^T \end{bmatrix}$ permanece o vetor nulo para todo $\Delta c_{j_0} \varepsilon R$.

A prova do Lema 4.7 segue imediatamente do Lema 4.3 substituindo c_J^T por $c_J^T + e_{J_0}^T \Delta c_{j_0}$.

O Lema 4.7 garante que modificando c_{j_0} para $c_{j_0} + \Delta c_{j_0}$ a projeção do gradiente não muda. Esta mudança no vetor custo influencia apenas as variáveis duais, para ser mais preciso, a variável dual corresponde à restrição ativa $e_{j_0}^T x = 0$.

Lema 4.8: Dada a mesma situação como no Lema 4.7, o máximo que Δc_{j_0} pode assumir sem violar a não-negatividade das variáveis duais é dado por:

$$\Delta c_{j_0}^0 = -u_{j_0} .$$

Prova: O vetor das variáveis duais é dado por

$$\left[c_{\bar{J}}^T A_{\bar{J}} D_J^{-1},\ c_J^T + e_{j_0}^T \Delta c_{j_0} - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_J\right] \quad \text{(compare com (4.7))}.$$

Obviamente este vetor é influenciado com a variação de Δc_{j_0} somente em:

$$c_J^T + e_{j_0}^T \Delta c_{j_0} - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_J,$$

mais especificamente, a variação de Δc_{j_0} influencia apenas a componente:

$$c_{j_0} + \Delta c_{j_0} - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_{j_0} = u_{j_0} + \Delta c_{j_0} ,$$

para manter a não-positividade de u_{j_0} o valor máximo de Δc_j é $\Delta c_{j_0}^0 - u_{j_0}$.

Fazendo $c_{j_0} := c_{j_0} + \Delta c_{j_0}^0$ a correspondente restrição ativa será relaxada para permitir que se façam novas iterações e o novo ótimo seja encontrado.

4.3.2. Variação de c_j para uma Variável Livre

A análise de sensibilidade para c_{j_0}, $j_0 \in \bar{J}$ é mais complexa do que o caso anterior, pois o novo gradiente projetado pode, ou não, permanecer o vetor nulo. O caso onde o gradiente projetado não permanecer o vetor nulo quando mudar c_{j_0} para $c_{j_0} + \Delta c_{j_0}$, que é uma situação muito rara, só acontece quando o gradiente projetado fica nulo com menos que n-m variáveis ativas.

Lema 4.9: Dada a mesma situação do Lema 4.7 e seja $|\bar{J}| = m$ então a projeção do gradiente da função objetivo sobre o espaço nulo da matriz $\begin{bmatrix} A \\ e_J^T \end{bmatrix}$ dará o vetor nulo para qualquer variação de Δc_{j_0}, $j_0 \in \bar{J}$.

A prova do Lema 4.9 é óbvia, pois a matriz $\begin{bmatrix} A \\ e_J^T \end{bmatrix}$ onde $|\bar{J}|=m$ tem rank cheio e o único elemento do seu espaço nulo é o vetor nulo. Podemos verificar que $|\bar{J}|=m$ é o caso normal, no qual o gradiente projetado fica nulo apenas quando já foram ativadas n - m variáveis. O caso oposto onde $|\bar{J}| > m$ significa um caso muito raro, o vetor fica ortogonal ao espaço nulo de $\begin{bmatrix} A \\ e_J^T \end{bmatrix}$ e e_J tem menos que n-m vetores unitários, ou seja, foram ativadas menos que n-m variáveis. Menciona-se este caso para salientar que as condições a seguir servem como complemento matemático, mas não influenciarão na eficiência numérica da Análise de Pós-Otimalidade.

Um critério que permite verificar se a projeção do gradiente muda ou não com a variação de Δc_{j_0} é dado pelo lema a seguir:

Lema 4.10: Seja $c_{\bar{J}}^T - c_J^T A_J^T D_J^{-1} A_{\bar{J}} = 0$, $\Delta c_{j_0} \neq 0$, $j_0 \varepsilon \bar{J}$ e $e_{j_0}^T$ o j_0-ésimo vetor unitário do $R^{|\bar{J}|}$, então tem-se:

$$c_{\bar{J}}^T + e_{j_0}^T \Delta c_{j_0} - \left[c_{\bar{J}}^T + e_{j_0}^T \Delta c_{j_0}\right] A_{\bar{J}}^T D_J^{-1} A_{\bar{J}} = 0 \quad \text{sse}$$

$$e_{j_0}^T - a_{j_0}^T D_J^{-1} A_{\bar{J}} = 0.$$

Prova:

$$c_{\bar{J}}^T + e_{j_0}^T . \Delta c_{j_0} - \left[c_{\bar{J}}^T + e_{j_0}^T \Delta c_{j_0}\right] A_{\bar{J}}^T D_J^{-1} A_{\bar{J}} = 0$$

$$c_{\bar{J}}^T - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_{\bar{J}} + e_{j_0}^T \Delta c_{j_0} - e_{j_0}^T . \Delta c_{j_0} . A_{\bar{J}}^T D_J^{-1} A_{\bar{J}} = 0$$

por hipótese

$$c_{\bar{J}}^T - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_{\bar{J}} = 0 \text{ logo}$$

$$\left[e_{j_0}^T - e_{j_0}^T A_{\bar{J}}^T D_J^{-1} A_{\bar{J}}\right] \Delta c_{j_0} = 0, \text{ como } \Delta c_{j_0} \neq 0 \text{ tem-se:}$$

$$e_{j_0}^T - a_{j_0}^T D_J^{-1} A_{\bar{J}} = 0.$$

O Lema 4.10 permite um teste imediato dos coeficientes do gradiente projetado

$$c_{\bar{J}}^T + e_{j_0}^T \Delta c_{j_0} - \left[c_{\bar{J}}^T + e_{j_0}^T \Delta c_{j_0}\right] A_{\bar{J}}^T D_J^{-1} A_{\bar{J}} = \left[e_{j_0}^T - a_{j_0}^T D_J^{-1} A_{\bar{J}}\right] \Delta c_{j_0}$$

e as componentes que ficarem diferentes de zero para $\Delta c_{j_0} \neq 0$ deverão ser ativadas. Uma vez verificada a influência de Δc_{j_0} no gradiente projetado, estudar-se-á a influência de Δc_{j_0} nas variáveis duais. Observando a expressão (4.7) e lembrando que as va riáveis duais a respeito das restrições estruturais Ax=b são não restritas de sinal, então é suficiente manter a não-positividade de:

$$c_J^T - \left[c_{\bar{J}}^T + e_{j_0}^T \; \Delta c_{j_0}\right] A_{\bar{J}}^T D_J^{-1} A_J \qquad (4.9)$$

Lema 4.11: (4.9) mantém a não-positividade para todo

$$\Delta c_{j_0} : 0 < \Delta c_{j_0} < \Delta c_{j_0}^0 \quad \text{sse}$$

$$\Delta c_{j_0}^0 = u_j*/t_{j*} = \min_{j \varepsilon J} \{u_j/t_j : t_j < 0\} , \qquad \text{onde}$$

$$t_j := a_{j_0}^T D_j^{-1} a_j .$$

Prova:

$$c_J^T - \left[c_{\bar{J}}^T + e_{j_0}^T \; \Delta c_{j_0}\right] A_{\bar{J}}^T D_J^{-1} A_J \leqq 0$$

$$c_J^T - c_{\bar{J}}^T A_{\bar{J}}^T D_J^{-1} A_J - e_{j_0}^T A_{\bar{J}}^T D_J^{-1} A_J \Delta c_{j_0} \leqq 0$$

$$u_j - a_{j_0}^T D_J^{-1} A_J \Delta c_{j_0} \leqq 0$$

Como $u_J \leqq 0$ e chamando $t_j := a_{j_0}^T D_J^{-1} a_j$, $j \varepsilon J$ ob~~têm-se~~:

$$\Delta c_{j_0} \leqq u_{j*}/t_{j*} = \min_{j \varepsilon J} \{u_j/t_j : t_j < 0\} , \quad \text{portanto}$$

$$\Delta c_{j_0}^0 = u_{j*}/t_{j*} = \min_{j \varepsilon J} \{u_j/t_j : t_j < 0\}$$

Por razões computacionais tem-se que lembrar do fato que PROJECT deve ser programado para o PPL (3.5), ou seja, a função objetivo faz parte da matriz de restrições $\bar{A}$. Portanto, para calcular t_j usa - se o Lema 3.6 para obter a inversa D_J^{-1} da expressão (4.9) que corresponde ao bloco central do PPL (3.3). O Lema 4.11 identifica a variável dual que primeiro chega a zero quando aumenta-se o valor de Δc_{j_0}, portanto, a restrição $e_{j*}^T x=0$ deve ser relaxada para que esta variável se mantenha zero e para dar condições de uma nova pesquisa do ótimo.

Como a função objetivo faz parte da matriz de restrições e foi alterado c_{j_0} para $c_{j_0} + \Delta c_{j_0}^0$, após relaxar x_{j*} deve-se atualizar o bloco central. Esta atualização pode ser feita ativando e relaxando os seguintes vetores colunas:

$$\begin{bmatrix} -c_{j_0} \\ \\ a_{j_0} \end{bmatrix} \text{ e } \begin{bmatrix} -(c_{j_0} + \Delta c_{j_0}^0) \\ \\ a_{j_0} \end{bmatrix}$$ respectivamente, esta mesma atualização poderia ser feita através do Teorema 4.1.

4.3.3. Implementação Algorítmica da Análise de Sensibilidade no Vetor Custo c

Usando a terminologia das seções 4.3.1 e 4.3.2 formular-se-ão os passos para a análise de sensibilidade do vetor custo c.

1. Seja x_0 a solução ótima atual do PPL (3.5).

2. Salve cs:= c_{j_0}, signum =-.

3. c_{j_0} := cs, x:=x_0, signum:=-signum

4. Se $j_0 \in \bar{J}$, vá para 6.

   Se signum=-, vá para 11.

5. $\Delta c_{j_0} := -u_{j_0}$,

$c_{j_0} := c_{j_0} + \text{signum } \Delta c_{j_0}$,

$j^* := j_0$, vá para 9.

6. Se $|\bar{J}| = m$, vá para 8.

Calcule $ca := \left[e^T_{j_0} - a^T_{j_0} D^{-1} A_{\bar{J}}\right]$ usando Lema 4.10.

7. Se ca = 0, vá para 8.

Ative a(s) restrição(ões) $e^T_j x=0$, para a(s) qual(is) $ca_j \neq 0$.

Se $j_0 \varepsilon J$, vá para 5.

8. $\Delta c_{j_0} = u_{j^*}/t_{j^*} = \underset{j \varepsilon J}{\text{Min}} \{u_j/t_j \text{ . signum : signum. } t_j > 0\}$.

Se para todo $j \varepsilon J$: signum. $t_j \leq 0$, vá para 11.

9. Relaxe $e^T_{j^*} x=0$.

Se $j_0 \varepsilon J$, vá para 10.

$c_{j_0} := c_{j_0} + \text{signum. } \Delta c_{j_0}$,

atualize o bloco central,

atualize o valor da função objetivo.

10. Chame PROJECT

Imprima:-Valor da função objetivo

-Solução ótima

-Variáveis duais

$-c_{j_0}$

vá para 4.

11. Imprima 'Δc_{j_0} ilimitado'.

Se signum = +, vá para 3.

Fim

4.4. Adição de Uma Variável

4.4.1. Considerações Numéricas para Adição de uma Variável

Seja $\begin{bmatrix} x_{\bar{J}} = x_{\bar{J}}^0 \\ x_J = 0 \end{bmatrix}$ a solução ótima atual e D_J^{-1} o bloco central correspondente, adiciona - se a variável x_{n+1}, cuja coluna correspondente na matriz de restrições será a_{n+1}. Considera-se a variável x_{n+1} como sendo uma das variáveis ativas, então faz-se $J := J \cup \{n+1\}$. Esta consideração pode ser feita pelo fato de que ativar uma variável é equivalente a retirar esta coluna da matriz de restrições e relaxar uma variável é equivalente a colocar uma coluna na matriz de restrições. (veja os lemas 4.2, 4.3 e 4.4).

Como considera-se x_{n+1} ativa, para verificar se a solução atual continua ótima após adicionar x_{n+1}, tem-se que verificar se a variável dual correspondente à restrição $e_{n+1}^T x = 0$ tem o sinal certo. No caso afirmativo, a solução atual é a solução ótima do problema novo; caso contrário, relaxa-se a variável x_{n+1} para otimizar o problema novo.

4.4.2. Implementação Algorítmica para Adição de uma Variável

Usando a nomenclatura da seção 4.4.1. serão formulados os passos para adicionar uma variável.

1. Seja x_0 a solução ótima atual.
2. Faça $J := J \cup \{n+1\}$.
3. Calcule u_{n+1}.
4. Se $u_{n+1} \leq 0$, vá para 7.
5. Relaxe x_{n+1} usando o corolário **3.1**.

6. Resolva o problema novo.

7. Imprima - O valor da função objetivo.

 - Solução ótima

 - Variáveis duais

Fim.

4.5. Retirada de uma Variável

4.5.1. Considerações Numéricas para Retirar uma Variável

Seja $\begin{bmatrix} x_{\bar{J}} = x^0_{\bar{J}} \\ x_J = 0 \end{bmatrix}$ a solução ótima atual e D_J^{-1} o bloco central correspondente, pretende-se tirar a variável x_j, o que significa tirar a coluna a_j da matriz de restrições, têm-se duas situações possíveis:

i) Se x_j é uma variável ativa, ou seja, $j \varepsilon J$, então a coluna a_j já foi tirada da matriz de restrições. Para garantir que x_j nunca mais será relaxada far-se-á $J := J - \{j\}$, e a solução ótima ficará $\begin{bmatrix} x_{\bar{J}} = x^0_{\bar{J}} \\ x_J = 0 \end{bmatrix}$, onde as componentes $x_{\bar{J}}$ não mudarão de valor, apenas x_J terá uma componente a menos.

ii) Caso x_j seja uma variável livre, ou seja, $j \varepsilon \bar{J}$, então tem-se que minimizar x_j até atingir o zero e depois ativá-la, mas não incluindo j no conjunto J, garantindo desta forma que x_j não será mais relaxada. Para minimizar x_j, resolve-se o problema:

$$\text{Max } - x_j,\ Ax=B,\ x \geq 0, \qquad (4.10)$$

partindo da solução incial $\begin{bmatrix} x_{\bar{J}} = x^0_{\bar{J}} \\ x_J = 0 \end{bmatrix}$.

O problema (4.10) pode ser representado da seguinte forma:

$$\text{Max } z, \begin{bmatrix} 1 & e_j^T \\ 0 & A \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}, \; x \geq 0 \qquad (4.11)$$

onde e_j é o j-ésimo vetor unitário do R^n.

Considerando que se partirá da solução inicial $\begin{bmatrix} x_{\bar{J}} = x_{\bar{J}}^0 \\ x_J = 0 \end{bmatrix}$,

então o bloco central do problema (4.11) será obtido pela aplicação do lema a seguir:

Lema 4.12: Para algum $\bar{J} \subset \{1,\ldots,n\}$, o bloco central do problema (4.11) é:

$$D_J^{-1} := \begin{bmatrix} 2 & a_j^T \\ a_j & D_J \end{bmatrix}^{-1}. \qquad (4.12)$$

O Lema 4.12 é um caso particular do Lema 4.5.

O problema (4.11) é igual ao problema (3.5) trocando apenas o vetor custo c por e_j, então pode-se calcular D_J^{-1} a partir de $\bar{D}_J^{-1}$ usando o Lema 4.6 e o teorema a seguir:

Teorema 4.4: Dada a matriz D_J^{-1} e $e_j \in R^n$, o novo vetor custo, então a matriz $\begin{bmatrix} \tilde{r} & \tilde{d} \\ \tilde{d}^T & \tilde{G} \end{bmatrix}$ que é a partição correspondente de (4.12)

pode ser calculada da seguinte forma:

$$\tilde{r} = (2 + a_j^T D_J^{-1} a_j)^{-1}$$

$$\tilde{d}^T = -\tilde{r} a_j^T D_J^{-1}$$

$$\tilde{G} = D_J^{-1} - D_J^{-1} a_j \tilde{d}^T .$$

O Teorema 4.4 é um caso particular do Teorema 4.1.

Tendo já calculado $\tilde{D}_J^{-1}$ poder-se-á otimizar o problema (4.11). Se após esta otimização a variável $x_j > 0$, isto é, x_j estiver livre, isto indica que x_j não pode ser tirada do problema original, pois se tirar a coluna a_j a matriz de restrições ficará instável[8]. Caso contrário, se $x_j = 0$, ou seja, x_j estiver ativada então faz-se $J := J - \{j\}$, garantindo desta forma que x_j não será mais relaxada.

Agora deve-se retornar ao problema original (3.5) para otimizá-lo novamente, mas sem a variável x_j, para isto tem-se que calcular o novo bloco central $\bar{D}_J^{-1}$ usando o lema a seguir e o Teorema 4.1.

Lema 4.13: Seja $\begin{bmatrix} \tilde{r} & \tilde{d}^T \\ \tilde{d} & \tilde{G} \end{bmatrix}$ a partição correspondente de (4.12), então tem-se:

$$D_J^{-1} = \tilde{G}[I + a_j\tilde{d}^T] \;.\; 1/(1 - a_j^T\tilde{d}).$$

o Lema 4.13 é um caso particular do Lema 4.6.

4.5.2. Implementação Algorítmica para Retirada de uma Variável.

Usando as fórmulas e nomenclatura da seção 4.4.1., serão apresentados os passos necessários para retirada de uma variável.

1. Seja x_0 a solução ótima atual.
2. Se $j \in J$, faça $J := J - \{j\}$, vá para 12.
3. Calcule D_J^{-1} usando o Lema 4.6.
4. Calcule $\tilde{D}_J^{-1}$ usando o Teorema 4.4.

[8] A matriz não tem rank cheio, ou seja, o rank é menor que m.

5. Resolva o problema (4.11).

6. Se $j \notin J$, vá para 11.

7. Faça $J := J - \{j\}$.

8. Calcule D_J^{-1} usando o Lema 4.13.

9. Calcule $\bar{D}_J^{-1}$ usando o Teorema 4.1.

10. Resolva novamente o problema original (agora sem a variável x_j). vá para 12.

11. Imprima: 'x_j não pode ser retirada'.
vá para o Fim

12. Imprima: - O valor da função objetivo
- Solução ótima
- Variáveis duais

Fim.

4.6. Adição de uma Restrição

4.6.1. Considerações Numéricas para a Adição de uma Restrição

Sendo $\begin{bmatrix} x_{\bar{J}} = x_{\bar{J}}^0 \\ x_J = 0 \end{bmatrix}$ a solução ótima atual e $\bar{D}_J^{-1}$ o bloco central correspondente, pretende-se agora adicionar uma restrição estrutural. Existem duas situações possíveis:

i) A solução ótima atual satisfaz a restrição a ser adicionada, portanto a solução ótima do problema novo permanece a do atual. Caso a restrição a ser adicionada seja uma inequação, basta acrescentar à solução ótima atual uma variável de folga, para $(\leq)$ $s = b_{m+1} - a_{m+1}^T x$, e para $(\geq)$ $s = a_{m+1}^T x - b_{m+1}$.

ii) A solução ótima atual não satisfaz a restrição a ser adicionada, então esta solução não é uma solução viável para este pro blema novo, por isso deve-se adicionar uma variável artificial $x^a = [a^T_{m+1}x - b_{m+1}]$ e construir desta forma uma solução viável artificial. No caso da restrição ser uma inequação, considera-se a variável de folga igual a zero, ou seja, ativada.

Para conseguir uma solução viável do problema novo, minimiza-se a variável artifical x^a, esta minimização pode ser feita em uma primeira fase, onde será resolvido o seguinte problema:

Max z

s.a.

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & A & 0 \\ 0 & a^T_{m+1} & \pm 1 \end{bmatrix} \begin{bmatrix} z \\ x \\ x^a \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ b_{m+1} \end{bmatrix} \qquad (4.13)$$

$$\begin{bmatrix} x \\ x^a \end{bmatrix} \geqq 0.$$

O bloco central do problema (4.13) pode ser calculado usando o lema e teorema a seguir:

<u>Lema 4.14</u>: Seja $\begin{bmatrix} 1 & 0 & 1 \\ 0 & A & 0 \\ 0 & a^T_{m+1} & \pm 1 \end{bmatrix}$ a matriz de restrições do problema (4.13), então seu bloco central é dado por:

$$\tilde{D}_J^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & A & 0 \\ 0 & a^T_{m+1} & \pm 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & A^T & a_{m+1} \\ 1 & 0 & \pm 1 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & 0 & \pm 1 \\ 0 & AA^T & Aa_{m+1} \\ \pm 1 & a^T_{m+1}A^T & 1+a^T_{m+1}a_{m+1} \end{bmatrix}^{-1} \qquad (4.14)$$

Corolário 4.1: Seja B uma matriz pxp de rank p, então a inversa

$$\begin{bmatrix} 2 & 0 \\ 0 & B \end{bmatrix}^{-1} = \begin{bmatrix} 1/2 & 0 \\ 0 & B^{-1} \end{bmatrix}$$

Teorema 4.5: Seja $\bar{J}$, J a partição de{1,...,n} correspondente à solução ótima atual, $D_J^{-1} = (A_{\bar{J}} A_{\bar{J}}^T)^{-1}$, $a_{m+1,\bar{J}}^T$ as componentes $a_{m+1,j}^T$, $j \in \bar{J}$ de a_{m+1}^T e $\begin{bmatrix} \tilde{G} & \tilde{d} \\ \tilde{d}^T & \tilde{r} \end{bmatrix}$ a partição correspondente de (4.14) pode ser calculada da seguinte forma:

$$\tilde{r} = \left(1 + a_{m+1,\bar{J}}^T a_{m+1,\bar{J}} - \begin{bmatrix} \pm 1, a_{m+1,\bar{J}}^T A_{\bar{J}}^T \end{bmatrix} \begin{bmatrix} 1/2 & 0 \\ 0 & D_J^{-1} \end{bmatrix} \begin{bmatrix} \pm 1 \\ A_{\bar{J}} a_{m+1,\bar{J}} \end{bmatrix}\right)^{-1}$$

$$\tilde{d} = -\tilde{r} \begin{bmatrix} 1/2 & 0 \\ 0 & D_J^{-1} \end{bmatrix} \begin{bmatrix} \pm 1 \\ A_{\bar{J}} a_{m+1,\bar{J}} \end{bmatrix}$$

$$\tilde{G} = \begin{bmatrix} 1/2 & 0 \\ 0 & D_J^{-1} \end{bmatrix} \begin{bmatrix} 1/2 & 0 \\ 0 & D_J^{-1} \end{bmatrix} \begin{bmatrix} \pm 1 \\ A_{\bar{J}} a_{m+1,\bar{J}} \end{bmatrix} \tilde{d}^T.$$

O Teorema 4.5 é uma aplicação de cálculo da inversa de uma matriz por partição, (veja /HA/), a inversa D_J^{-1} que aparece no teorema pode ser calculada usando o Lema 4.6.

Calculado o bloco central do problema (4.13) através do Teorema 4.5, pode-se agora resolvê-lo para obter uma solução viável do problema novo. Se após resolver o problema (4.13) a variável artificial x^a for igual a zero, a solução obtida é uma solução viável do problema novo, no caso contrário este não tem solução viável.

No caso onde a variável artificial zera na primeira fase, será feita uma segunda fase para resolver o problema novo, ou seja, troca-se a função objetivo artificial pela função objetivo original no problema (4.13), e resolve-se novamente o problema. Esta troca de função objetivo pode ser feita usando o Lema 4.6 e Teorema 4.1.

4.6.2. Implementação Algorítmica para Adição de uma Restrição

Usando a nomenclatura e as fórmulas da seção 4.6.1. serão formulados os passos para a adição de uma restrição.

1. Seja x_0 a solução ótima atual.
2. Se x_0 não satisfaz a restrição a ser adicionada, vá para 7.
3. Se a nova restrição for uma equação, vá para 14.
4. Se a restrição for uma inequação com relação ($\leq$), faça:

   $s = b_{m+1} - a_{m+1}^T x_0$, vá para 6.

5. $s = a_{m+1}^T x_0 - b_{m+1}$.

6. $\bar{J} := \bar{J} \cup \{n+1\}$, vá para 14.

7. Se a nova restrição for uma inequação, faça $J := J \cup \{n+1\}$.

8. $x^a = \left[b_{m+1} - a_{m+1}^T x_0\right]$, $\bar{J} := \bar{J} \cup \{n+2\}$.

9. Calcule o bloco central da primeira fase usando o Lema 4.6 e o Teorema 4.5.
10. Resolva o problema (4.13).
11. Se $x^a \neq 0$, vá para 15.
12. Calcule o bloco central da segunda fase usando o Lema 4.6 e o Teorema 4.1.
13. Resolva o problema novo.
14. Imprima: - O valor da função objetivo

- Solução ótima
- Variáveis duais

Vá para o Fim.

15. Imprima: 'O problema novo não tem solução viável'.

Fim.

4.7. Retirada de uma Restrição

4.7.1. Considerações Numéricas para a Retirada de uma Restrição.

Seja $\begin{bmatrix} x_{\bar{J}} & = & x^0_{\bar{J}} \\ X_J & = & 0 \end{bmatrix}$ a solução ótima atual e $\bar{D}_J^{-1}$ o bloco central correspondente, pretende-se neste capítulo verificar a influência na otimalidade do PPL quando retirar a restrição $a_i^T x = b_i$. Pode-se observar facilmente que se adicionar uma variável de folga s_i sem restrição de sinal à restrição $a_i^T x = b_i$, tem-se a seguinte equação: $a_i^T x + s_i = b_i$, que por sua vez não é uma equação do problema original, pois para qualquer x existe um $s_i = b_i - a_i^T x$ que satisfaz a equação acima. Em outras palavras, com a adição da variável de folga s_i a equação $a_j^T x + s_i = b_i$ deixa de ser uma restrição para o problema.

Com a adição da variável de folga s_i a nova matriz de restrição será $[\bar{A}, e_i]$, onde e_i é o i-ésimo vetor unitário do R^m. Os Lemas 4.15 e 4.16 a seguir, mostrarão como pode ser calculado o bloco central do problema novo.

Lema 4.15: Seja $\bar{J}$, J a partição de {1,...,n} correspondente à solução ótima atual, $\bar{D}_J = \bar{A}_{\bar{J}} \bar{A}_{\bar{J}}^T$ e e_i o i-ésimo vetor unitário do

R^m, então:

$$\left[[\bar{A}_{\bar{J}},\ e_i]\begin{bmatrix}\bar{A}_{\bar{J}}^T\\ e_i^T\end{bmatrix}\right]^{-1} = \left[\bar{D}_J + e_i e_i^T\right]^{-1}.$$

Lema 4.16: Dadas as mesmas condições do Lema 4.15, tem-se:

$$\left[\bar{D}_J + e_i\ e_i^T\right]^{-1} = \bar{D}_J^{-1} - \bar{D}_{J\,(.,i)}^{-1},\bar{D}_{J(i,.)}^{-1} \cdot 1/(1 + \bar{D}_{J(i,i)}^{-1}),$$

onde $\bar{D}_{J(.,i)}^{-1}$ e $\bar{D}_{J(i,.)}^{-1}$ são i-ésima coluna e linha respectivamente $\bar{D}_J^{-1}$.

O Lema 4.16 é um caso particular do Lema 3.1 item iii) para $a=b=e_i$.

O lema a seguir mostrará a condição para que a solução atual permaneça ótima após a retirada da restrição $a_i^T x = b_i$.

Lema 4.17: Seja $u_i = c_{\bar{J}}^T\bar{A}_{\bar{J}}^T\bar{D}_J^{-1}(.,i) = 0$, a variável dual correspondente à restrição $a_i^T x = b_i$, então a solução ótima atual permanece ótima após tirarmos esta restrição.

Prova: O novo gradiente projetado é dado por:

$$ca := \left[c_{\bar{J}}^T\ ,\ 0\right] - \left[c_{\bar{J}}^T\ ,\ 0\right]\begin{bmatrix}\bar{A}_{\bar{J}}^T\\ e_i^T\end{bmatrix}\left[\bar{D}_J + e_i\ e_i^T\right]^{-1}\left[\bar{A}_{\bar{J}},\ e_i\right]$$

$$:= c_{\bar{J}}^T - \left[c_{\bar{J}}^T\ \bar{A}_{\bar{J}}^T\ \bar{D}_J^{-1}\bar{A}_{\bar{J}}^T\ ,\ c_{\bar{J}}^T\ \bar{A}_{\bar{J}}^T\ \bar{D}_{J(.,i)}^{-1}\right] + c_{\bar{J}}^T\ \bar{A}_{\bar{J}}^T\ \bar{D}_{J(i,.)}^{-1} \cdot 1/(1+\bar{D}_{J(i,i)}^{-1})\left[\bar{A}_{\bar{J}},e_i\right],$$

como a solução atual é ótima, tem-se $c_{\bar{J}}^T - c_{\bar{J}}^T\bar{A}_{\bar{J}}^T\bar{D}_J^{-1}\ \bar{A}_{\bar{J}} = 0$, e por hipótese $c_{\bar{J}}^T\bar{A}_{\bar{J}}\bar{D}_{J(.,i)}^{-1} = 0$, então:

$ca: = 0.$

O novo vetor $u_{\bar{J}}$ de variáveis duais é dado por:

$$u_J := c_J^T - \begin{bmatrix} c_{\bar{J}}^T , 0 \end{bmatrix} \begin{bmatrix} \bar{A}_{\bar{J}}^T \\ e_i^T \end{bmatrix} \begin{bmatrix} \bar{D}_J^{-1} + e_i \, e_i^T \end{bmatrix}^{-1} \bar{A}_J$$

$$:= c_J^T - c_{\bar{J}}^T \bar{A}_{\bar{J}}^T \bar{D}_J^{-1} \bar{A}_J - c_{\bar{J}}^T \bar{A}_{\bar{J}}^T \bar{D}_{J(.,i)}^{-1} \bar{D}_{J(i,.)}^{-1} . 1/(1+\bar{D}_{J(i,i)}^{-1}) . \bar{A}_J ,$$

sendo a solução atual ótima, sabe-se que $c_J^T - c_{\bar{J}}^T \bar{A}_{\bar{J}}^T \bar{D}_J^{-1} \bar{A}_J \leq 0$

e por hipótese $c_{\bar{J}}^T \bar{A}_{\bar{J}} \bar{D}_{J(.,1)}^{-1} = 0$, então $u_J = 0$.

O Lema 4.17 garante que se $u_i = 0$, então a solução ótima permanece a mesma após retirada a restrição $a_i^T x = b_i$.

No caso onde $u_i \neq 0$, fazem-se algumas iterações sem a restrição $a_i^T x = b_i$, para obter-se a solução ótima do problema novo.

4.7.2. Implementação Algorítmica para a Retirada de uma Restrição.

Usando os símbolos da seção 4.7.1, serão formulados os passos para a retirada de uma restrição.

1. Seja x_0 a solução ótima atual.
2. Faça $\bar{A} := \begin{bmatrix} \bar{A}_{\bar{J}} , e_i \end{bmatrix}$.
3. Calcule o bloco central do problema novo usando o Lema 4.16.
4. Faça $\bar{J} := \bar{J} \cup \{n+1\}$ e x_{n+1} sem restrição de sinal.
5. Se $u_i = 0$, vá para 7.
6. Resolva o problema novo.
7. Imprima: - O valor da função objetivo
   - Solução ótima
   - Variáveis duais

Fim.

CAPÍTULO V

5. Conclusões e Recomendações

5.1. Conclusões

As pesquisas sobre métodos de projeção para resolver o PPL já vêm sendo desenvolvidas por outros autores (veja /HN/,/LE/). Uma das principais dificuldades encontradas por estes autores foi o aumento muito rápido do número de linhas da matriz de restrições, por este motivo a aplicação destes métodos se tornava inviável na resolução do PPL.

No caso do algoritmo PROJECT isto foi resolvido com o uso de um Bloco Central, o qual mantém suas dimensões durante todo o desenvolvimento do algoritmo, este bloco mantém todas as informações necessárias para a resolução do PPL, e é relativamente fácil de ser calculado. Desta forma o algoritmo PROJECT tornou-se um algoritmo de projeção numericamente eficiente para resolver o PPL.

Considerando o fato de que o algoritmo PROJECT é numericamente eficiente, pode-se afirmar que a Análise de Pós-Otimalidade desenvolvida no presente trabalho é um dos instrumentos necessários para que esse algoritmo se torne competitivo.

A teoria desenvolvida neste trabalho foi totalmente adaptada aos conceitos e à filosofia do algoritmo PROJECT, tornando desta forma relativamente fácil a implementação algorítmica da Análise de Pós-Otimalidade, como foi apresentada no decorrer do trabalho.

Este trabalho além de fornecer uma opção de Análise de Pós Otimalidade como se propunha anteriormente, dá uma visão mais de talhada do comportamento deste algoritmo. Pois, analisando os parâmetros do PPL, tem-se uma idéia mais clara do comportamento de cada um deste parâmetros durante o desenvolver do algoritmo PROJECT.

5.2. Recomendações

Para o algoritmo PROJECT existem ainda vários ramos de pesquisa que podem ser desenvolvidos, aqui serão citados alguns deles:

- Desenvolver uma forma mais estável de calcular o Bloco Central, bem como armazená-lo de forma mais racional.
- Adaptar o PROJECT para a solução de problemas com estrutura de Blocos.
- Desenvolver um algoritmo dual adaptado aos princípios de procura do ótimo do algoritmo PROJECT.

Destas pesquisas a que tem uma ligação mais direta com a Análise de Pós-Otimalidade é o desenvolvimento de um algoritmo dual, o qual auxiliaria na análise do vetor de requerimento e na adição de uma restrição.

REFERÊNCIAS BIBLIOGRÁFICAS

/BE/ BEALE, E. M. L. The current algorithmic scope of matematical programming systems, in: Balinski M. L., Hellerman E. (eds.) Mathematical Programming Study 4 (1975), pp. 1-11.

/BM/ BISSCHOP, J., MEERAUS, A. Matrixaugmentation and partitioning in the uptating of the basis inverse, Mathematical Programming 13 (1977), pp. 241-254.

/BO/ BOULLIVON, T. L. ODELL, P. L. Generalized inverse matrices, New York, London, Sydney, Toronto. 1971.

/CK/ COOPER, L., KENNINGTON, J. Nonextreme point solution strategies for Linear Programming. Naval Research Logistics Quarterly 3 (1979), pp. 447-461.

/DA/ DANTZIG, G. B. Linear programming and extensions, Princeton University Press, Princeton, New Jersey. 1963.

/FU/ FULKERSON, D. R. An Out-of-Kilter method for minimal-cost flow problems, Journal of the Society of Industrial and Applied Mathematics 9,(1), (1971), pp. 18-27.

/GA/ GAL, T. Postoptimal analyses, parametric programming, and related topics, McGraw-Hill International Book Company, 1979.

/GO/ GOFFIN, J. L. Variable metric relaxation-or ellipsoid-methods, artigo apresentado no 'International Congress on Mathematical Programming', Rio de Janeiro, abril de 1981.

/GR/ GROETSCHEL, M. The ellipsoid method ant its consequences in combinatorial optimization, artigo apresentado no 'International Congress on Mathematical Programming', Rio de Janeiro, Abril 1981.

/HL/ HADLEY, G. Linear programming, 1. edition, Reading/...., 1962.

/HN/ HADLEY, G. Nonlinear and dynamic programming, 1. edition, Reading/..., 1964.

/HO/ HOULDEN, B. T. Some techniques of operations research, The English Universities Press Ltd, St. Paul's House Warwick Lane, London EC4, 1962.

/KH/ KHACHIAN, L. G. A polynomial algorithm in linear programming (em russo), Doklady Akademii Nauk SSSR, 1979, Vol. 244, nº 5 pp. 1093-1096.

/KN/ KÜHN, H. W. The hungarian method for the assigmment problem, Naval Research Logistics Quarterly 2(1975) pp. 83-97.

/KU/ KÜNZI, H. P. Die Duoplexmethode, Unternehmensforschung 7 (1963), pp. 103-116.

/KT/ KÜNZI, H. P., TAN, S. T. Lineare optimierung grosser systeme, Lecture Notes in Mathematics 4, Berlin/..., 1966.

/KZ/ KÜNZI, H. P., TZSCHACH, H. The duopex-algorithm, Numerische Mathematik 7 (1965), pp. 222-225.

/LE/ LEMKE, C. E. The constrained gradient method of linear programming, Journal of the Society for Industrial and Applied Mathematics, 9, 1961, pp. 1-17.

/PA/ PARANJAPE, S. R. The simplex method; Two variables replacement, Mg.Sc. 12 (1965), pp. 135-141.

/RB1/ RÖDDER, W., BLAUTH, M. PROJECT - An alternative LP-code, Boletim de Produção e Sistemas - UFSC, vol.2., Florianópolis (1980), pp. 24-35.

/RB2/ RÖDDER, W., BLAUTH, M. PROJECT - An alternative LP-algorithm. Artigo apresentado no 'International Congress on Mathematical Programming', Rio de Janeiro, abril de 1981.

/RO1/ RÖDDER, W., A note on linear dependency in PROJECT, Boletim de Produção e Sistemas - UFSC; vol.3, nº 2 Florianópolis (1981), pp. 51-62.

/RO2/ RÖDDER, W. PROJECT and its Invertion-Free Form, submetido a Operation Research Spektrum.

/SI/ SIMONNARD, M. Linear programming, traduzido por W. S. JEWEL. Englewood Cliffs, N.J., Prentice-Hall, Inc., 1966.

/WH/ WHITE, W.W. A status report on computing algorithms for mathematical programming, Computing Survey, vol. 5, nº3, (1973), pp. 135-166.

/WO/ WOLFE, P. Computation dificulty in mathematical programming: The efficienty of present techniques and their limits and the story of the ellipsoid (KHACHIAN) algorithm, artigo apresentado no 'International Congress on Mathematical Programming', Rio de Janeiro, abril de 1981.