

UTILIZAÇÃO DE ALGORITMOS DE PONTOS INTERIORES  
NA METODOLOGIA DE PLANOS CORTANTES

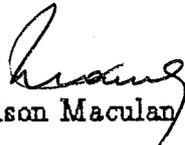
Angel Ramón Sánchez Delgado

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

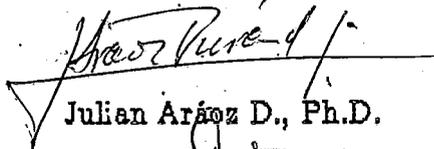
Aprovada por:



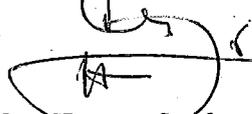
Clovis Caesar Gonzaga, D.Sc.  
(Presidente)



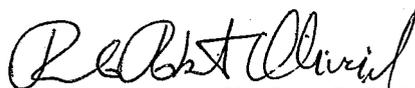
Nelson Maculan Filho, D.Sc.



Julian Araújo D., Ph.D.



Carlos Humes Junior, Ph.D.



Paulo Roberto Oliveira, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

Julho de 1991

DELGADO, ANGEL RAMÓN SÁNCHEZ

Utilização de Algoritmos de Pontos Interiores na Metodologia de Planos Cortantes (Rio de Janeiro), 1991.

134, 13 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 1991).

Tese – Universidade Federal do Rio de Janeiro, COPPE.

I. (Assunto da tese)

I. COPPE/UFRJ    II. Título (Série)

A minha esposa Marta e a minhas  
filhas: Priscila e Camila

E dos estudantes vagabundos, nessas  
noites mornas do Brasil, quando há  
muitas estrêlas no céu e muito  
desejo na terra.

## AGRADECIMENTOS

Quero expressar meu mais sincero agradecimento à COPPE – Sistemas e Computação da UFRJ pelo apoio brindado na realização do doutorado. Muito especialmente ao Professor Clovis C. Gonzaga pela seu grande labor como orientador e ante todo como guia na elaboração deste trabalho, pelo seu animo dentro e fora da sala. Creio que o que mais me aproxima a ele é sua sensibilidade e dedicação ao trabalho pela formação de jovens cientistas da América Latina.

Aproveito a oportunidade para agradecer com muito fervor ao Professor Nelson Maculan Filho nosso grande Reitor, docente e parceiro na minha realização de cursos e pesquisas nesta Universidade; ao Professor Julian Aráoz (USB–Venezuela) pela continuidade na minha formação como cientista; aos Professores Paulo Roberto Oliveira e Carlos Humes J. pela aceitação na participação desta banca de dissertação.

À Fundacion Gran Mariscal de Ayacucho (Venezuela), CAPES, FAPERJ, CNPq (Brasil) pelo apoio acadêmico e econômico digno para à realização desta tese. À meus companheiros de trabalho e seminários: Pedro, Daniel, Marcos, Carlos, etc.; muito especial a Daisy da Engenharia de Produção e à família Tinoco por suportar-me todos estes anos. Finalmente a minha grande família dos quais sempre tive o maior apoio na realização deste trabalho: Maria Antonia, Luis Delfin, Ismael, Nino, Carlos, Rosa, Marta, Marcelo, Cesar, Lenin, Asdrubal, etc.

Resumo da tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências (D.Sc.)

## UTILIZAÇÃO DE ALGORITMOS DE PONTOS INTERIORES NA METODOLOGIA DE PLANOS CORTANTES

Angel Ramón Sánchez Delgado

Julho de 1991

**Orientador:** Clovis Caesar Gonzaga

**Programa:** Engenharia de Sistemas e Computação

Nesta dissertação resolvemos o problema que qualquer algoritmo de pontos interiores enfrenta em uma metodologia de planos cortantes para resolver problemas de programação linear inteira. Se as relaxações do problema inteiro são resolvidas por um algoritmo de pontos interiores, então cada vez que um novo corte é encontrado a iteração presente se faz inviável e uma inicialização eficiente do algoritmo de pontos interiores resulta muito complicado. Neste trabalho descrevemos um algoritmo de pontos interiores que segue a trajetória central com baixa complexidade para reinicializar o problema. O método começa desde a solução inviável de baixo custo gerada na iteração prévia do método de planos cortantes e tem como saída uma solução viável com propriedades especiais (quase central) que faz este um bom ponto inicial para a seguinte iteração. O algoritmo é aplicado na resolução do problema de Matching Perfeito.

Abstract of thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Doctoral of Science (D.Sc.)

**THE USE OF INTERIOR POINT ALGORITHMS IN  
CUTTING PLANE METHODOLOGY**

**Angel Ramón Sánchez Delgado**

July 1991

**Thesis Supervisors:** Clovis Caesar Gonzaga

**Department:** Engenharia de Sistemas e Computação

In this dissertation we solve a problem that must be faced by an interior point algorithm when used in a cutting plane method for solving integer linear programming methods. If relaxation of the integer programming problema are solved by an interior point algorithm, then each time a new cut is found, the present iterate becomes infeasible, and an efficient initialization of the interior point algorithm is very difficult. We here describes a low complexity path following interior point algorithm for the intialization problem. The method start from a low-cost infeasible solution generated in the previous iteration of the cutting plane method and generates a feasible solution with special properties (near centrality) that make it a good initial point for the next iteration. The resulting algorithm is then applied to the solution of the Perfect Matching Problem.

## ÍNDICE

	Pág.
<b>CAPÍTULO I – INTRODUÇÃO E REVISÃO BIBLIOGRÁFICA</b>	<b>1</b>
I.1 – Programação Linear e a Revolucionária Metodologia	
Pontos Interiores	1
I.1.1 – Métodos Projetivos	4
I.1.2 – Métodos Afim-Escala	4
I.1.3 – Métodos de Trajetória Central	5
I.1.4 – Método de Redução Potencial	7
I.2 – Pontos Interiores e Problemas Combinatórios	9
I.2.1 – Introdução	9
I.2.2 – Alguns Elementos da Análise Convexa	10
I.2.3 – Método de Planos Cortantes	12
I.3 – Organização da Tese	18
 <b>CAPÍTULO II – UMA VARIANTE DO ALGORITMO DE HERTOG- ROOS-TERLAKY PARA RESOLVER O PROBLEMA INICIAL</b>	 <b>22</b>
II.1 – Introdução	22
II.2 – O Problema Inicial	23
II.2.1 – Observação	24
II.2.2 – Definição	25
II.2.3 – Proposição	26
II.3 – Obtenção de uma Direção Newton	27

	Pág.
II.3.1 – Cálculos	27
II.3.2 – Direção	28
I.3.3 – Proximidade	28
I.3.3.1 – Definição	28
II.3.3.2 – Definição	29
II.4 – O Algoritmo Início	29
II.4.1 – Características Gerais	29
II.4.2 – Dados	30
II.4.3 – Procedimento Geral	31
II.5 – Complexidade do Algoritmo	31
II.5.1 – Propriedades Primal–Dual	31
II.5.1.1 – Proposição	31
II.5.1.2 – Observação	32
II.5.1.3 – Proposição	33
II.5.1.4 – Proposição	35
II.5.2 – O Lema Auxiliar	36
II.5.2.1 – Lema Auxiliar	36
II.5.2.2 – Observação	37
II.5.2.3 – Proposição	37
II.5.2.4 – Proposição	38
II.5.3 – Teoremas Centrais	39
II.5.3.1 – Deduções	39
II.5.3.2 – Teorema	40
II.5.3.3 – Teorema	41
II.5.3.4 – Corolário	44
II.5.3.5 – Observação	45

	<b>Pág.</b>
II.6 – Obtenção do Ponto Inicial	45
<b>CAPÍTULO III – REOTIMIZAÇÃO COM METODOLOGIA</b>	
<b>PONTOS INTERIORES</b>	<b>47</b>
III.1 – Introdução	47
III.2 – Agregação de uma Restrição ao Problema Original	48
III.2.1 – Trajetória Central Associada ao Problema	48
III.2.1.1 – Introdução	48
III.2.1.2 – Objetivo Geral	50
III.2.1.3 – Cálculos	51
III.2.1.4 – Direção Newton e Proximidade	52
III.2.1.4.1 – Definição	52
III.2.1.4.2 – Definição	52
III.2.1.4.3 – Observação	53
III.2.1.4.4 – Lema	53
III.2.1.4.5 – Lema	55
III.2.1.4.6 – Teorema	56
III.2.1.4.7 – Observação	57
III.2.2 – Trajetória Central Associada ao Problema	
Perturbado com 1-Cópia	57
III.2.2.1 – Observação	57
III.2.2.2 – Observação	58
III.2.2.3 – Definição	59

	Pág.
III.2.2.4 – Proposição	59
III.2.3 – Propriedade Dual de Pontos Próximos a Trajetória Central do problema	61
III.2.3.1 – Notação	61
III.2.3.2 – Proposição	62
III.2.3.3 – Observação	63
III.2.3.4 – Lema	63
III.2.3.5 – Teorema	64
III.2.3.6 – Corolário	67
III.2.3.7 – Corolário	68
III.2.4 – Alcance à região Perturbada e Condição de Parada	69
III.2.4.1 – Observação	69
III.2.4.2 – Observação	72
III.2.4.3 – Cálculos	74
III.2.4.4 – Lema	75
III.2.4.5 – Lema	75
III.2.4.6 – Lema	77
III.3 – O Algoritmo Reiniciação	79
III.3.1 – Características Gerais	79
III.3.2 – Dados	80
III.3.3 – Procedimento Geral	81
III.3.3.1 – Observação (Saída do Algoritmo)	83
III.4 – Complexidade do Algoritmo	84
III.4.1 – Lema	84
III.4.2 – Proposição	88

	Pág.
III.4.3 – Proposição	88
III.4.4 – Observação	89
III.5 – Agregação de um Conjunto de Restrições	90
III.6 – Agregação de uma Variável no Problema Original	92
III.6.1 – Observação	92
III.6.2 – Observação	93
III.6.3 – Observação	94
CAPÍTULO IV – ALGORITMOS DE PLANOS CORTANTES PARA RESOLVER O PROBLEMA DE MATCHING PERFEITO	97
IV.1 – Introdução	97
IV.2 – O Problema de Matching Perfeito	100
IV.2.1 – Formulação do Problema	100
IV.2.2 – Formulação do Problema como Programa Linear	100
IV.2.3 – Caracterização da Envoltória Convexa das Soluções do problema	102
IV.2.4 – O Problema de Separação	108
IV.3 – Algoritmo para o Problema de Matching Perfeito	109
IV.3.1 – Características Gerais	109
IV.3.2 – Solução Viável Inicial do PMP	111
IV.3.3 – Relaxação Inicial do PMP	113
IV.3.4 – Arredondamento de Soluções	114
IV.3.5 – Rotinas de Separação	116
IV.3.6 – Descrição Esquemática do Algoritmo	118

	<b>Pág.</b>
<b>CAPÍTULO V – CONCLUSÕES</b>	123
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	126

## CAPÍTULO I

## INTRODUÇÃO E REVISÃO BIBLIOGRÁFICA

I.1- PROGRAMAÇÃO LINEAR E A REVOLUCIONÁRIA  
METODOLOGIA – PONTOS – INTERIORES

Nesta dissertação consideramos o problema de programação linear no formato dual:

$$\text{Minimizar } c^T x$$

$$(P)$$

$$\text{Sujeito a: } Ax \leq b$$

onde:  $c \in \mathbb{R}^n$ ;  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$

Mesmo que o problema (P) pode ser visto como um problema de programação não linear (PNL); durante as últimas décadas sua resolução foi por um caminho muito diferente ao conhecido em PNL. Muito dos trabalhos na bibliografia consideram o problema de programação linear no formato primal, isto é, o conjunto de soluções viáveis está definido por um sistema de equações e restrição de não negatividade sobre as variáveis. A relação entre ambos formatos pode ser vista em GONZAGA [19].

É bem sabido que as soluções ótimas de (P) estão nos vértices do poliedro descrito pelo sistema de desigualdades:  $Ax \leq b$ .

Tradicionalmente o problema (P) se resolve usando o conhecido método

Simplex de DANTZIG [8]. Neste, a busca de uma solução Ótima se concretiza em uma visita aos vértices do poliedro de forma ordenada. No pior caso, o Simplex é exponencial na dimensão do problema, mas na prática este tende a requerer um número linear de iterações na dimensão do problema; mais ainda, existem muitas implementações eficientes do método. Outra forma de resolver (P) aproximadamente é com métodos que atravessam o interior do poliedro, (suponha que este tem interior não vazio), isto é conhecido como pontos interiores.

Desde 1947 pesquisadores como Von Neumann, Hoffman, etc., apresentarão métodos que gerarão pontos no interior do poliedro para evitar as complexidades combinatórias dos métodos que vão de vértice em vértice; mas isso não foi melhor na prática que o Simplex.

O método Simplex reinou como único merecedor de atenção para resolver (P) até 1978, quando veio à luz o Algoritmo de Elipsóides de KHACHIYAN [38]. Este método de pontos interiores resolvia um problema teórico importante: mostrava que o problema (P) podia ser resolvido com um número de operações aritméticas limitado por um polinômio no "tamanho do problema", mas rapidamente se verificou que embora tivesse propriedades teóricas muito bonitas, o método de Elipsóides era irremediavelmente lento em problemas reais.

Devemos declarar que o tamanho do problema é dado por  $L = BT + n + 1$ , onde  $BT$  é o número total de bits utilizados pela entrada de dados do problema e  $n$  é a dimensão do espaço. Por vários anos, convivemos com a existência de duas técnicas: o método Simplex tinha uma complexidade exponencial (péssima) e um desempenho excelente em problemas do mundo real; e o método de Elipsóides.

Durante este tempo a pergunta mais importante foi: é possível construir um algoritmo tipo ponto interior que resolva a (P) em tempo polinomial e que na prática seja também melhor que o Simplex ? Em 1984 N. KARMARKAR [37] publicou seu algoritmo, baixando muito a complexidade teórica em relação ao método de Elipsóides. Ele obteve um limite para o número de iterações de  $O(nL)$ , e um limite para o número de operações de  $O(N^{3.5}L)$ . Ao mesmo tempo anunciou resultados computacionais superiores aos obtidos com o método Simplex. A maior dificuldade está na projeção de um vetor em cada iteração. Este é um clássico problema da análise numérica. Os avanços mais importantes abertos ao pessoal deve-se a um grupo de pesquisadores brasileiros: M. Resende, G. Veiga e M. Carvalho liderados por I. ADLER [1] da Universidade de Berkeley.

O algoritmo original de Karmarkar utilizava um formato especial para o problema, hipóteses muito restritivas e era muito difícil.

Em 1985 várias pessoas introduziram melhoramentos ao método e desenvolveram variantes para o problema em formato padrão, incluindo os seguintes pesquisadores (ver bibliografia): Anstreicher, Gay, de Ghellinck e Vial, Gonzaga, Todd e Burrell, Ye. Ao mesmo tempo uma simplificação radical do algoritmo gerou o método afim-escala, que se acredita não polinomial. Esse algoritmo já existia havia vinte anos, publicado por Dikin na União Soviética e era desconhecido, foi redescoberto simultaneamente por Vial, Barnes, por Vanderbei, Meketon e Freedman, Cavalier e Soyster, mas antes desses, já era certamente conhecido e utilizado por Karmarkar, que não o publicou.

O método de Karmarkar dependia da utilização de uma transformação

não linear conhecida como: "Transformação Projetiva", que se mostrou equivalente à utilização do cone gerado pelo conjunto viável (ver GONZAGA [26]), também, GONZAGA [28] mostrou que essa transformação não é necessária para obter o comportamento polinomial.

Em geral, as metodologias de pontos interiores podem ser classificadas em quatro grandes macro-categorias:

### I.1.1 Métodos Projetivos

### I.1.2 Métodos Afim-Escala

### I.1.3 Métodos de Trajetória Central

### I.1.4 Métodos de Redução Potencial

#### I.1.1 – Métodos Projetivos

Os algoritmos projetivos podem ser resumidos como segue: em cada iteração tem-se um ponto viável sobre o qual se faz uma mudança de escala do problema, depois é chamado um algoritmo interno que constroi uma função homogênea de grau zero e otimiza esta sobre o cone gerado pelo conjunto de soluções viáveis. YE-KOJIMA [56] propôs um algoritmo projetivo primal usando variáveis duais. TODD-YE [52] propôs um projetivo primal-dual usando uma classe de funções potenciais para primal e para dual.

#### I.1.2 – Métodos Afim-Escala

Nesta categoria não se conhecem algoritmos polinomiais que resolvam o problema, MEGIDDO-SHUB [42] mostrou que estes métodos podem requerer um número exponencial de iterações se partirmos de um ponto muito perto da

fronteira do conjunto de soluções viáveis. VANDERBEI [55] também mostrou que estes métodos podem ter dificuldades para reconhecer soluções duais e para provar otimalidade. No caso de degeneração. Curioso é que estes métodos funcionam muito bem na prática. MONTEIRO [45] apresentou um método afim primal-dual polinomial; também, GONZAGA-TODD [25] obtém um algoritmo polinomial "passos longos" afim primal-dual.

O nome afim deve-se a que os pontos interiores gerados estão no conjunto de soluções viáveis, em contraste com os métodos projetivos que geram pontos em uma região maior. Todos os algoritmos usam mudanças de escala (isto equivale a usar regiões de confiança Elipsoidais) e depois se faz uma busca na direção de máximo declive sobre o conjunto de soluções viáveis. Em geral os métodos afim-escala tem um problema muito grave: não poder evitar a proximidade a fronteira da região em questão; existem exemplos onde os elipsóides ficam muito compridos e impossível poder evitar a proximidade à pontos da fronteira da região.

A pergunta natural agora é: porque isto não é bom ? A resposta é porque se quer simultaneamente andar o mais longe possível e reduzir o custo o mais rápido possível; só longe da fronteira é que se consegue ambas as coisas. É evidente que com o tempo temos que aproximar-nos da fronteira, o mais perigoso é a redução de muitas variáveis que ao final serão diferentes de zero (variáveis básicas na solução ótima).

### **1.1.3 – Métodos de Trajetória Central**

Resultados essencialmente novos foram obtidos a partir de 1986, com o estudo de métodos de trajetória central. Um ponto é central se maximiza o

produto das variáveis em um conjunto de pontos viáveis com o custo constante. Para cada valor do custo, obtém-se um ponto central; e esses pontos compõem uma curva conhecida como trajetória central. A trajetória foi inicialmente estudada por BAYER-LAGARIAS [5], e por MEGIDDO [43].

RENEGAR [48] desenvolveu um algoritmo do tipo método de centros que segue a trajetória central, obtendo pela primeira vez um limite de complexidade de  $O(\sqrt{n} L)$  iterações, mas com complexidade em número de operações aritméticas igual à de Karmarkar. Em 1987, VAIDYA [53], seguindo a metodologia de Renegar, reduziu este limite para  $O(n^2 L)$ . Por um caminho independente GONZAGA [27] obteve o mesmo limite simultaneamente, utilizando um algoritmo de tipo penalidades. Novos algoritmos foram gerados a partir destes, incluindo os de KOJIMA, MIZUNO e YOSHISE [39], de MONTEIRO-ADLER [45]. Fizeram-se extensões para programação quadrática convexa, obtendo-se os mesmos limites de complexidade. Entre os anos 1989 e 1990 a atividade neste campo foi muito grande, surgem novos algoritmos que procuram acompanhar a trajetória central por "passos longos" ao invés dos "passos curtos" que são essenciais para as demonstrações de complexidade da ordem de  $\sqrt{n} L$ .

Quase todos os algoritmos existentes nesta categoria seguem de perto a trajetória central. Nesses algoritmos, cada iteração gera um ponto muito próximo à trajetória central, e cada ponto está próximo do ponto seguinte. Dessa forma, todos os passos do algoritmo estarão limitados a distância da ordem de 0,1 (passos curtos). Resulta que estes algoritmos são ineficientes na prática. Para aliviar tal dificuldade, surgem nesta categoria os algoritmos que seguem a trajetória com passos aumentados (passos longos); permite-se que a seqüência

de pontos afaste-se da trajetória, voltando a ela de maneira controlada. Os algoritmos de passos longos tem a mesma estrutura dos algoritmos de passos curtos, varia-se o valor do parâmetro que parametriza a trajetória central e usa-se o método de Newton para reduzir o valor da função auxiliar associada ao problema. A diferença está em que as variações do parâmetro são muito maiores e permitem-se várias iterações Newton entre cada duas atualizações do parâmetro.

#### 1.1.4 – Método de Redução Potencial

Muitos dos algoritmos desta categoria podem ser vistos como algoritmos de trajetória central passos longos. Entre eles está: o método de redução potencial primal-dual de YE [57] e FREUD [13]. Estes algoritmos estão baseados em reduções da função potencial primal-dual dada por:

$$f(x, z) = q \ln x^T z - \sum_{j=1}^n \ln x_j z_j$$

onde  $q$  é um real positivo. As variáveis duais são recalculadas se o ponto da iteração primal está perto da trajetória central. Os métodos de redução potencial primal trabalham com a função potencial primal, definida por:

$$f_v(x) = q \ln(c^T x - v) - \sum_{j=1}^n \ln x_j$$

onde  $v$  é um limitante inferior do custo ótimo do problema em formato primal dado por:

$$\begin{array}{ll} \text{Minimizar } c^T x & \\ \text{sujeito a } & Ax = b \\ & x \geq 0 \end{array}$$

O primeiro método de redução potencial primal é o algoritmo de Karmarkar; com uma atualização TODD-BURELL [51] do limitante inferior obtém-se um algoritmo que não segue a trajetória central. GONZAGA [21] propôs um algoritmo de trajetória central passos longos baseado na função potencial primal. Em cada iteração se incrementa o valor de  $v$  e se toma um passo para reduzir a função potencial. Devemos declarar que o procedimento de atualização de limitantes Todd-Burell está baseado em argumentos primal-dual e só é aplicado em algoritmos projetivos. GONZAGA [22] provou que a mesma complexidade dos algoritmos tipo Karmarkar pode ser alcançada usando  $q \geq n + \sqrt{n}$ ,  $q = O(n)$  na função potencial primal. No algoritmo de GONZAGA [22] se atualiza o limitante inferior quando o ponto está perto da trajetória central; neste caso, existe uma variável de folga dual com a qual é possível fazer a atualização do limitante inferior.

A diferença entre os métodos de redução potencial primal-dual e primal não é tão clara, posto que em ambos os casos a atualização do limitante inferior está baseada diretamente em um argumento primal-dual, tanto os procedimentos de atualização de Todd-Burell como os de Gonzaga. Os algoritmos de Ye usam variáveis duais explicitamente, mas elas são simples ferramentas para gerar limitantes inferiores e para o desenvolvimento das provas de complexidade. Em [22], Gonzaga chama de métodos primais a todos aqueles que não guardam em memória variáveis duais entre uma iteração e outra. Em contraste com isto, ele chama de métodos primal-dual à aqueles que trabalham tanto no espaço primal como no espaço dual, isto tem um passo primal e um

passo dual; por exemplo, o algoritmo de KOJIMA-MIZUNO-YOSHISE [39], GONZAGA-TODD [25].

## 1.2 – PONTOS INTERIORES E PROBLEMAS COMBINATÓRIOS

### 1.2.1 – Introdução

Sem dúvida que uma das aplicações mais importantes da programação linear (e portanto de pontos interiores) apresenta-se nos problemas combinatórios. Existem problemas, tais como o Problema de Matching Perfeito (PMP) (o qual estudaremos no Capítulo IV), que podem ser completamente descritos através de um sistema finito (mais muito grande) de desigualdades lineares. Fazendo uso de técnicas sofisticadas de poliedros combinatórios é possível identificar e gerar "planos cortantes".

Os métodos de planos cortantes (ver 1.2.3) são muito usados para resolver problemas de programação inteira. Na estrutura básica da metodologia tem-se que resolver uma seqüência de problemas de programação linear correspondentes as relaxações do problema original. Classicamente estes PL são resolvidos usando o conhecido Simplex.

O problema central nesta dissertação é sobre o uso dos métodos pontos interiores em a metodologia de planos cortantes para resolver o PMP. MITCHELL-TODD [44] resolvem o PMP fazendo uso de uma variante do algoritmo de Karmarkar na determinação de soluções dos problemas relaxados, mas eles apresentam sérias dificuldades: cada vez que um novo corte é encontrado, o ponto da iteração presente se faz inviável e uma inicialização eficiente do algoritmo ponto interior é dificultoso. No Capítulo III nós aliviamos

tal dificuldade, esta é a maior contribuição na dissertação.

Outra aplicação de pontos interiores a problemas combinatórios, e com uma visão totalmente diferentes a nossa, é dada por KARMARKAR, RESENDE & RAMAKRISHMAN [36], eles apresentam um método de região de confiança aplicado a uma função potencial associada ao problema de viabilidade 0-1.

### 1.2.2 – Alguns Elementos da Análise Convexa

Seguidamente daremos um conjunto de conceitos básicos da análise convexa. Nossas referências são: ROCKAFELLAR [49] e SCHRIJVER [50].

1.2.2.1 – Um sub-conjunto  $M \subseteq \mathbb{R}^n$  se diz **afim** se  $(1 - \lambda)x + \lambda y \in M$ , para cada  $x, y \in M$  e  $\lambda \in \mathbb{R}$ . Um conjunto  $M$  se diz **paralelo** a um conjunto afim  $L$  se  $M := L + \alpha$  para algum  $\alpha \in \mathbb{R}^n$ .

A **dimensão de um conjunto afim** é definida como a dimensão do único sub-espço paralelo a este. Os conjuntos afim de dimensão 0, 1 e 2 são chamados pontos, linhas e planos. Os conjuntos afim de dimensão  $(n-1)$  são chamados **hiperplanos**.

1.2.2.2 – A **envoltura afim** de um conjunto  $S$  em  $\mathbb{R}^n$  é a interseção de todos os conjuntos afim que contém a  $S$ ; isto é denotado com  $\text{AFF}(S)$ . Um conjunto de  $(n + 1)$  pontos,  $v_0, v_1, \dots, v_n$  se diz **afim independentes** se  $\text{AFF}\{v_0, v_1, \dots, v_n\}$  tem dimensão  $n$ .

Note que qualquer hiperplano contém  $n$  pontos afim independentes e  $n$  pontos afim independentes definem um hiperplano (único).

**I.2.2.3** — Um subconjunto  $C$  de  $\mathbb{R}^n$  se diz **convexo** se os pontos  $\lambda x + (1-\lambda) y \in C$ ; sempre que  $x, y \in C$  e  $0 < \lambda < 1$ . Para qualquer  $a \in \mathbb{R}^n$  (não-nulo) e qualquer  $\beta \in \mathbb{R}$ , o conjunto:

$$\{y: a^T y \leq \beta\}$$

é chamado **semi-espaço fechado**. Note que este conjunto é convexo.

A interseção de um número finito de semi-espaços fechados se denomina **poliedro**. Um poliedro limitado se denomina **politopo**.

O vetor soma  $\lambda_1 y_1 + \dots + \lambda_n y_n$  é chamado **combinação convexa** dos pontos  $\{y_1, \dots, y_n\}$  se todos os coeficientes  $\lambda_i$  são não-negativos e  $\sum_{i=1}^n \lambda_i = 1$ .

Um **ponto**  $y$  se diz **extremo** do conjunto convexo  $C$ , se  $y \in C$  e  $y$  não é combinação convexa de pontos em  $C$ .

**I.2.2.4** — Dado um conjunto de pontos  $S$  em  $\mathbb{R}^n$ , a **envoltura convexa de  $S$**  está definida como a interseção de todos os conjuntos convexos que contém a  $S$ . Isto será denotada com  $\text{CONV}(S)$ .

A envoltura convexa de  $S \subset \mathbb{R}^n$  é o conjunto convexo mais pequeno que contém a  $S$ . Note que este é único. A envoltura convexa de um subconjunto finito  $\{v_1, \dots, v_n\}$  é o conjunto de todos os pontos da forma:  $\lambda_1 v_1 + \dots + \lambda_n v_n$  com  $\sum_{i=1}^n \lambda_i = 1$  e todos os  $\lambda_i$  são não-negativos,  $i = 1, \dots, n$ .

O conjunto de pontos extremos desta envoltura convexa é um subconjunto de  $\{v_1, \dots, v_n\}$ , e a envoltura convexa deste é um poliedro.

**I.2.2.5** — Seja  $Ax \leq b$  um sistema viável de desigualdades lineares e definida  $P := \{x : Ax \leq b\}$ ; de aí que  $P$  é um poliedro. Seja  $c$  um vetor não-nulo e defina:

$$\delta := \min\{c^T x : x \in P\}$$

Então o hiperplano  $\{x : c^T x = \delta\}$  é chamado **hiperplano suporte** de  $P$ .

Um subconjunto  $F$  de  $P$  é chamado uma **face** de  $P$  se  $F = P$  ou  $F$  é a interseção de  $P$  com um hiperplano suporte. A face se diz **própria** se esta não é igual a  $P$ .

Uma **faceta** (facet) de  $P$  é uma face própria maximal (maximal relativa à inclusão).

### **I.2.3 – Método de Planos Cortantes**

O problema geral de programação matemática pode ser escrito como:

Minimizar  $\psi(x)$

sujeito a  $x \in X$

onde  $X$  é um conjunto e  $\psi : X \rightarrow \mathbb{R}$ . Para os problemas de **otimização combinatória** o conjunto  $X$  de soluções viáveis é finito; por exemplo,  $X$  pode ser uma coleção de conjuntos, ou vetores inteiros, ou permutações ou grafos. No que se segue,  $\psi$  será uma função linear do tipo  $\psi(x) = c^T x$ , onde  $c$  é um vetor de

custo dado sobre os elementos de  $X$ .

Se  $X$  é uma coleção de subconjuntos de um conjunto dado  $E$ , então  $X$  pode ser envolvido no espaço euclidiano  $\mathbb{R}^{|E|}$  ao considerar os vetores incidentes destes subconjuntos. Em geral, os elementos de  $X$  não são conhecidos explicitamente, mais eles são as soluções viáveis de um conjunto de restrições. Um **problema de programação linear inteira** é um problema de programação linear com a restrição de que as variáveis (ou algumas delas) sejam inteiras, isto é:

$$\begin{aligned} &\text{Minimizar } c^T x \\ &\text{sujeito a } \quad Ax \leq b \\ &\quad \quad \quad x_i \text{ inteira, } i \in I \end{aligned}$$

onde  $I$  é um conjunto de índices das variáveis. Se todas as variáveis são inteiras, o problema anterior é um **problema combinatório**.

Considere o problema combinatório dado por:

$$\begin{aligned} &\text{Minimizar } c^T x \\ &\text{sujeito a: } x \in X \subseteq \mathbb{R}^n \end{aligned} \tag{1}$$

onde  $X$  é um conjunto finito. Considere também o problema de:

$$\begin{aligned} &\text{Minimizar } c^T x \\ &\text{sujeito a: } x \in \text{CONV}(X) \end{aligned} \tag{2}$$

Note que como  $X$  é um conjunto finito, então  $\text{CONV}(X)$  é um polítopo e o problema anterior é um problema de programação linear. Segue-se que se  $\hat{x}$  é uma solução ótima do problema (1) então este é também uma solução ótima de (2); ao contrário, se  $\hat{x}$  é uma solução ótima do problema (2) então existe uma solução viável do problema (1) com o mesmo valor. Portanto em princípio, o problema (1) pode ser resolvido, resolvendo-se o problema (2). A maior dificuldade é que o poliedro  $\text{CONV}(X)$  não está descrito em termos de equações e desigualdades lineares.

Em muitos casos, o número de equações e desigualdades lineares necessárias para uma descrição completa de  $\text{CONV}(X)$  é exponencial no tamanho do problema.

Uma forma de atacar o problema é considerando relaxações do problema (2). Usualmente, estas relaxações são problemas de programação linear para os quais tem-se uma descrição completa em termos de sub-espacos fechados; a relaxação tem a forma:

$$\begin{aligned} &\text{Minimizar } c^T x \\ &\text{sujeito a } x \in P \end{aligned} \tag{3}$$

onde  $\text{CONV}(X) \subseteq P$  e  $P$  é um poliedro.

Note que se uma solução ótima  $\hat{x}$  de (3) está no conjunto  $X$ , então  $\hat{x}$  é uma solução ótima de (1). Se  $P$  é uma boa aproximação de  $\text{CONV}(X)$  na vizinhança de uma solução ótima do problema (1), então este ponto deve ser

também uma solução ótima do problema (3). Isto último sugere um procedimento iterativo baseado em planos cortantes.

### Procedimento Planos Cortantes (GOMORY [17])

#### Passo 0

Escolher um poliedro  $P^0 \supset X$

Fazer  $k := 0$

#### Passo 1

Encontrar uma solução ótima  $\hat{x}$  do problema:

Minimizar  $c^T x$

sujeito a:  $x \in P^k$

#### Passo 2

Se  $\hat{x} \in \text{CONV}(X)$  então parar.

#### Passo 3

Encontrar  $\pi \in \mathbb{R}^n$  e  $\pi_0 \in \mathbb{R}$  tal que: o semi espaço fechado  $H^k := \{x \in \mathbb{R}^n; \pi^T x \leq \pi_0\}$  satisfaz:

- i)  $X \subseteq H^k$
- ii)  $\hat{x} \notin H^k$  (i.e.,  $\pi^T \hat{x} > \pi_0$ )

**Passo 4**

Fazer  $P^{k+1} := P^k \cap H^k$

**Passo 5**

Fazer  $k := k + 1$  e voltar ao Passo 1.

O procedimento começa com um poliedro inicial  $P^0 \subset \mathbb{R}^n$  descrito por  $A_0x \leq b_0$ .

Restrições novas no problema inicial da forma  $\pi^T x \leq \pi_0$  são agregadas, e depois de  $k$ -iterações temos o poliedro  $P^k$  dado por  $A_kx \leq b_k$ . No passo 1 resolvemos:

Minimizar  $c^T x$   
 sujeito a:  $A_kx \leq b_k$

Considere agora o critério de parada do Passo 2, se o método Simplex é usado para resolver o problema relaxado, então o ponto  $\hat{x}$  é um ponto extremo de  $P^k$  e portanto se este pertence a  $\text{CONV}(X)$  então este deve ser um elemento de  $X$ .

Se algum método pontos interiores é usado para resolver o problema relaxado então o ponto  $\hat{x}$  pode não pertencer a  $X$  mesmo que  $\hat{x} \in \text{CONV}(X)$ . Em tal situação, existem pontos em  $X$  que tem o mesmo valor que  $\hat{x}$ , e existem formas de "arredondar" (Capítulo IV) o ponto  $\hat{x}$  para encontrar tal ponto em  $X$ .

Os procedimentos tipo Gomory mostram a ação de uma rotina de separação. Tal rotina tem como entrada um ponto  $\hat{x}$  e um conjunto de pontos  $X$ ; como saída, este confirma que  $\hat{x}$  pertence a  $\text{CONV}(X)$ , ou encontra um hiperplano que separa  $\hat{x}$  de  $\text{CONV}(X)$ . O poliedro  $P^0$  está descrito por desigualdades lineares, portanto que a atualização do poliedro no passo 4 é trivial, somente agregamos uma nova desigualdade.

GOMORY [17] sugere uma forma particular de escolher os semi-espacos fechados no passo 3 com o qual garante a convergência em um número finito de iterações.

Na prática o procedimento de Gomory não é eficiente, isto porque os cortes agregados não são "profundos"; melhores cortes resultam hiperplanos suportes de  $\text{CONV}(X)$ . Nos últimos anos se conhecem muitos trabalhos que encontram facetas (estes são os melhores cortes) do poliedro associado ao problema combinatório em questão. Estas pesquisas podem ser usadas na metodologia de planos cortantes; mas para que tal conhecimento possa ser incorporado em um algoritmo eficiente, é necessário que a rotina de separação retorne facetas como hiperplanos de separação se o ponto não está em  $\text{CONV}(X)$ . Em muitos casos, as rotinas de separação são heurísticas (nosso caso para o problema de matching perfeito, ver Capítulo IV) e por isso não existe garantia de que estas retornem hiperplanos de separação, mesmo que uma das facetas conhecida seja tal hiperplano. Neste caso, para resolver o problema é necessário usar um branch and bound. Este envolve a partição em dois da região de soluções viáveis, e o encontro do melhor ponto em cada uma das sub-regiões, a melhor entre as duas será a melhor do problema geral. Existem muitas descrições de branch and bound, este método foi desenvolvido por LAND-DOIG [40], para mais informação recente do método ver

PAPADIMITRIOU-STEIGLITZ [47] e SCHRIJVER [50].

### 1.3 – ORGANIZAÇÃO DA TESE

A organização desta dissertação é como segue:

No Capítulo II consideramos o problema de programação linear no formato dual como o problema inicial. A eleição deste formato deve-se em primeiro lugar a facilitar a geometria desenvolvida no Capítulo III para resolver o problema central (reiniciação); em segundo lugar, permite mantermos compatível com o formato do problema relaxado do problema de matching perfeito no Capítulo IV; sobre ele são aplicados todos os resultados algorítmicos do Capítulo III.

O problema inicial é resolvido com um algoritmo de trajetória central de passos longos; este resulta uma variante do algoritmo de HERTOOG-ROOS-TERLAKY [10]. A teoria desenvolvida associa ao problema uma família de problemas fáceis de resolver. As soluções ótimas destes problemas convergem a uma solução ótima do problema inicial. A função auxiliar associada ao problema é do tipo **centro**, ver [48].

Uma caracterização da trajetória central (parametrizada com limitante superior do custo ótimo) é dada, e propriedades importantes primais-duais são apresentadas.

O conceito de proximidade à trajetória central é também dado, e um conjunto de propriedades ligadas à complexidade do algoritmo para pontos suficientemente pertos da trajetória central são mostradas.

O Capítulo II termina mostrando como obter o ponto inicial do algoritmo que resolve o problema original.

O Capítulo III é o capítulo mais importante da tese; nele resolvemos o problema que qualquer algoritmo de pontos interiores enfrenta dentro de um método de planos cortantes que procura resolver determinado problema de programação inteira; isto é, se o problema relaxado do problema inteiro é resolvido usando um algoritmo tipo ponto interior; quando novos cortes sejam agregados à relaxação, o ponto interior que era solução ótima aproximada do problema, já não será um ponto interior da nova região, e um procedimento polinomial que reinicialize eficientemente ao algoritmo deve ser apresentado.

Neste capítulo de reotimização com metodologia pontos interiores começamos considerando em primeiro lugar o caso de agregação de uma restrição ao problema original.

Para recuperar informações valiosas, relaxamos a restrição agregada e consideramos esta com um número conveniente de cópias. De aí temos propriedades importantes da trajetória central associada ao problema com determinado número de cópias. Para este caso também damos o conceito de proximidade à trajetória central, e baixo certas condições provamos que o ponto que inicialmente resolvia a relaxação inicial do problema inteiro está próximo da nova trajetória central.

Como ao final desejamos obter um ponto interior perto da trajetória central associada ao problema perturbado com exatamente uma cópia, nós estudamos a trajetória central associada à relaxação do problema perturbado

com uma cópia. De aí obtemos um resultado muito importante que relaciona o conceito de proximidade com uma e mais cópias da restrição relaxada. Um resultado central deste capítulo é dado na seção III.2.3; onde provamos que para cada ponto primal viável com proximidade baixa existe uma solução dual viável de gap baixo. Este resultado é aplicado depois na seção III.6.

Na seção III.2.4 estudamos o alcance a região perturbada e a condição de parada do algoritmo polinomial que reinicializa o problema. O alcance é definido como uma distância à região perturbada desde o ponto em questão. Mais ainda, seu cálculo no algoritmo reinicialização permite a tomada de decisão para a atualização do limitante superior do custo.

Na seção III.3 apresentamos o algoritmo reinicialização com suas características gerais, dados e procedimentos. Na seção III.4 se faz todo o referente com a complexidade do algoritmo reinicialização. Finalmente na seção III.5 resolve-se o problema de reinicialização com agregação de um conjunto de restrições, e na seção III.6 o caso de agregação de novas variáveis ao problema.

O Capítulo IV é o capítulo de aplicação dos resultados dados nos Capítulos II e III. O bem conhecido problema combinatório de matching perfeito é o escolhido. Este problema é um dos problemas fundamentais da otimização combinatória, e Jack Edmonds foi o primeiro em provar que o problema é polinomial; mostrando um algoritmo combinatório de ordem cúbica para sua resolução.

Uma outra metodologia para resolver o problema de matching perfeito (PMP) é planos cortantes. Neste capítulo nós apresentamos um algoritmo de planos cortantes para resolver o PMP, o qual usa uma variante tipo trajetória

central passos longos para resolver o problema relaxado.

Na seção IV.2 formulamos o problema, como um problema em grafos não-dirigidos e como um problema de programação linear fazendo uso da caracterização da envoltória convexa dada por J. Edmonds. Seguidamente enunciamos o problema de separação para o PMP como também o algoritmo polinomial de Padberg e Rao para sua resolução.

Na seção IV.3 desenvolve-se a obtenção de uma solução viável inicial do PMP, a relaxação inicial do problema, o arredondamento de soluções, as rotinas de separação e a descrição esquemática do algoritmo.

## CAPÍTULO II

UMA VARIANTE DO ALGORITMO DE HERTOG-ROOS-TERLAKY  
 PARA RESOLVER O PROBLEMA INICIAL

## II.1 - INTRODUÇÃO

Neste capítulo apresentamos um algoritmo polinomial tipo ponto interior para resolver o problema de Programação Linear (PL). O algoritmo é uma variante do algoritmo de HERTOG-ROOS-TERLAKY [10], este pode ser classificado como um algoritmo de trajetória central passos longos (ver GONZAGA [29]). O problema a considerar é um PL no formato dual, isto é:

$$\text{Minimizar } c^T x \quad (P)$$

$$\text{Sujeito a: } Ax \leq b$$

onde  $c \in \mathbb{R}^n$ ;  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ .

O método associa a (P) a função tipo centro dada por:

$$f_k(x) := -q \ln(k - c^T x) - \sum_{i=1}^m \ln(b_i - a_i x);$$

onde  $q$  é um inteiro positivo,  $q \geq m$ , e  $k > c^T x$ ,  $Ax < b$ .

Se  $q = m$ ,  $f_k(\cdot)$  é a mesma função que considera J. RENEGAR [48].

Baseados nesta função centro, o algoritmo realiza buscas lineares ao largo de direções Newton.

O capítulo está organizado como segue:

Primeiro consideramos o PL e olhamos este como um problema de programação não-linear. A obtenção de uma direção Newton e o conceito de proximidade são dados. Finalmente, apresentamos o algoritmo início com sua complexidade e a obtenção do ponto inicial.

## II.2 – O PROBLEMA INICIAL

Considere o problema de programação linear:

$$\text{Minimizar } c^T x \tag{P}$$

$$\text{Sujeito a: } Ax \leq b$$

onde  $c \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $m > n$ .

Suponha que  $A$  é de rank completo. Note que (P) pode ser escrito como:

$$\text{Minimizar } c^T x \tag{P_{=}}$$

$$\text{Sujeito a: } Ax + z = b$$

$$z \geq 0$$

$z$  denomina-se **folga primal**.

### Notação

Seja  $S := \{x \in \mathbb{R}^n: Ax \leq b\}$  e  $S^0$  o interior de  $S$ .

Seja  $S_{\underline{}} := \{(x, z) \in \mathbb{R}^n \times \mathbb{R}^m; Ax + z = b, z \geq 0\}$  e  $S_{\underline{}}^0$  o interior relativo de  $S_{\underline{}}$ .

Suponha que  $S^0 \neq \emptyset$  e  $S_{\underline{}}^0 \neq \emptyset$ .

O problema dual associado a (P) é:

$$\text{Maximizar } -b^T y \quad (D)$$

$$\begin{aligned} \text{Sujeito a: } \quad & A^T y + c = 0 \\ & y \geq 0 \end{aligned}$$

**note que:** dados  $x, z$  viáveis em (P) e  $y$  viável para (D).

$$\begin{aligned} \text{GAP} &= c^T x - (-b^T y) = c^T x + b^T y = c^T x + (Ax+z)^T y = c^T x + x^T A^T y + \\ &+ z^T y = z^T y \end{aligned}$$

#### II.2.1 – Observação

Associe a (P) a função tipo centro:

$$x \in S^0, k > c^T x \rightarrow f_k(x) := -q \ln(k - c^T x) - \sum_{i=1}^m \ln(b_i - a_i x)$$

$f_k(\cdot)$  corresponde a função barreira com  $c^T x < k$  repetida  $q$ -vezes, onde  $k$  é uma cota superior do custo ótimo dada,  $q$  um inteiro positivo,  $q \geq m$  e  $a_i = (a_{i1}, \dots, a_{in})$ ,  $i = 1, \dots, m$ .

$$\text{Seja } S_k := S \cap \{x \in \mathbb{R}^n: c^T x \leq k\}$$

Note que  $f_k(\cdot)$  é estritamente convexa sobre  $S_k^0$ . Suponha que  $S_k^0 \neq \emptyset$ .

Associe a (P) a família de problemas:

Minimizar  $f_k(x)$

(P<sub>k</sub>)

Sujeito a:  $x \in S_k^0$

É bem sabido que a seqüência de soluções dos problemas (P<sub>k</sub>) com um decréscimo de  $k$  tende a uma solução ótima de (P); mais ainda, para cada  $k$ , (P<sub>k</sub>) tem uma única solução que denotamos com  $x(k)$ ; isto é:

$$x(k) := \text{ARGMIN}\{f_k(x): x \in S_k^0\} \quad (\text{II.1})$$

## II.2.2 – Definição

A curva  $k \in \mathbb{R} \rightarrow x(k)$  é a **trajetória central** associada ao problema de programação linear (P).  $x(k)$  denomina-se **ponto central**.

A curva é suave e tem propriedades importantes.

### II.2.3 – Proposição

Dado  $k \in \mathbb{R}$ , se  $x = x(k)$  satisfaz (II.1) então:

$$Yz = \frac{(k - c^T x)}{q} e$$

$$Ax + z = b, z > 0 \tag{II.2}$$

$$A^T y + c = 0, y > 0$$

$$c^T x < k$$

onde  $Y := \text{diagonal}(y_1, \dots, y_n)$ ,  $e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^m$ .

#### Prova

Pelas condições de Karush–Kuhn–Tucker, se  $x = x(k)$  satisfaz (II.1) então

$$\frac{q}{k - c^T x} c + \sum_{i=1}^m \frac{1}{b_i - a_i x} a_i^T = 0 \tag{II.3}$$

$$x \in S_k^0 \tag{II.4}$$

De (II.3) tem-se que:

$$\sum_{i=1}^m \frac{(k-c^T x)}{q(b_i - a_i x)} a_i^T + c = 0 \quad (II.5)$$

Fazendo  $y_i = \frac{(k-c^T x)}{q(b_i - a_i x)} > 0$ ; ou,  $Yz = \frac{(k-c^T x)}{q} e$ , onde  $Y = \text{diagonal}(y_1, \dots, y_m)$  e  $e^T = (1, \dots, 1) \in \mathbb{R}^m$ ; temos que (II.5) pode ser escrita como:

$$\sum_{i=1}^m a_i^T y_i + c = 0, \text{ ou,}$$

$$A^T y + c = 0, y > 0$$

Finalmente, se  $x = x(k)$  satisfaz (II.1) então  $x = x(k)$  satisfaz (II.2).

## II.3 – OBTENÇÃO DE UMA DIREÇÃO NEWTON

### II.3.1 – Cálculos

O gradiente e a matriz hessiana da função  $f_k(\cdot)$  são dados por:

$$g = g(x, k) := \nabla f_k(x) = \frac{q}{k-c^T x} c + \sum_{i=1}^m \frac{1}{b_i - a_i x} a_i^T$$

$$H = H(x, k) := \nabla^2 f_k(x) = \frac{q}{(k-c^T x)^2} cc^T + \sum_{i=1}^m \frac{1}{(b_i - a_i x)^2} a_i^T a_i$$

Note que  $H$  é definida positiva.

### II.3.2 – Direção

O método Newton para resolver  $(P_k)$  considera a série de Taylor do gradiente de  $f_k(x)$  no mínimo  $x(k)$  em torno a  $x$ , isto é:

$$g(x(k), k) = g(x, k) + H(x, k) (x(k) - x) + \dots$$

Seja  $x^*$  o mínimo da aproximação quadrática de  $f_k(x)$ . Então

$$g(x, k) + H(x, k) (x^* - x) = 0; \text{ ou,}$$

$$(x^* - x) = -H^{-1}(x, k) g(x, k)$$

Em seguida faz-se uma busca linear ao longo da direção Newton definida por:

$$h(x, k) = -H^{-1}(x, k) g(x, k) = -H^{-1}g$$

### II.3.3 – Proximidade

#### II.3.3.1 – Definição

Dado  $x \in \mathbb{R}^n$  a  $H$ -norma de  $x$  denotada por  $\|x\|_H$  é definida como:

$$\|x\|_H := \sqrt{x^T H x}$$

Note que esta norma está bem definida posto que  $H$  é definida positiva.

### II.3.3.2 – Definição

Dado  $k \in \mathbb{R}$  e  $x \in S_k^0$ . A proximidade de  $x$  em relação a  $x(k)$  é caracterizada por:

$$\delta = \delta(x, k) := \|h(x, k)\|_H$$

O ponto será considerado **aproximado central** se  $\delta(x, k) \leq \epsilon$ ,  $\epsilon \in (0, 1)$ . Neste caso diz-se que  **$x$  está próximo da trajetória central**.

Note que  $\delta(x, k) = 0$  se e somente se  $x = x(k)$ .

## II.4 – O ALGORITMO INÍCIO

### II.4.1 – Características Gerais

O algoritmo início que resolve o problema (P) é uma variante do algoritmo de HERTOOG–ROOS–TERLAKY [10] este começa com um ponto próximo da trajetória central (esta suposição será aliviada mais tarde). Em uma iteração qualquer se atualiza o limitante superior  $k$  do custo ótimo e se realiza um procedimento interno composto de uma direção Newton e uma busca linear. O procedimento é repetido até que um novo ponto próximo da trajetória central é encontrado.

O procedimento geral é repetido até que algum critério de parada é satisfeito. A atualização do limitante superior  $k$  é como segue:

$$\text{Fazer } k' - c^T x = \beta(k - c^T x), \beta \in (0, 1); \text{ ou } k' = \beta k + (1-\beta) c^T x.$$

Note que  $k'$  é uma cota superior do valor ótimo  $V^*$ ; isto é imediato posto que:

$$c^T x < k, \text{ então}$$

$$\beta c^T x < \beta k, \text{ ou}$$

$$c^T x < \beta k + c^T x - \beta c^T x$$

$$= \beta k + (1 - \beta) c^T x$$

$$= k'$$

Portanto temos que  $V^* \leq c^T x < k'$ .

#### II.4.2 – Dados

$\beta \in (0,1)$  fator de redução de  $k$ .

$\sigma > 0$  parâmetro de precisão (parada).

**Nota:** O algoritmo pode parar antes, quando alguma heurística gerar um novo corte.  $\epsilon = 0.34$  tolerância da proximidade,  $k^0 \in \mathbb{R}$  cota superior do custo ótimo  $V^*$  tal que  $k^0 - V^* < 2^L$ .

$x^0 \in S_k^0$  tal que  $\delta(x^0, k^0) \leq \epsilon$ .

### II.4.3 – Procedimento Geral

$i = 0$

Repita

$$\text{Cota } \bar{k} = \beta k^i + (1-\beta) c^T x^i$$

$$j := 0; \xi^0 = x^i;$$

Repita

**Direção** calcular  $h(\xi^j, \bar{k}); \delta(\xi^j, \bar{k});$

**Busca** resolver aproximadamente

$$\bar{\lambda} = \underset{\lambda > 0}{\text{ARGMIN}} \{f_k(\xi^j + \lambda h): z - \lambda Ah > 0\}$$

**Novo ponto**  $\xi^{j+1} := \xi^j + \bar{\lambda}h$

$$j = j + 1$$

Até que  $\delta(\xi^j, \bar{k}) \leq 0.34$

$$x^{i+1} := \xi^j;$$

$$k^i = \bar{k};$$

$$i := i + 1;$$

Até que  $k^i - c^T x^i < \sigma$

## II.5 – COMPLEXIDADE DO ALGORITMO

### II.5.1 – Propriedades Primal-Dual

#### II.5.1.1 – Proposição

$z \in \mathbb{R}^m$  é folga primal viável se e somente se:

i)  $z \geq 0$

$$\text{ii)} \quad P_{A^T} z = P_{A^T} b$$

onde  $P_{A^T}$  é a matriz projeção sobre o espaço nulo de  $A^T$ .

### Prova

Seja  $z \geq 0$ , então  $z$  é uma folga primal viável se e somente se,  $b - z = Ax$  para algum  $x \in \mathbb{R}^n$ . Mas,  $b - z$  pode ser decomposto de uma única forma:

$$b - z = P_{A^T}(b - z) + Ax;$$

comparando temos que:

$$P_{A^T}(b - z) = P_{A^T}b - P_{A^T}z = 0.$$

### II.5.1.2 – Observação

Considere o problema dual:

Minimizar  $b^T y$

(D)

Sujeito a:  $A^T y + c = 0$

$$y \geq 0$$

Seja  $T := \{y \in \mathbb{R}^n; A^T y + c = 0, y \geq 0\}$

Suponha que o interior relativo de  $T$ ,  $T^\circ \neq \emptyset$ .

Analogamente ao feito para (P), associamos a (D) a função tipo centro:

$$y \in T^0, \rho > b^T y \rightarrow \hat{f}_\rho(y) = -q \ln(\rho - b^T y) - \sum_{i=1}^m \ln y_i;$$

e para cada  $\rho \in \mathbb{R}$  o problema:

$$\begin{aligned} &\text{Minimizar } \hat{f}_\rho(y) \\ & \hspace{15em} (D_\rho) \\ &\text{Sujeito a: } y \in T_\rho^0 \end{aligned}$$

onde  $T_\rho = T \cap \{y \in \mathbb{R}^n: b^T y \leq \rho\}$ ;  $T_\rho^0 \neq \emptyset$ .

As condições de Karush–Kuhn–Tucker aplicadas a  $(D_\rho)$  são exatamente as mesmas que descrevemos em II.2.3 para  $(P_k)$ .

O ponto  $y = y(\rho) = \text{ARGMIN}\{\hat{f}_\rho(y): y \in T_\rho^0\}$  denomina-se central.

### II.5.1.3 – Proposição

Seja  $y = y(\rho)$  um ponto central para algum  $\rho \in \mathbb{R}$ . Então

$$Z(\rho) = \frac{(\rho - b^T y)}{q} y^{-1}$$

é uma folga primal viável e o  $\text{GAP}(\rho) = \frac{m}{q} (\rho - b^T y)$

**Prova:**

Se  $y = y(\rho)$  é um ponto central então pelas condições de Karush–Kuhn–Tucker aplicadas a  $(D_\rho)$  temos que:

$$\frac{q}{\rho - b^T y} b - y^{-1} + Ax_1 = 0; x_1 \in \mathbb{R}$$

Fazendo  $x = -x_1 \in \mathbb{R}$ :

$$\frac{q}{\rho - b^T y} b - y^{-1} = Ax; \text{ de aí que } \frac{q}{\rho - b^T y} b - y^{-1} \in \text{IMAGEM}(A)$$

$Z(\rho) > 0$  e por II.5.1.

$Z(\rho)$  é folga primal viável; mais ainda:

$$\text{GAP}(\rho) = z^T(\rho) y(\rho) = \frac{(\rho - b^T y)}{q} (y^{-1}(\rho))^T y(\rho) = \frac{m}{q} (\rho - b^T y)$$

#### II.5.1.4 – Proposição (Lema 4.4 em GONZAGA [29])

Considere a função tipo centro  $f_k(\cdot)$  com  $q \geq m$  e o ponto central  $x = x(k)$  para  $k > V^*$  (o valor ótimo de (P)). Seja

$$k' = \beta k + (1-\beta) c^T x; \beta \in (0,1).$$

Então:

$$i) \quad \frac{k' - V^*}{k - V^*} \leq \frac{m/q + \beta}{1 + \beta} \leq 1 + \beta/2$$

$$\text{ii) } \text{GAP}(k) \leq k - V^* \leq \frac{m+q}{m} \text{GAP}(k)$$

Desta proposição concluímos que  $(k-V^*)$  está limitado suficientemente por uma seqüência que decresce uma quantidade fixa por iteração. A redução por iteração depende do valor de  $q$ .

## II.5.2 – O Lema Auxiliar

O lema seguinte é um resumo das propriedades associadas ao conceito de proximidade; a prova é análoga é feita por HERTOOG–ROOS–TERLAKY [10], ou GONZAGA [29].

### II.5.2.1 – Lema Auxiliar

Considere o parâmetro  $k > V^*$ , um ponto  $x \in S_k^0$  e sua proximidade  $\delta = \delta(x, k) = \|h(x, k)\|_H$ :

#### i) Eficiência do passo Newton

Se  $\delta \leq 1$  então  $\bar{x} = x + h(x, k)$  é viável e  $\delta(\bar{x}, k) \leq \delta^2$ .

#### ii) Proximidade e distância

Se  $\delta < \epsilon < 0.5$  então

$$\|x - x(k)\|_H \leq \frac{\epsilon(1-\epsilon)(1+2\epsilon)}{(1+\epsilon)(1-2\epsilon)}$$

## iii) Proximidade e função centro

$$\text{Se } \delta \leq \epsilon < 1 \text{ então } f_k(x) - f_k(x(k)) \leq \frac{\epsilon^2}{2(1-\epsilon)(1-\epsilon^2)}$$

## iv) Decréscimo da função centro na direção Newton

$$\bar{x} = x + \frac{1}{1+\delta} h(x, k) \text{ é viável e } f_k(\bar{x}) \leq f_k(x) - (\delta - \text{Ln}(1+\delta)).$$

## II.5.2.2 – Observação

No lema anterior, i e iv são os mais importantes. i diz que o passo Newton decresce em proporção quadrática desde qualquer ponto  $x$  tal que  $\delta(x, k) < 1$ . Isto significa que os algoritmos Newton reduzem a proximidade em uma seqüência de valores dada por:

$$\delta, \delta^2, \delta^4, \dots, \delta^{2^p}$$

ii e iii mostram que o conceito de proximidade está bem definido do ponto de vista geométrico.

iv dá a informação para os pontos que não são próximos à trajetória central. Se  $\delta(x, k)$  é grande, então uma iteração Newton garante um decréscimo da função objetivo.

### II.5.2.3 – Proposição

Se  $x \in S_k^0$  ( $k \in \mathbb{R}$ ) e  $k' = \beta k + (1-\beta) c^T x$  ( $\beta \in (0,1)$ ) então  
 $f_{k'}(x) - f_k(x) = -q \operatorname{Ln}(\beta) = q \operatorname{Ln}(1/\beta)$ .

**Prova**

$$\begin{aligned} f_{k'}(x) - f_k(x) &= -q \operatorname{Ln}(k' - c^T x) - \sum_{i=1}^m \operatorname{Ln}(b_i - a_i x) + q \operatorname{Ln}(k - c^T x) + \\ &+ \sum_{i=1}^m \operatorname{Ln}(b_i - a_i x) = -q \operatorname{Ln}\left(\frac{k' - c^T x}{k - c^T x}\right) = -q \operatorname{Ln}(\beta); \text{ pois } (k' - c^T x) = \\ &= \beta(k - c^T x). \end{aligned}$$

### II.5.2.4 – Proposição

Se  $\|x - x(k)\|_H \leq \theta$ . Então

$$\left(1 - \frac{\theta}{\sqrt{q}}\right) (k - c^T x(k)) \leq k - c^T x \leq \left(1 + \frac{\theta}{\sqrt{q}}\right) (k - c^T x(k))$$

**Prova**

Pela hipótese

$$\theta^2 \geq \|x - x(k)\|_H^2 = (x - x(k))^T H(x - x(k))$$

$$= (\mathbf{x} - \mathbf{x}(k))^T \frac{q \mathbf{c} \mathbf{c}^T}{(k - \mathbf{c}^T \mathbf{x}(k))^2} (\mathbf{x} - \mathbf{x}(k)) = \frac{q(\mathbf{c}^T \mathbf{x} - \mathbf{c}^T \mathbf{x}(k))^2}{(k - \mathbf{c}^T \mathbf{x}(k))^2}$$

Portanto que

$$-\frac{\theta}{\sqrt{q}} (k - \mathbf{c}^T \mathbf{x}(k)) \leq \mathbf{c}^T \mathbf{x} - \mathbf{c}^T \mathbf{x}(k) \leq \frac{\theta}{\sqrt{q}} (k - \mathbf{c}^T \mathbf{x}(k));$$

o que implica que:

$$\left(1 - \frac{\theta}{\sqrt{q}}\right) (k - \mathbf{c}^T \mathbf{x}(k)) \leq k - \mathbf{c}^T \mathbf{x} \leq \left(1 + \frac{\theta}{\sqrt{q}}\right) (k - \mathbf{c}^T \mathbf{x}(k)).$$

### II.5.3 – Teoremas Centrais

Baseados nos resultados anteriores, é possível encontrar cotas superiores para o número total de iterações do algoritmo início; tanto internas como externas. Suponha que  $q > 1$  e que a tolerância  $\epsilon$  da proximidade é igual a  $1/3$ .

#### II.5.3.1 – Deduções

i) Se o critério de proximidade não é satisfatório, então:

$$f_k(\mathbf{x}) - f_k(\hat{\mathbf{x}}) \geq 1/22$$

(ver iv em II.5.2.1).

ii) Se o critério de proximidade é satisfeito, então:

$$f_k(x) - f_k(x(k)) \leq 1/10 \text{ e } \|x - x(k)\|_H < 1$$

(ver ii e iii em II.5.2.1).

iii) Para  $\theta = 1$  temos que:

$$\left(1 - \frac{1}{\sqrt{q}}\right) (k - c^T x(k)) \leq k - c^T x \leq \left(1 + \frac{1}{\sqrt{q}}\right) (k - c^T x(k))$$

(ver II.5.2.4).

### II.5.3.2 – Teorema

O algoritmo início requer  $i = O\left(\frac{1}{1+\beta} L\right)$  iterações externas quando é usado para encontrar uma solução exata de (P).

#### Prova

Seja  $k^i$  a cota superior na  $i$ -ésima iteração, e  $x^i$  o ponto obtido ao final de  $i$  iterações externas.

Por i de II.5.1.4, temos que:

$$\frac{k^i - V^*}{k^{i-1} - V^*} \leq (1 + \beta)/2$$

Depois de  $i$ -iteraões externas tem-se:

$$c^T x^i - V^* \leq k^{i+1} - V^* \text{ (ver II.4.1)}$$

$$\leq \frac{(1+\beta)}{2} (k^i - V^*)$$

$$\leq \left(\frac{1+\beta}{2}\right)^i (k^0 - V^*)$$

então em cada iteração:

$$c^T x^i - V^* \geq 2^{-L}$$

é válida se

$$\left(\frac{1+\beta}{2}\right)^i (k^0 - V^*) \geq 2^{-L}$$

portanto

$$i \log\left(\frac{1+\beta}{2}\right) + \log(k^0 - V^*) \geq -L$$

e daí,

$$i \leq \frac{(L + \log(k^0 - V^*))}{-\log \frac{1+\beta}{2}} \leq \frac{L + \log(k^0 - V^*)}{(1 + \beta) \log 2}$$

Lembre que  $\log(1+\beta) \leq \beta$  ( $0 < \beta < 1$ )

$$i \leq \frac{(L + \log(k^0 - V^*))}{1 + \beta}$$

Agora por II.4.2;  $k^0 - V^* < 2^L$  e finalmente que:

$$i = O\left(\frac{1}{1+\beta} L\right).$$

### II.5.3.3 – Teorema

Seja  $J$  o número de passos internos em uma iteração do algoritmo início, então:

$$J \leq 3 + \frac{44q}{(\sqrt{q-1})} + 22q \left( \frac{1}{\beta(1-\beta)} - \beta \right)$$

#### Prova

Denotemos a cota superior de uma iteração externa qualquer com  $k'$ , e a cota superior da iteração anterior com  $k$ .

Denote com  $x^0, x^1, \dots, x^J$  os pontos gerados durante esta iteração externa,  $x^0$  é o ponto obtido no começo da iteração exterior.

De  $i$  em II.5.3.1 temos que:

$$f_{k'}(x^J) \leq f_k(x^0) - \frac{1}{22} J \quad (\text{II.6})$$

Como a cota superior é atualizada ao começo da iteração externa, então por II.5.2.3 temos que:

$$f_{k'}(x^0) - f_k(x^0) = -q \text{Ln}(\beta) \quad (\text{II.7})$$

mais ainda,

$$f_{k'}(x^J) - f_k(x^J) = -q \operatorname{Ln} \left( \frac{\beta(k - c^T x^0)}{(k - c^T x^J)} \right) + \frac{c^T x^0 - c^T x^J}{(k - c^T x^J)}$$

$$\geq -q \operatorname{Ln} \left( \frac{\beta(k - c^T x^0)}{(k - c^T x^J)} + 1 \right)$$

e daí:

$$-f_{k'}(x^J) \leq -f_k(x^J) + q \operatorname{Ln} \left( \frac{\beta(k - c^T x^0)}{(k - c^T x^J)} + 1 \right) \quad (\text{II.8})$$

Substituindo (II.7) e (II.8) em (II.6) tem-se:

$$\frac{1}{22} J \leq f_{k'}(x^0) - f_{k'}(x^J) \leq f_k(x^0) - f_k(x^J) - q \operatorname{Ln}(\beta) +$$

$$+ q \operatorname{Ln} \left( \frac{\beta(k - c^T x^0)}{(k - c^T x^J)} + 1 \right) \quad (\text{II.9})$$

De II.5.2.4 e iii em II.5.3.1 obtemos que:

$$K - c^T x^0 \leq \left(1 + \frac{1}{\sqrt{q}}\right) (k - c^T x(k))$$

Além disso,

$$k - c^T x^J = (k' - c^T x^J) + (k - k')$$

$$\geq (1 - 1/\sqrt{q}) (k' - c^T x(k')) + (1 - \beta) (k - c^T x^0)$$

$$\geq (1 - 1/\sqrt{q})(1-\beta)(k - c^T x(k))$$

Portanto:

$$\frac{\beta(k - c^T x^0)}{(k - c^T x^J)} \leq \frac{\beta(1 + 1/\sqrt{q})}{(1-\beta)(1 - 1/\sqrt{q})}$$

e daí:

$$\frac{1}{22} J \leq \frac{1}{10} - q \operatorname{Ln}(\beta) + q \operatorname{Ln}\left(\frac{\beta(1 + 1/\sqrt{q})}{(1-\beta)(1 - 1/\sqrt{q})} + 1\right), \text{ ou}$$

$$\frac{1}{22} J \leq \frac{1}{10} + q \operatorname{Ln}\left(\frac{1}{\beta} + \frac{(\sqrt{q} + 1)}{\beta(1-\beta)(\sqrt{q}-1)}\right)$$

$$\leq \frac{1}{10} + q \operatorname{Ln}\left(\frac{1 + \frac{2 + (1-\beta)(\sqrt{q}-1)}{\beta(1-\beta)}}{(\sqrt{q}-1)}\right)$$

$$\leq \frac{1}{10} + q \left( \frac{2 + (1-\beta)(\sqrt{q}-1)}{(\sqrt{q}-1)} \right) + q \left( \frac{\beta(\beta-1)+1}{\beta(1-\beta)} \right)$$

$$= \frac{1}{10} + q \left( \frac{1}{\beta(1-\beta)} - \beta \right) + \frac{2q}{(\sqrt{q}+1)}$$

portanto:

$$J \leq 3 + \frac{44q}{\sqrt{q-1}} + 22q \left( \frac{1}{\beta(1-\beta)} - \beta \right)$$

### II.5.3.4 – Corolário

O número total de iterações do algoritmo início é:

$$\left( 3 + \frac{44q}{\sqrt{q-1}} + 22q \left( \frac{1}{\beta(1-\beta)} - \beta \right) \right) O\left(\frac{1}{1+\beta} L\right)$$

### II.5.3.5 – Observação

É claro que se tomamos  $q = O(m)$  então se  $\beta = O\left(\frac{1}{\sqrt{m-1}}\right)$  o algoritmo requer  $O(\sqrt{m} L)$  iterações.

Se  $\beta = O(1)$  o algoritmo requer  $O(mL)$  iterações.

O primeiro caso corresponde a um fator de redução  $\beta$  pequeno. Neste caso voltamos a uma vizinhança da trajetória central em  $O(1)$  de passos, enquanto a cota superior deve ser atualizada em  $O(\sqrt{m} L)$  vezes.

O segundo caso corresponde a um fator de redução  $\beta$  grande. Neste caso voltamos a uma vizinhança da trajetória central em  $O(mL)$  buscas lineares e a cota superior deve ser atualizada em  $O(L)$  vezes.

## II.6 – OBTENÇÃO DO PONTO INICIAL

Nos dados do algoritmo início tem-se que o ponto inicial está perto da

trajetória central. Esta suposição pode ser aliviada com:

$$\text{II.6.1} \quad f_{k^0}(x^0) - f_{k^0}(x^0(k^0)) \leq O(\sqrt{mL})$$

no primeiro caso, e

$$\text{II.6.2} \quad f_{k^0}(x^0) - f_{k^0}(x^0(k^0)) \leq O(mL)$$

no segundo caso.

Note que pedir isto não afeta o número de iterações do algoritmo (ver II.5.2). Para o segundo caso ( $\beta = O(1)$ ) tal alívio implica que podemos começar quase de qualquer ponto interior; suponha somente que o ponto inicial viável e interior satisfaz:

$$\text{II.6.3} \quad z_i^0 = b_i - a_i x^0 \geq 2^{-L}$$

e a cota superior  $k^0$  satisfaz:

$$\text{II.6.4} \quad \frac{k^0 - c^T x^0(k^0)}{k^0 - c^T x^0} \leq 2^L$$

É fácil ver que neste caso:

$$\Delta f_{k^0} = f_{k^0}(x^0) - f_{k^0}(x^0(k^0)) \leq (m+q) O(L) \text{ (HERTOG [10])}.$$

mais ainda, se  $q = O(m)$  então  $\Delta f_{k^0} \leq O(mL)$ .

## CAPÍTULO III

REOTIMIZAÇÃO COM METODOLOGIA  
PONTOS INTERIORES

## III.1 – INTRODUÇÃO

Este é o capítulo mais importante da tese, nele resolvemos o problema que qualquer algoritmo pontos interiores enfrenta em uma metodologia de planos cortantes, isto é, se o problema relaxado do problema de programação inteira é resolvido com um algoritmo pontos interiores, quando novos cortes sejam agregados à relaxação, o ponto interior que era solução ótima aproximada do problema anterior, já não será um ponto interior na região perturbada.

Suponha que a relaxação do problema é dada por:

$$\text{Minimizar } c^T x$$

$$(P)$$

$$\text{Sujeito a: } Ax \leq b$$

onde  $c \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $m > n$ ,  $\text{RANK}(A)$  máximo.

Suponha que o algoritmo início (Capítulo II) é aplicado para resolver (P) e ao final temos um ponto  $\hat{x} \in S_k^0$  ( $k \in \mathbb{R}$ ) tal que:

$$\delta(\hat{x}, k) \leq 0.1$$

Também suponha que agora perturbamos (P) ao agregar o conjunto de

restrições:

$$\Pi x \leq \pi_0$$

onde  $\Pi \in \mathbb{R}^{p \times n}$ ,  $\pi_0 \in \mathbb{R}^p$ ; o novo problema a resolver é:

$$\text{Minimizar } c^T x$$

( $\bar{P}$ )

$$\text{Sujeito a: } Ax \leq b$$

$$\Pi x \leq \pi_0$$

O problema central que agora atacaremos é a construção de um algoritmo polinomial que inicialize eficientemente o algoritmo início para resolver ( $\bar{P}$ ). O termo eficiente será entendido aqui por um ponto perto da trajetória central associada ao problema ( $\bar{P}$ ) e com um gap comparável ao gap de início. O primeiro tópico a considerar será o problema perturbado que se obtém depois de agregar uma restrição ao problema (P).

## III.2 – AGREGAÇÃO DE UMA RESTRIÇÃO AO PROBLEMA ORIGINAL

### III.2.1 – Trajetória Central Associada ao Problema

#### III.2.1.1 – Introdução

Considere de novo o problema (P):

Minimizar  $c^T x$

(P)

Sujeito a:  $Ax \leq b$

e suponha que depois de uma aplicação do algoritmo início (Capítulo II) obtemos um ponto  $\hat{x} \in S_k^0$  (algum  $k \in \mathbb{R}$ ) tal que:

$$\delta(\hat{x}, k) \leq 0.1 \quad (\text{III.1})$$

Suponha que agora perturbamos o problema (P) depois de agregar uma nova restrição dada por:

$$\pi^T x \leq \pi_0$$

onde  $\pi \in \mathbb{R}^n$ ,  $\pi_0 \in \mathbb{R}$ .

O novo problema que se deseja resolver é:

Minimizar  $c^T x$

(P( $\pi$ ,  $\pi_0$ ))

Sujeito a:  $Ax \leq b$   
 $\pi^T x \leq \pi_0$

Suponha que o ponto  $\hat{x}$  que satisfaz (III.1) é tal que:

$$\pi^T \hat{x} > \pi_0$$

senão uma simples centralização desde  $\hat{x}$  resolve o objetivo proposto (ver [30]).

### III.2.1.2 – Objetivo Central

Para recuperar a  $\hat{x}$  como dado no problema de reotimização, consideremos a seguinte relaxação do problema  $P(\pi, \pi_0)$ :

$$\begin{array}{ll} \text{Minimizar } c^T x & \\ & P(\pi, \gamma) \\ \text{Sujeito a: } & Ax \leq b \\ & \pi^T x \leq \gamma \end{array}$$

onde  $\gamma$  é um número real.

É claro que se  $\gamma \geq \pi^T \hat{x}$ , então  $\hat{x}$  é viável para  $P(\pi, \gamma)$ , mas isso não é suficiente, porque se quer também que o ponto  $\hat{x}$  esteja perto da trajetória central associada a  $P(\pi, \gamma)$  através de uma função tipo centro.

Para isto é necessário dar um certo peso ( $\ell$ ) à restrição relaxada  $\pi^T x \leq \gamma$  na função centro associada ao sistema, então consideramos a restrição com  $\ell$ -cópias em  $P(\pi, \gamma)$ , isto é:

$$\begin{array}{ll} \text{Minimizar } c^T x & \\ & (P_\ell(\pi, \gamma)) \\ \text{Sujeito a: } & Ax \leq b \\ & \left. \begin{array}{l} \pi^T x \leq \gamma \\ \pi^T x \leq \gamma \\ \vdots \\ \pi^T x \leq \gamma \end{array} \right\} \ell\text{-vezes} \end{array}$$

ou simplesmente,

$$\begin{aligned} & \text{Minimizar } c^T x \\ & \text{Sujeito a: } \quad Ax \leq b \\ & \quad \quad \quad \pi^T x \leq \gamma \text{ (\ell-vezes)} \end{aligned}$$

Note que  $P(\pi, \gamma)$  e  $P_\ell(\pi, \gamma)$  representam o mesmo problema.

Seja  $S(k, \gamma) = S_k \cap \{x \in \mathbb{R}^n: \pi^T x \leq \gamma\}$ .  $S^0(k, \gamma)$  denota o interior de  $S(k, \gamma)$  que por hipótese não é vazio. Como no Capítulo II associe a  $P_\ell(\pi, \gamma)$  a função centro:  $x \in S^0(k, \gamma) \rightarrow \varphi_{k, \gamma}(x) = f_k(x) - \ell \ln(\gamma - \pi^T x)$ ; onde  $\ell$ -inteiro positivo,  $\ell \geq m + q$ .

### III.2.1.3 – Cálculos

Como no Capítulo II associe a  $P(\pi, \gamma)$  a família de problemas:

$$\begin{aligned} & \text{Minimizar } \varphi_{k, \gamma}(x) \\ & \hspace{15em} (P(k, \gamma)) \end{aligned}$$

$$\text{Sujeito a: } x \in S^0(k, \gamma)$$

O gradiente e a matriz hessiana da função  $\varphi_{k, \gamma}(\cdot)$  estão dados por:

$$\bar{g} = \bar{g}(x, k, \gamma) = \nabla \varphi_{k, \gamma}(x) := \nabla f_k(x) + \frac{\ell}{\gamma - \pi^T x} \pi = g + \ell \lambda \pi$$

$$\text{onde } \lambda = 1/\gamma - \pi^T x;$$

$$\bar{H} = \bar{H}(x, k, \gamma) = \nabla^2 \varphi_{k, \gamma}(x) := \nabla^2 f_k(x) + \frac{\ell}{(\gamma - \pi^T x)^2} \pi \pi^T = H + \ell \lambda^2 \pi \pi^T$$

Note que  $\bar{H}$  é definida positiva.

Pela fórmula de SHERMAN-MORRISON-WOODBURY [44]

$$\bar{H}^{-1} = (H + \ell \lambda^2 \pi \pi^T)^{-1} := H^{-1} - \frac{\ell \lambda^2 H^{-1} \pi \pi^T H^{-1}}{1 + \ell \lambda^2 \pi^T H^{-1} \pi}$$

Fazendo  $\theta = \pi^T H^{-1} \pi > 0$ , temos que:

$$\bar{H}^{-1} := H^{-1} - \frac{\ell \lambda^2}{1 + \theta \ell \lambda^2} H^{-1} \pi \pi^T H^{-1}$$

### III.2.1.4 – Direção Newton e Proximidade

#### III.2.1.4.1 – Definição

Dados  $k, \gamma \in \mathbb{R}$  (fixo) a direção Newton definida desde  $x \in S^0(k, \gamma)$  está dada por:

$$\bar{h} = \bar{h}(x, k, \gamma) := -\bar{H}^{-1} \bar{g}$$

#### III.2.1.4.2 – Definição

Seja:

$$\bar{x} = \bar{x}(k, \gamma) := \text{ARGMIN}\{\varphi_{k, \gamma}(x) : x \in S^0(k, \gamma)\}$$

Dado  $\mathbf{x} \in S^0(k, \gamma)$ . A proximidade de  $\mathbf{x}$  em relação a  $\bar{\mathbf{x}} = \bar{\mathbf{x}}(k, \gamma)$  é caracterizada por:

$$\bar{\delta} = \bar{\delta}(\mathbf{x}, k, \gamma) := \|\bar{h}(\mathbf{x}, k, \gamma)\|_{\bar{H}}$$

O ponto será considerado **aproximadamente central** se:

$$\bar{\delta}(\mathbf{x}, k, \gamma) \leq \epsilon, \epsilon \in (0, 1)$$

### III.2.1.4.3 – Observação

Para simples cálculos, temos que:

$$\|\mathbf{x}\|_{\bar{H}}^2 := \|\mathbf{x}\|_{\bar{H}}^2 + \ell\lambda^2 \|\mathbf{x}\|_{\pi\pi^T}^2 \text{ para todo } \mathbf{x} \in \mathbb{R}^n$$

$$\|H^{-1}\pi\|_{\bar{H}}^2 = \|\pi\|_{H^{-1}}^2 = \theta$$

$$\|H^{-1}\pi\|_{\pi\pi^T}^2 = \|\pi\|_{H^{-1}\pi\pi^T H^{-1}}^2 = \theta^2$$

### III.2.1.4.4 – Lema

Dado  $k, \gamma \in \mathbb{R}$  e  $\mathbf{x} \in S^0(k, \gamma)$

$$\bar{h}(\mathbf{x}, k, \gamma) = h(\mathbf{x}, k) - \frac{\ell(\alpha + (\gamma - \pi^T \mathbf{x}))}{\theta\ell + (\gamma - \pi^T \mathbf{x})^2} \frac{\pi}{\pi}$$

onde

$$\alpha := \pi^T h$$

$$\theta := \pi^T H^{-1} \pi$$

$$\bar{\pi} := H^{-1} \pi$$

**Prova**

Pela definição de  $\bar{h}(x, k, \gamma)$ :

$$\begin{aligned} \bar{h}(x, k, \gamma) &:= \left( -H^{-1} + \frac{\ell \lambda^2}{1 + \theta \ell \lambda^2} H^{-1} \pi \pi^T H^{-1} \right) (\mathfrak{g} + \ell \lambda \pi) \\ &= h(x, k) - \ell \lambda \bar{\pi} - \frac{\alpha \ell \lambda^2}{1 + \theta \ell \lambda^2} \bar{\pi} + \frac{\theta \ell^2 \lambda^3}{1 + \theta \ell \lambda^2} \bar{\pi} \end{aligned}$$

Portanto:

$$\begin{aligned} \bar{h}(x, k, \gamma) &:= h(x, k) - \left( \ell \lambda + \frac{\alpha \ell \lambda^2}{1 + \theta \ell \lambda^2} - \frac{\theta \ell^2 \lambda^3}{1 + \theta \ell \lambda^2} \right) \bar{\pi} \\ &:= h(x, k) - \frac{\ell \lambda (1 + \alpha \lambda)}{1 + \theta \ell \lambda^2} \bar{\pi} \\ &:= h(x, k) - \frac{\ell (\alpha + (\gamma - \pi^T x))}{\theta \ell + (\gamma - \pi^T x)^2} \bar{\pi} \end{aligned}$$

Pois  $\lambda := 1/\gamma - \pi^T x$

## III.2.1.4.5 – Lema

Dado  $k, \gamma \in \mathbb{R}$ , se  $x \in S^0(k, \gamma)$  então

$$\bar{\delta} = \bar{\delta}(x, k, \gamma) : \leq \delta(1 + \sqrt{\theta}) \ell \lambda^2 + \sqrt{\theta} \ell \lambda + \delta$$

onde:  $\theta = \pi^T H^{-1} \pi > 0$   
 $\lambda = 1/\gamma - \pi^T x$

**Prova**

$$\begin{aligned} \bar{\delta} &:= \|\bar{h}\|_{\bar{H}} = \left\| h - \frac{\ell \lambda}{1 + \theta \ell \lambda^2} (1 + \alpha \lambda) \bar{\pi} \right\|_{\bar{H}} \leq \|\bar{h}\|_{\bar{H}} + \frac{\ell \lambda}{1 + \theta \ell \lambda^2} (1 + |\alpha \lambda|) \|\bar{\pi}\|_{\bar{H}} \\ &\leq \|\bar{h}\|_{\bar{H}} + \frac{\ell \lambda}{1 + \theta \ell \lambda^2} (1 + |\alpha| |\lambda|) \|\bar{\pi}\|_{\bar{H}}. \end{aligned}$$

Mas,

$$\|\bar{h}\|_{\bar{H}} = \sqrt{\frac{\delta^2 + \ell \lambda^2 \|\bar{h}\|_{\pi}^2}{\pi \pi^T}} = \sqrt{\delta^2 + \alpha^2 \ell \lambda^2}; \text{ e}$$

$$\|\bar{\pi}\|_{\bar{H}} = \sqrt{\frac{\theta + \ell \lambda^2 \|\bar{\pi}\|_{\pi}^2}{\pi \pi^T}} = \sqrt{\theta} \sqrt{1 + \theta \ell \lambda^2}$$

Desta forma que:

$$\bar{\delta} \leq \sqrt{\delta^2 + \alpha^2 \ell \lambda^2} + \frac{\ell \lambda}{1 + \theta \ell \lambda^2} (1 + |\alpha| |\lambda|) \sqrt{\theta} \sqrt{1 + \theta \ell \lambda^2}$$

$$\leq \sqrt{\delta^2 + \alpha^2 \ell \lambda^2} + \sqrt{\theta} \ell \lambda (1 + |\alpha| \lambda)$$

s.p.g. Suponha que  $\|\pi\|_H = 1$ ; agora:

$$|\alpha| = |\pi^T h| \leq \|\pi\|_H \|h\|_H = \delta \text{ (Cauchy-Schwarz)}$$

De aí que:

$$\bar{\delta} \leq \sqrt{1 + \ell \lambda^2} + \sqrt{\theta} \ell \lambda (1 + \delta \lambda)$$

$$\leq \delta(1 + \ell \lambda^2) + \sqrt{\theta} \ell \lambda (1 + \delta \lambda)$$

$$= \delta(1 + \sqrt{\theta}) \ell \lambda^2 + \sqrt{\theta} \ell \lambda + \delta$$

Dos dois lemas anteriores segue-se o seguinte resultado:

### III.2.1.4.6 – Teorema

s.p.g. suponha que  $\theta = 1$  e que  $\delta = \delta(\mathbf{x}, \mathbf{k}) \leq 0.1$  com  $\mathbf{x} \in S_k^0$ . Então  $\bar{\delta} = \bar{\delta}(\mathbf{x}, \mathbf{k}, \gamma) \leq 0.2$  para

$$\gamma := \pi^T \mathbf{x} + \frac{2\ell}{\sqrt{25\ell^2 + 2\ell - 5\ell}}$$

Note que  $\gamma$  está bem definido, posto que:

$$\sqrt{25\ell^2 + 2\ell} - 5\ell > 0$$

### III.2.1.4.8 – Observação

O resultado anterior pode ser representado graficamente em  $\mathbb{R}^2$  como segue:

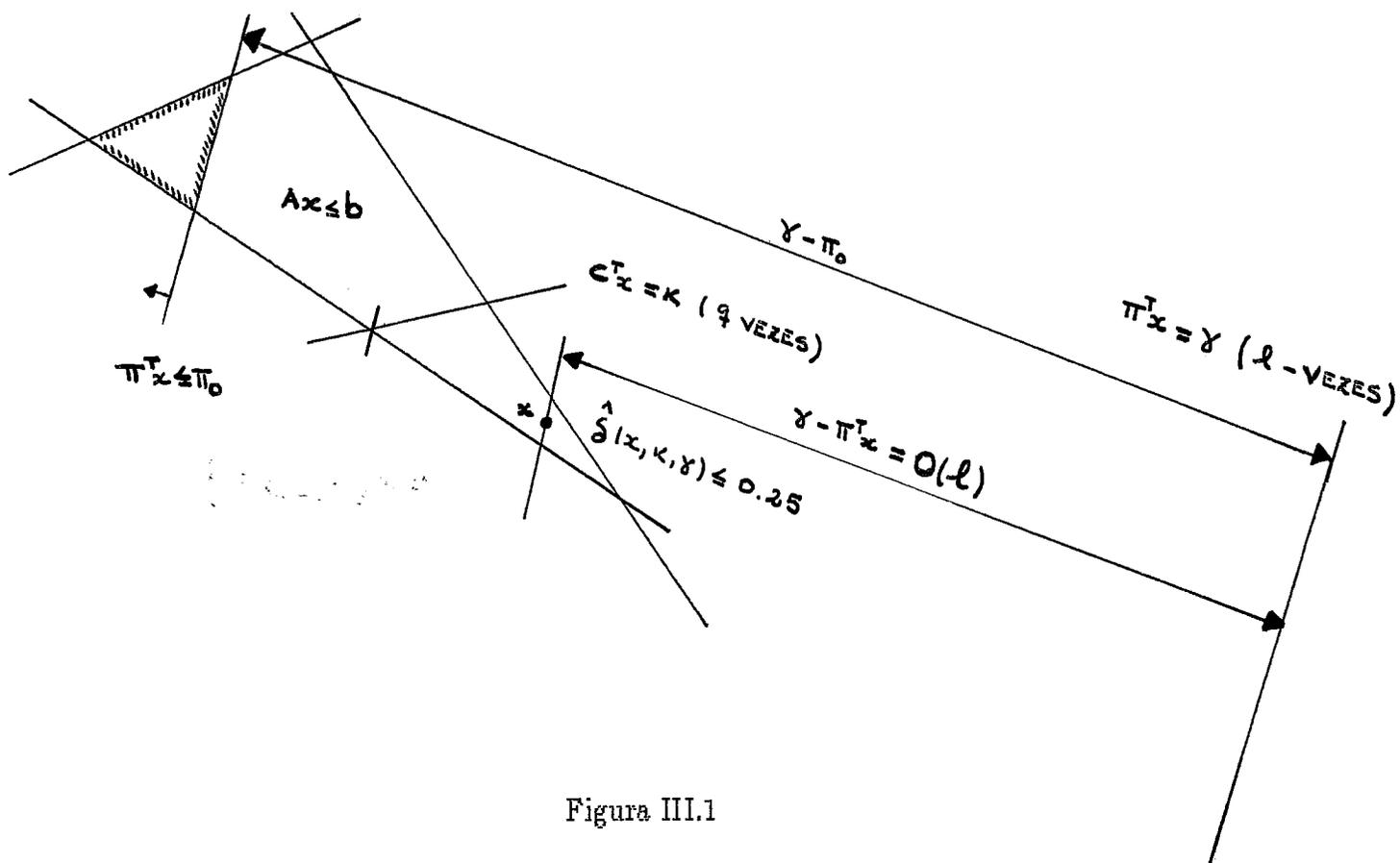


Figura III.1

## III.2.2 – Trajetória Central Associada ao Problema Perturbado com 1-Cópia

### III.2.2.1 – Observação

O teorema III.2.1.4.7 garante que se temos um ponto  $x \in S_k^0$  tal que

$\delta(\mathbf{x}, k) \leq 0.1$ ; então existe  $\gamma := \pi^T \mathbf{x} + \frac{2\ell}{\sqrt{25\ell^2 + 2\ell - 5\ell}}$  tal que  $\bar{\delta}(\mathbf{x}, k, \gamma) \leq 0.2$ .

Desta maneira, é possível rodar um algoritmo de trajetória central a partir deste ponto; o problema é que terminaríamos com um ponto perto da trajetória central associada ao problema com a restrição  $\pi^T \mathbf{x} \leq \gamma$   $\ell$ -vezes, e lembre que nós desejamos resolver o problema com a restrição  $\pi^T \mathbf{x} \leq \pi_0$  satisfeita só uma vez. Nesta seção apresentamos algumas relações existentes (e de muita utilidade) entre a trajetória central associada ao problema perturbado com  $\ell$ -cópias e 1-cópia respectivamente.

### III.2.2.2 – Observação

Considere a família de problemas:

$$\text{Minimizar } \varphi_{k,\gamma}^1(\mathbf{x}) = f_k(\mathbf{x}) - \text{Ln}(\gamma - \pi^T \mathbf{x}) \quad (\text{P}_1(k, \gamma))$$

Sujeito a:  $\mathbf{x} \in S^0(k, \gamma)$

Note que:  $\varphi_{k,\gamma}^1(\mathbf{x}) - \varphi_{k,\gamma}(\mathbf{x}) = (\ell - 1) \text{Ln}(\gamma - \pi^T \mathbf{x})$

É fácil ver que o vetor gradiente e a matriz hessiana associada a  $\varphi_{k,\gamma}^1(\cdot)$  são dados por:

$$\mathbf{g}_1 = \mathbf{g}_1(\mathbf{x}, k, \gamma) := \nabla \varphi_{k,\gamma}^1(\mathbf{x}) = \nabla f_k(\mathbf{x}) + \frac{1}{\gamma - \pi^T \mathbf{x}} \pi = \mathbf{g} + \lambda \pi = \bar{\mathbf{g}} + \lambda(1 - \ell) \pi$$

$$H_1 = H_1(\mathbf{x}, k, \gamma) := \nabla^2 \varphi_{k,\gamma}^1(\mathbf{x}) = \nabla^2 f_k(\mathbf{x}) + \frac{1}{(\gamma - \pi^T \mathbf{x})^2} \pi \pi^T = H + \lambda^2 \pi \pi^T =$$

$$= \bar{H} + \lambda^2(1-\ell) \pi \pi^T$$

### III.2.2.3 – Definição

Sejam  $k, \gamma \in \mathbb{R}$  e  $x \in S^0(k, \gamma)$ .

Seja  $x_1 = x_1(k, \gamma) := \text{ARGMIN}\{\varphi_{k, \gamma}^1(x) : x \in S^0(k, \gamma)\}$  a proximidade de  $x$  em relação a  $x_1(k, \gamma)$  é caracterizada por:

$$\delta_1(x, k, \gamma) := \|h_1(x, k, \gamma)\|_{H_1}$$

onde  $h_1(x, k, \gamma) := -H_1^{-1} g_1$

O ponto será considerado **aproximadamente central** se:

$$\delta_1(x, k, \gamma) \leq \epsilon, \quad \epsilon \in (0, 1)$$

### III.2.2.4 – Proposição

Dado  $k, \gamma \in \mathbb{R}$  e  $x \in S^0(k, \gamma)$ . Então

$$\delta_1(x, k, \bar{\gamma}) = \delta(x, k, \gamma), \text{ onde}$$

$$\bar{\gamma} = \frac{1}{\ell} \gamma + \left(1 - \frac{1}{\ell}\right) \pi^T x$$

**Prova**

Pela definição de  $\bar{\gamma}$  tem-se que:

$$\bar{\gamma} - \pi^T \mathbf{x} = \frac{1}{\ell} (\gamma - \pi^T \mathbf{x}); \text{ ou}$$

$$\frac{1}{\bar{\gamma} - \pi^T \mathbf{x}} = \frac{\ell}{\gamma - \pi^T \mathbf{x}}; \text{ portanto,}$$

$$\mathbf{g}_1 = \nabla \varphi_{k, \bar{\gamma}}^1(\mathbf{x}) = \nabla \varphi_{k, \gamma}(\mathbf{x}) = \bar{\mathbf{g}}$$

Segue-se que:

$$\nabla^2 \varphi_{k, \bar{\gamma}}^1(\mathbf{x}) = \nabla^2 \varphi_{k, \gamma}(\mathbf{x}); \text{ ou, } H_1(\mathbf{x}, k, \bar{\gamma}) = \bar{H}(\mathbf{x}, k, \gamma)$$

finalmente:

$$\delta_1(\mathbf{x}, k, \bar{\gamma}) := -H_1^{-1} \mathbf{g}_1 = -\bar{H}^{-1} \bar{\mathbf{g}} = \delta(\mathbf{x}, k, \gamma).$$

Do resultado anterior segue-se que:

$$\text{Se } \bar{\delta}(\mathbf{x}, k, \gamma) = 0.1 \text{ então } \delta_1(\mathbf{x}, k, \bar{\gamma}) = 0.1 \text{ para } \bar{\gamma} = \frac{1}{\ell} \gamma + \left(1 - \frac{1}{\ell}\right) \pi^T \mathbf{x}.$$

O resultado anterior pode ser representado em  $\mathbb{R}^2$  como segue:

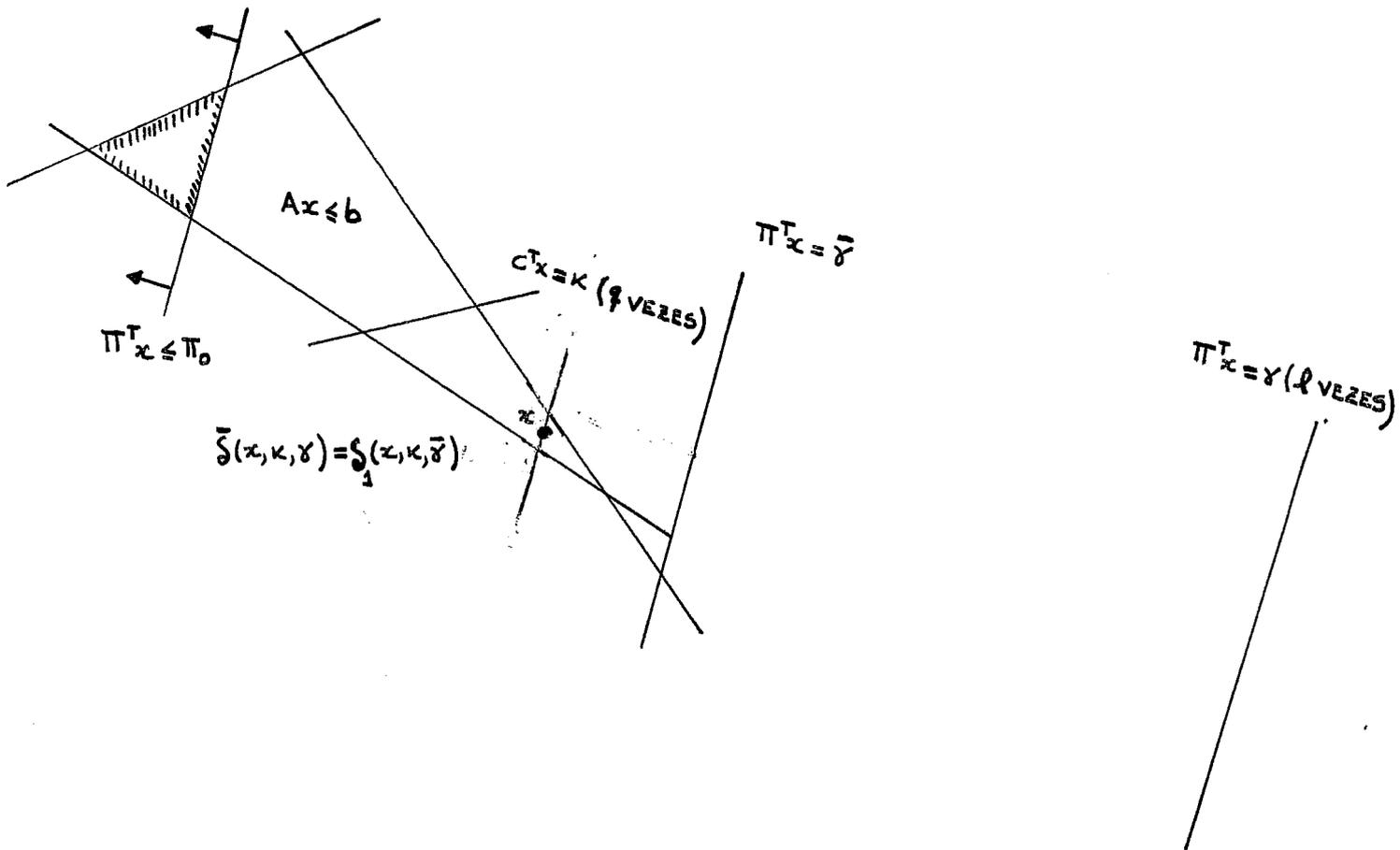


Figura III.2

### III.2.3 – Propriedade Dual de Pontos Próximos a Trajetória Central do Problema

#### III.2.3.1 – Notação

Seja  $\bar{A} = [A^T \underbrace{c \dots c}_{q\text{-vezes}}]$ , e

$$\tilde{Z} := \left[ \begin{array}{cccc} z_1 & & & \\ & \ddots & & \\ & & z_m & \\ & & & z \\ & & & & \ddots \\ & & & & & z \end{array} \right] \left. \vphantom{\begin{array}{c} z_1 \\ \ddots \\ z_m \\ z \\ \ddots \\ z \end{array}} \right\} q\text{-vezes}$$

onde:  $z_i = b_i - a_i x$  ( $i = 1, \dots, m$ ); e  $z = k - c^T x$ ;  $x \in S_k$ .

Note que  $\tilde{A} \in \mathbb{R}^{n \times (m+q)}$  e  $\tilde{Z} \in \mathbb{R}^{(m+q) \times (m+q)}$ .

Seja  $B = \tilde{A} \tilde{Z}^{-1}$ ;  $B \in \mathbb{R}^{n \times (m+q)}$ ; e

$$e = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{m+q}$$

### III.2.3.2 – Proposição

Sejam  $k, \gamma \in \mathbb{R}$  e  $x \in S^0(k, \gamma)$  então

$$H = H(x, k, \gamma) = BB^T \quad (\text{III.2})$$

$$g = g(x, k, \gamma) = Be \quad (\text{III.3})$$

#### Prova

Imediato de III.2.3.1.

### III.2.3.3 – Observação

Note que pela proposição anterior, o passo Newton pode ser determinado por:

$$BB^T h = -Be \quad (\text{III.4})$$

Este vetor será importante a seguir:

$$v = B^T h; v \in \mathbb{R}^{m+q} \quad (\text{III.5})$$

Note que:

$$\|v\|^2 = h^T BB^T h = h^T Hh = \|h\|_H^2 = \delta^2 \quad (\text{III.6})$$

Portanto que se  $\delta < 1$  então:

$$-e < v < e \quad (\text{III.7})$$

### III.2.3.4 – Lema

Considere  $\omega = v + e$  e  $x \in S_k^0$ . Se  $\delta < 1$  então:

$$y_i = \frac{(k - c^T x)}{q} \frac{\omega_i}{z_i \omega_{m+1}} \quad (\text{III.8})$$

( $i = 1, \dots, m$ ) é dual viável.

**Prova**

Pela hipótese  $\delta < 1$  então por (III.7);  $\omega \geq 0$ , e de portanto  $y_i \geq 0$ ,  $i = 1, \dots, m$ .

Agora de (III.4) e (III.5) temos que:

$$B\omega = 0$$

Da definição de B tem-se:

$$\sum_{i=1}^m \frac{\omega_i}{z_i} a_i^T = - \frac{q}{k-c^T x} \omega_{m+1} c;$$

porque todos os  $\omega_{m+1}$  são iguais,  $i = 1, \dots, q$ . Definindo:

$$y_i = \frac{(k-c^T x)}{q} \frac{\omega_i}{z_i \omega_{m+1}}$$

temos que,  $A^T y + c = 0$  e portanto que  $y$  é dual viável.

**III.2.3.5 - Teorema**

Seja  $x \in S_k^0$ . Se  $\delta = \delta(x, k) < 1$  e  $y$  definido por (III.8) então o gap de dualidade satisfaz:

$$\frac{m}{q} (k-c^T x) \frac{(\sigma - \delta \sqrt{\frac{\sigma}{m}})}{(\sigma + \delta \sqrt{\frac{\sigma}{q}})} \leq z^T y \leq \frac{m}{q} (k-c^T x) \frac{(\sigma + \delta \sqrt{\frac{\sigma}{m}})}{(\sigma - \delta \sqrt{\frac{\sigma}{q}})} \quad (\text{III.9})$$

**Prova**

De (III.8) temos que:

$$z_i y_i = \frac{(k-c^T x)}{q} \frac{\omega_i}{\omega_{n+1}}$$

portanto:

$$\text{GAP} = z^T y = \frac{(k-c^T x)}{q} \frac{\sum_{i=1}^m \omega_i}{q \omega_{m+1}} = \frac{(k-c^T x)}{q} \chi(\omega)$$

onde:

$$\chi(\omega) = \sum_{i=1}^m \omega_i / q \omega_{m+1}$$

Note que  $\chi(\lambda \omega) = \lambda \chi(\omega)$  para todo  $\lambda > 0$ . A seguir procuramos uma cota superior para  $\chi(\omega)$ .

**Note que:**

$$\omega^T \omega = \|v\|^2 + 2v^T e + \|e\|^2;$$

$$\begin{aligned} \|h\|_H^2 &= h^T H h = h^T (-B e) \text{ (por (III.4))} \\ &= -(B e)^T h = -e^T B^T h = -e^T v \text{ (por (III.5))} \end{aligned}$$

Portanto,

$$\omega^T \omega = \|e\|^2 - \|h\|_H^2 = m + q - \delta^2$$

Mas,

$$\omega^T e = \|e\|^2 - \|h\|_H^2 = m + q - \delta^2$$

Desta maneira  $\omega$  está sobre a esfera:

$$\omega^T \omega = m + q - \delta^2 \tag{III.10}$$

e o hiperplano

$$\omega^T e = m + q - \delta^2 \tag{III.11}$$

De (III.10) e (III.11) segue-se que  $\omega$  pertence a esfera  $(m+q-1)$ -dimensional com centro  $\sigma e$  e raio  $\delta \sqrt{\sigma}$  onde  $\sigma = 1 - \frac{\delta^2}{m+q}$ , isto é:

$$\sum_{i=1}^m (\omega_i - \sigma)^2 + q(\omega_{m+1} - \sigma)^2 = \sigma \delta^2 \tag{III.12}$$

Para obter uma cota superior de  $\chi(\omega)$ , maximizamos  $\chi(\omega)$  sujeito a (III.12). Este máximo é certamente menor que o máximo de  $\chi(\omega)$  sujeito a:

$$\sum_{i=1}^m (\omega_i - \sigma)^2 = \sigma \delta^2$$

$$q(\omega_{m+1} - \sigma)^2 = \sigma \delta^2$$

Este último é fácil encontrar (condições de Karush-Kuhn-Tucker) e chegar que:

$$\chi(\omega) \leq \frac{m}{q} \frac{(\sigma + \delta \sqrt{\frac{\sigma}{m}})}{(\sigma - \delta \sqrt{\frac{\sigma}{q}})}$$

Da mesma forma pode-se chegar a:

$$\chi(\omega) \geq \frac{m}{q} \frac{(\sigma - \delta \sqrt{\frac{\sigma}{m}})}{(\sigma + \delta \sqrt{\frac{\sigma}{q}})}$$

### III.2.3.6 - Corolário

Seja  $x \in S_k^0$ . Se  $\delta = \delta(x, k) < 1$ .

Então

$$c^T x - v^* \leq \frac{m}{q} (k - c^T x) + \frac{(\sigma + \delta \sqrt{\frac{\sigma}{m}})}{(\sigma - \delta \sqrt{\frac{\sigma}{q}})}$$

onde  $v^*$  é o valor ótimo de (P).

### III.2.3.7 – Corolário

Seja  $x \in S^0(k, \gamma)$ . Se  $\bar{\delta} = \bar{\delta}(x, k, \gamma) < 1$  então  $\pi^T x - u^* \leq \Delta(\bar{\delta}) (\gamma - \pi^T x)$ , onde  $u^*$  é o valor ótimo do problema.

Minimizar  $\pi^T x$

Sujeito a:  $Ax \leq b$

$c^T x \leq k$  (q-vezes)

e

$$\Delta(\bar{\delta}) = \frac{(m+q) \left( \bar{\sigma} + \bar{\delta} \sqrt{\frac{\bar{\sigma}}{m+q}} \right)}{\ell \left( \bar{\sigma} - \bar{\delta} \sqrt{\frac{\bar{\sigma}}{\ell}} \right)}$$

onde  $\bar{\sigma} = 1 - \bar{\delta}^2 / (m+q+\ell)$ .

Note que pela definição de  $\bar{\sigma}$  e  $\Delta(\bar{\delta})$ ,  $\bar{\delta} < 1$ ;  $\Delta(\bar{\delta}) \approx m+q/\ell$ .

No que segue elegemos  $\Delta(\bar{\delta}) = m+q/\ell$ ; portanto que:

$$\pi^T x - u^* \leq \frac{m+q}{\ell} (\gamma - \pi^T x) \quad (\text{III.13})$$

### III.2.4 – Alcance à Região Perturbada e Condição de Parada

#### III.2.4.1 – Observação

Pelos resultados obtidos até agora podemos supor que é conhecido um ponto  $\bar{x} \in S^0(k, \gamma)$ ;  $k \in \mathbb{R}$  tal que  $c^T \bar{x} < k$  e  $\gamma = \pi^T \bar{x} + O(\ell)$  com  $\delta(\bar{x}, k, \gamma) \leq 0.25$ ; ver Figura (III.3).

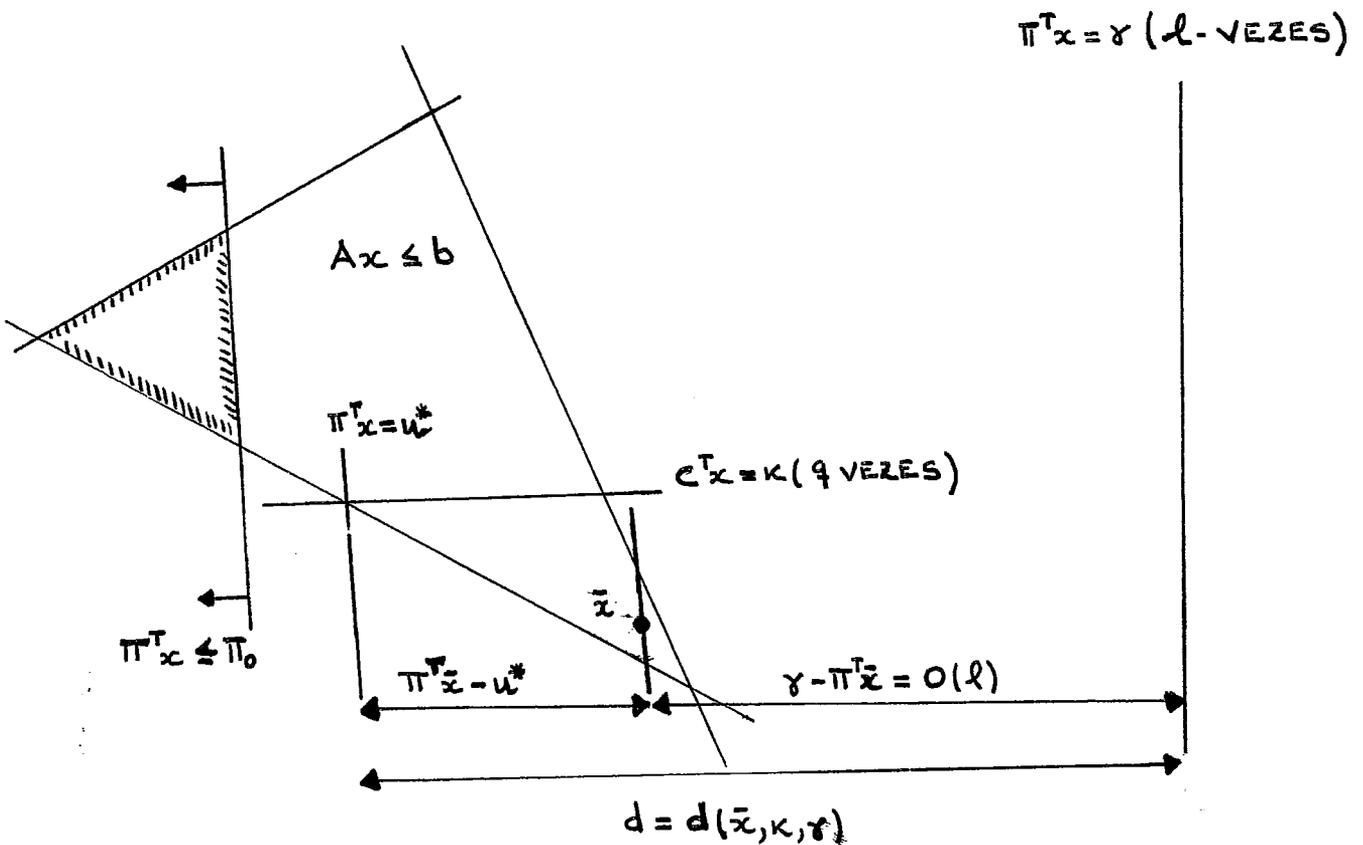


Figura III.3

Desde este ponto podemos rodar o algoritmo início para o problema:

Minimizar  $\pi^T x$

Sujeito a:  $Ax \leq b$   
 $c^T x \leq k$  (q-vezes)

e baixo decréscimos de  $\gamma$  (que é uma cota superior de  $\pi^T x$ ) obter um ponto  $x^*$  ótimo do problema anterior com valor ótimo  $u^* = \pi^T x^*$  (ver Figura III.3).

O problema é que o ponto  $x^*$  não pertence à região perturbada:

$Ax < b$   
 $\pi^T x < \pi_0$

e portanto representa um ponto não-desejado. Neste caso o mais natural a fazer (tomando em conta que ao final deseja-se minimizar  $c^T x$ ) é aumentar o valor do parâmetro  $k$ .

Para decidir quando fazer isto, é necessário estimar o alcance máximo à região perturbada desde  $\bar{x}$  definido como:

$$d = d(\bar{x}, k, \gamma) = (\gamma - \pi^T \bar{x}) + (\pi^T \bar{x} - u^*)$$

(ver Figura III.3). Note que por (III.13)

$$\begin{aligned} d &= (\gamma - \pi^T \bar{x}) + (\pi^T \bar{x} - u^*) \leq (\gamma - \pi^T \bar{x}) + \frac{(m+q)}{\ell} (\gamma - \pi^T \bar{x}) \\ &= \left(1 + \frac{m+q}{\ell}\right) (\gamma - \pi^T \bar{x}) \end{aligned}$$

Portanto que se  $(1 + \frac{m+q}{\ell})(\gamma - \pi^T \bar{x}) < \gamma - \pi_0$  ou equivalente a:

$$\gamma - \pi^T \bar{x} < \frac{\ell}{\ell+m+q} (\gamma - \pi_0) \quad (\text{III.14})$$

Então uma mudança de  $k$  deve ser feita. A mudança de  $k$  por um  $k'$  deve ser tal que  $(k' - c^T \bar{x})$  aumente em relação a  $(k - c^T \bar{x})$ , por exemplo, fazemos:

$$(k' - c^T \bar{x}) = \frac{1}{\beta} (k - c^T \bar{x})$$

onde:

$$\beta \in (0,1); \text{ ou } k' = \left(\frac{\beta-1}{\beta}\right) c^T \bar{x} + \frac{1}{\beta} k.$$

No que segue suponha que temos:

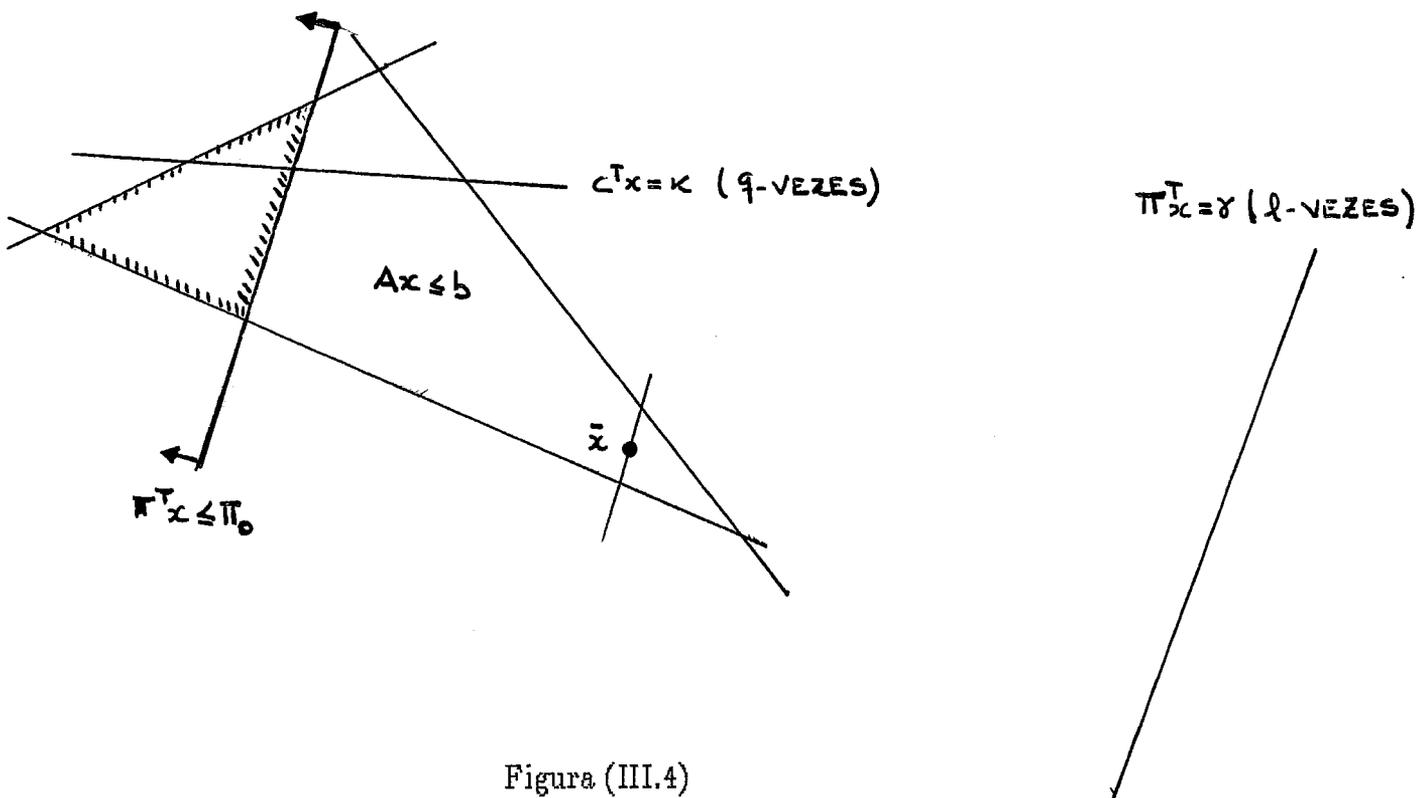


Figura (III.4)

isto é, (III.14) não é satisfeita.

### III.2.4.2 - Observação

Na situação da Figura III.4 podemos iniciar um decréscimo de  $\gamma$ ;  $\gamma' = \frac{1}{\beta} \gamma + (1 - \frac{1}{\beta}) \pi^T x$  e realizar um passo tipo Renegar.

A pergunta natural que agora surge é: Quando parar? Para  $x \in S^0(k, \gamma)$  existe  $\bar{\gamma}$  dado por

$$\bar{\gamma} = \frac{1}{\ell} \gamma + (1 - \frac{1}{\ell}) \pi^T x$$

Tal que:  $\delta_1(x, k, \gamma) = \delta(x, k, \gamma)$ ; então uma vez que um passo tipo Renegar é tomado, calculamos  $\bar{\gamma}$  e determinamos se este é menor que  $\pi_0$ . Se for paramos, senão repetimos o passo Renegar. Note que se na iteração  $j$ ,  $\bar{\gamma}^j \leq \pi_0$ , então  $\bar{\gamma}^{j-1} > \pi_0$  e um algoritmo tipo biseção pode ser aplicado sobre o intervalo que definem  $\bar{\gamma}^{j-1}$  e  $\bar{\gamma}^j$  respectivamente para encontrar um  $\bar{\gamma}$  que seja quase  $\pi_0$ ; ver Figura III.5.

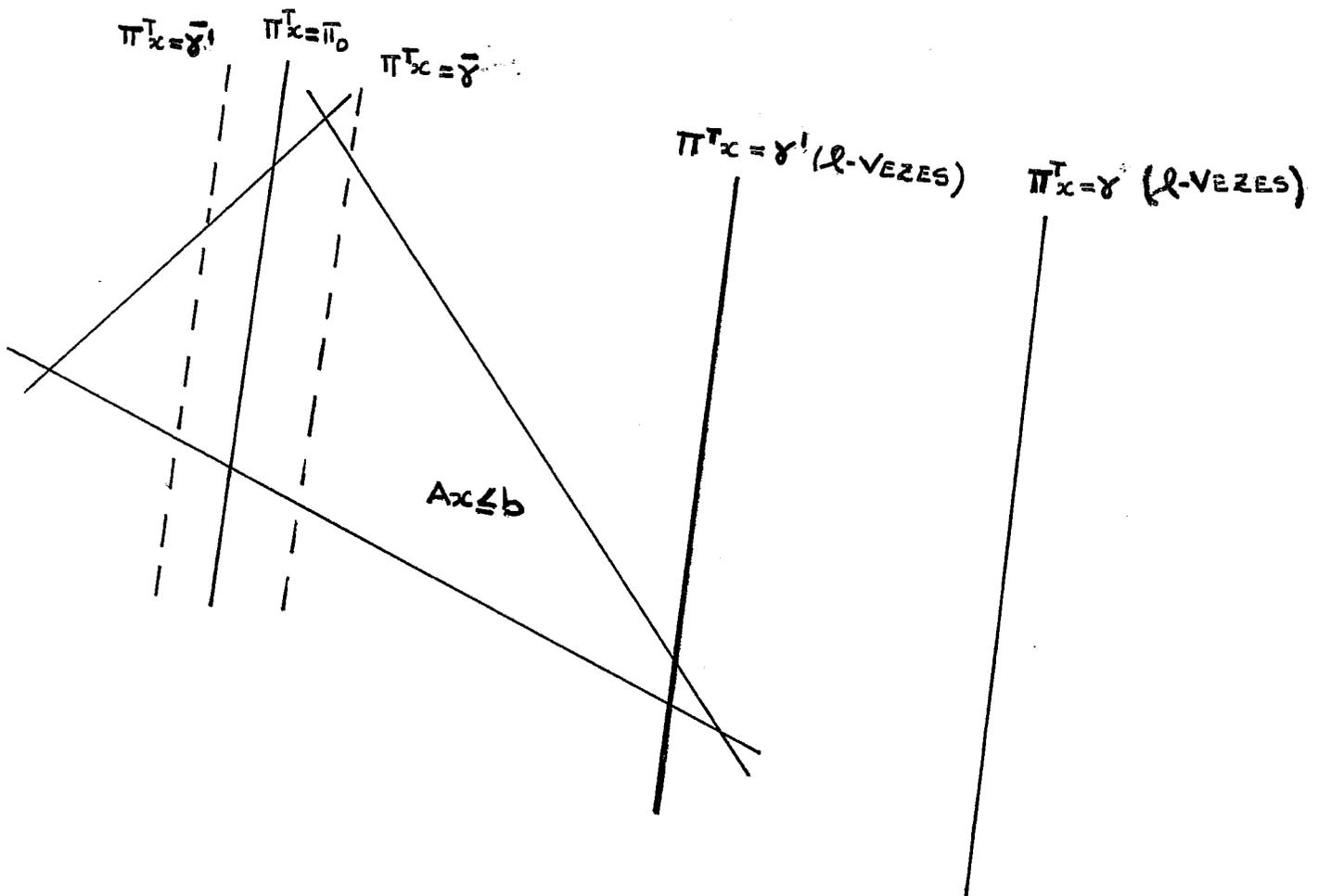


Figura III.5

Só faltaria achar a condição sobre a redução do intervalo que garante que o ponto final obtido está perto da trajetória central associada a  $(\bar{P})$ . Buscamos resolver isto considerando o seguinte problema:

"Dado  $x \in S^0(k, \gamma)$  e tal que  $\bar{\delta}(x, k, \gamma) \leq 0.1$  encontrar uma condição sobre  $\epsilon \in \mathbb{R}$  para que  $\bar{\delta}(x, k, \gamma + \epsilon) \leq 0.2$ ".

Suponha que  $\epsilon > 0$ , então  $\epsilon + \gamma - \pi^T x > 0$  e  $\varphi_{k, \gamma + \epsilon}(\cdot)$  está bem definida.

## III.2.4.3 - Cálculos

$$\Delta \varphi_{k, \gamma + \epsilon}(x) = \nabla f_k(x) + \frac{\ell}{\gamma + \epsilon - \pi^T x} \pi = \bar{g} + \ell \lambda_\epsilon \pi$$

onde:

$$\lambda_\epsilon = 1 / \gamma + \epsilon - \pi^T x$$

$$\nabla^2 \varphi_{k, \gamma + \epsilon}(x) := \nabla^2 f_k(x) + \frac{\ell}{(\gamma + \epsilon - \pi^T x)^2} \pi \pi^T = \bar{H} + \ell \lambda_\epsilon^2 \pi \pi^T$$

Note que:

$$\nabla \varphi_{k, \gamma + \epsilon}(x) := \bar{g} + \ell(\lambda_\epsilon - \lambda) \pi \text{ onde } \lambda = 1 / \gamma - \pi^T x$$

$$\nabla^2 \varphi_{k, \gamma + \epsilon}(x) := \bar{H} + \ell \left( \frac{1}{\lambda_\epsilon^2} - \frac{1}{\lambda^2} \right) \pi \pi^T \text{ definida positiva.}$$

Posto que:

$$\frac{1}{\lambda_\epsilon^2} - \frac{1}{\lambda^2} > 0$$

Seja:

$$\rho = 1 / \lambda_\epsilon^2 - 1 / \lambda^2 \text{ então:}$$

$$\nabla^2 \varphi_{k, \gamma + \epsilon}(x) := \bar{H} + \ell \rho \pi \pi^T$$

$$[\nabla^2 \varphi_{k, \gamma + \epsilon}(x)]^{-1} := \bar{H}^{-1} - \frac{\ell \rho}{1 + \bar{\theta} \ell \rho} \bar{H}^{-1} \pi \pi^T \bar{H}^{-1}$$

onde:

$$\bar{\theta} = \pi^T \bar{H}^{-1} \pi > 0.$$

#### III.2.4.4 - Lema

$$\bar{h}(x, k, \gamma + \epsilon) = \bar{h}(x, k, \gamma) + \frac{\ell(\hat{\alpha} \rho + (\lambda \epsilon - \lambda))}{1 + \bar{\theta} \ell \rho} \hat{\pi}$$

onde:

$$\hat{\alpha} = \pi^T \bar{h}$$

$$\hat{\pi} = \bar{H}^{-1} \pi$$

**Prova**

Imediata.

#### III.2.4.5 - Lema

$$\bar{\delta}(x, k, \gamma + \epsilon) \leq 2\ell \bar{\delta}[(\epsilon + \Delta)^2 - \Delta^2] + \frac{\ell}{\Delta(\epsilon + \Delta)} \epsilon + \bar{\delta}$$

onde:

$$\Delta = \gamma - \pi^T x > 0 \text{ e } \bar{\delta} = \bar{\delta}(x, k, \gamma)$$

Prova

$$\begin{aligned} \bar{\delta}(x, k, \gamma + \epsilon) &= \|\bar{h}(x, k, \gamma + \epsilon)\|_{\bar{H}(x, k, \gamma + \epsilon)} = \|\bar{h}(x, k, \gamma + \epsilon)\|_{\bar{H} + \ell\rho\pi\pi^T} \\ &= \|\bar{h}(x, k, \gamma) + \frac{\ell(\hat{\alpha}\rho + (\lambda_\epsilon - \lambda))}{1 + \bar{\theta}\ell\rho} \hat{\pi}\|_{\bar{H}(x, k, \gamma + \epsilon)} \\ &\leq \|\bar{h}(x, k, \theta + \theta)\|_{\bar{H}(x, k, \gamma + \epsilon)} + \frac{|\ell(\hat{\alpha}\rho + (\lambda_\epsilon - \lambda))|}{1 + \bar{\theta}\ell\rho} \|\hat{\pi}\|_{\bar{H}(x, k, \gamma + \epsilon)} \end{aligned}$$

Mas

$$\|\bar{h}(x, k, \theta + \theta)\|_{\bar{H}(x, k, \gamma + \epsilon)} = \sqrt{\bar{\delta}^2 + \ell\rho\hat{\alpha}^2}$$

e

$$\|\hat{\pi}\|_{\bar{H}(x, k, \gamma + \epsilon)} = \sqrt{\bar{\theta}} \sqrt{1 + \ell\rho\bar{\theta}}$$

Portanto que:

$$\bar{\delta}(x, k, \gamma + \epsilon) \leq \sqrt{\bar{\delta}^2 + \ell\rho\hat{\alpha}^2} + \frac{|\ell(\hat{\alpha}\rho + (\lambda_\epsilon - \lambda))|}{1 + \bar{\theta}\ell\rho} \sqrt{\bar{\theta}} \sqrt{1 + \ell\rho\bar{\theta}}$$

$$\leq \sqrt{\bar{\delta}^2 + \ell\rho\hat{\alpha}^2} + \ell|\hat{\alpha}\rho + (\lambda_\epsilon - \lambda)| \sqrt{\bar{\theta}}$$

$$\leq \sqrt{\bar{\delta}^2 + \ell\rho\hat{\alpha}^2} + \ell(\rho|\hat{\alpha}| + |\lambda_\epsilon - \lambda|) \sqrt{\bar{\theta}}$$

s.p.g. Suponha que  $\|\pi\|_{\bar{H}} = 1$ ; então:

$$|\hat{\alpha}| = |\pi^T \bar{h}| \leq \|\pi\|_{\bar{H}} \|\bar{h}\|_{\bar{H}} = \bar{\delta} \text{ (Cauchy-Schwarz)}$$

Portanto que:

$$\begin{aligned} \bar{\delta}(x, k, \gamma + \epsilon) &\leq \sqrt{\bar{\delta}^2 + \ell \rho \bar{\delta}^2} + \ell \rho \bar{\delta} \sqrt{\bar{\theta}} + \ell |\lambda_{\epsilon} - \lambda| \sqrt{\bar{\theta}} \\ &\leq \ell \bar{\delta} (1 + \sqrt{\bar{\theta}}) [(\epsilon + \Delta)^2 - \Delta^2] + \frac{\ell \sqrt{\bar{\theta}}}{\Delta(\epsilon + \Delta)} \epsilon + \bar{\delta} \end{aligned}$$

s.p.g. Suponha  $\bar{\theta} = 1$  então:

$$\bar{\delta}(x, k, \gamma + \epsilon) \leq 2\ell \bar{\delta} [(\epsilon + \Delta)^2 - \Delta^2] + \frac{\ell}{\Delta(\epsilon + \Delta)} \epsilon + \bar{\delta}$$

onde:

$$\Delta = \gamma - \pi^T x > 0 \text{ e } \bar{\delta} = \bar{\delta}(x, k, \gamma)$$

### III.2.4.6 - Lema

Seja  $x \in S^0(k, \gamma)$ ,  $\epsilon > 0$ ,  $\bar{\delta}(x, k, \gamma) \leq 0.1$ .

Se  $\Delta = \gamma - \pi^T x > 20\ell$  então  $\bar{\delta}(x, k, \gamma + \epsilon) \leq 0.2$ .

Para  $\epsilon \leq -\Delta + \sqrt{\Delta^2 + \frac{1}{4\ell}}$ .

Señão  $\bar{\delta}(x, k, \gamma + \epsilon) \leq 0.2$ .

Para  $\epsilon \leq \min\{-\Delta + \sqrt{\Delta^2 + \frac{1}{4\ell}}, \frac{\Delta^2}{20\ell - \Delta}\}$ .

### Prova

Por III.2.4.5 e a hipótese temos que:

$$\bar{\delta}(x, k, \gamma + \epsilon) \leq \frac{\ell}{5} ((\epsilon + \Delta)^2 - \Delta^2) + \frac{\ell}{\Delta(\epsilon + \Delta)} \epsilon + \frac{1}{10} \quad (\text{III.15})$$

Note que se  $\Delta \geq 20\ell$  então  $\frac{\Delta}{20\ell} \geq 1$  e  $\epsilon \leq \frac{\Delta}{20\ell} \epsilon + \frac{\Delta^2}{20\ell} = \frac{\Delta}{20\ell} (\epsilon + \Delta)$ ;

portanto:

$$\frac{\ell}{\Delta} \frac{\epsilon}{(\epsilon + \Delta)} \leq 0.05 \quad (\text{III.16})$$

Também como  $\epsilon \leq -\Delta + \sqrt{\Delta^2 + \frac{1}{4\ell}}$  então  $(\epsilon + \Delta)^2 \leq \Delta^2 + \frac{1}{4\ell}$ ; ou

$$\frac{\ell}{5} ((\epsilon + \Delta)^2 - \Delta^2) \leq \frac{1}{20} = 0.05 \quad (\text{III.17})$$

Por (III.15), (III.16) e (III.17),  $\bar{\delta}(x, k, \gamma + \epsilon) \leq 0.2$ .

No caso em que  $20\ell - \Delta > 0$  segue-se imediatamente pela escolha de  $\epsilon$ ; i.e., (III.16) e (III.17) são satisfeitas.

### III.3 – O ALGORITMO REINICIAÇÃO

#### III.3.1 – Características Gerais

Dado  $k, \gamma \in \mathbb{R}$  definimos  $P(k, \gamma)$  como o problema de:

Minimizar  $\varphi_{k, \gamma}(x)$

Sujeito a:  $x \in S^0(k, \gamma)$

O algoritmo é composto de duas partes. Na primeira parte o algoritmo começa com um ponto  $x^0 \in S^0(k^0, \gamma^0)$  ( $k^0, \gamma^0$  dados) tal que  $\bar{\delta}(x^0, k^0, \gamma^0) \leq 0.25$ ; i.e., um ponto perto da trajetória central associada a  $P(k^0, \gamma^0)$ . Também se faz  $s = 0$ ;  $j = 0$ ;  $i = 0$ ; e escolhemos  $\beta \in (0, 1)$ . A primeira coisa a fazer é calcular o alcance máximo à região perturbada desde este ponto (ver III.2.4); e enquanto a condição (III.13) em III.2.3.7. Se for satisfeita, uma mudança de  $k$  deve ser feita; i.e., um aumento do valor de  $k$  é providenciado fazendo:

$$k^{i+1} := \frac{(\beta-1)}{\beta} c^T x^s + \frac{1}{\beta} k^i; \quad i = 1, 2, \dots$$

Seguidamente resolvemos  $P(k^i, \gamma^0)$  e guardamos sua solução em  $x^{s+1}$ ; isto é denotado por  $x^{s+1} := \text{SOL } P(k^i, \gamma^0)$ ;  $s = s + 1$ . Isto é o que representa o passo **mudança de  $k$**  na I—parte do algoritmo.

Uma vez que este processo é esgotado, i.e., estamos em uma situação como a mostrada na Figura III.4 (ver III.2.4.1), um passo tipo Renegar é realizado com uma redução do parâmetro  $\gamma$ , i.e., fazemos:

$$\gamma^{i+1} := \frac{1}{\beta} \gamma^i + \left(1 - \frac{1}{\beta}\right) \pi^T x^s$$

$$x^{s+1} := \text{SOL P}(k^i, \gamma^{i+1})$$

$$j = j + 1; \quad s = s + 1.$$

Seguidamente calculamos o parâmetro  $\bar{\gamma}^j$  associado a  $\gamma^j$  e  $x^s$  (ver III.2.2.4) e repetimos isto até que  $\bar{\gamma}^j \leq \pi_0$ .

Na segunda parte do algoritmo tem-se um  $\bar{\gamma}^j \leq \pi_0$  e  $\bar{\gamma}^{j-1} > \pi_0$ ; então um algoritmo tipo biseção é aplicado sobre o intervalo que definem  $\bar{\gamma}^{j-1}$  e  $\bar{\gamma}^j$  respectivamente; em cada iteração deve ser resolvido  $\text{P}(k^i, \gamma)$  onde  $\gamma$  é o ponto médio do intervalo em questão. Agora o processo é repetido até obter um ponto perto da trajetória central associada ao problema; isto é comprovado com o uso do Lema III.2.4.6.

### III.3.2 – Dados

$\beta \in (0,1)$  fator de aumento de  $k$  e redução de  $\gamma$ .  $\epsilon = 0.1$  tolerância da proximidade.

$k^0 \in \mathbb{R}$  cota superior do custo ótimo  $v^*$  tal que:

$$k^0 - v^* < 2^L$$

$x^0 \in S_{k^0}^0$  tal que  $\delta = \delta(x^0, k^0) \leq \epsilon$ .

$\pi \in \mathbb{R}^n$ ,  $\pi_0 \in \mathbb{R}$  tal que  $D := \{x \in \mathbb{R}^n; \pi^T x \leq \pi_0\}$  satisfaz:

$$(S \cap D)^0 \neq \emptyset$$

$q$  inteiro positivo;  $q \geq m$ ;  $\ell$  inteiro positivo,  $\ell \geq m + q$ .

$$\gamma^0 := \pi^T x^0 + \frac{2\ell}{\sqrt{25\ell^2 + 2\ell - 5\ell}}$$

### III.3.3 – Procedimento Geral

$i := 0$ ;

$j := 0$ ;

$s := 0$ ;

#### I Parte

Repita

Enquanto  $\gamma^j - \pi^T x^s < \frac{\ell}{\ell + m + q} (\gamma^j - \pi_0)$  fazer

#### Mudança de $k$

$i := i + 1$ ;

$$k^i := \frac{(\beta - 1)}{\beta} c^T x^s + \frac{1}{\beta} k^{i-1};$$

$u = 0$ ;  $\xi^0 := x^s$

Repita

**Direção** calcular  $\bar{h}(\xi^u, k^i, \gamma^j)$  e  $\bar{\delta} = \bar{\delta}(\xi^u, k^i, \gamma^j)$

**Busca** resolver aproximadamente

$$\bar{\lambda} = \text{ARGMIN}_{\lambda > 0} \{ \varphi_{k^i, \gamma^j}(\xi^u + \lambda \bar{h}) : \xi^u + \lambda \bar{h} \in S^0(k^i, \gamma^j) \};$$

**Novo ponto**  $\xi^{u+1} := \xi^u + \bar{\lambda} \bar{h}$ ;

$u := u + 1$ ;

Até que  $\bar{\delta} \leq 0.1$

$$x^s := \xi^u;$$

$$s = s + 1;$$

**Passo Renegar**

$$\gamma^{j+1} := \frac{1}{\beta} \gamma^j + \left(1 - \frac{1}{\beta}\right) \pi^T x^s$$

$$u := 0; \xi^0 := x^s;$$

Repita

**Direção** calcular  $\bar{h}(\xi^u, k^i, \gamma^{j+1})$  e  $\bar{\delta}(\xi^u, k^i, \gamma^{j+1})$ ;

**Busca** resolver aproximadamente

$$\bar{\lambda} = \underset{\lambda > 0}{\text{ARGMIN}} \{ \varphi_{k^i, \gamma^{j+1}}(\xi^u + \lambda \bar{h}) : \xi^u + \lambda \bar{h} \in S^0(k^i, \gamma^{j+1}) \};$$

**Novo ponto**  $\xi^{u+1} := \xi^u + \bar{\lambda} \bar{h}$

$$u = u + 1$$

Até que  $\bar{\delta} \leq 0.1$

$$x^{s+1} := \xi^u ; s := s + 1 ; j = j + 1 ;$$

$$\text{Calcular } \bar{\gamma}^j := \frac{1}{\ell} \gamma^j + \left(1 - \frac{1}{\ell}\right) \pi^T x^s;$$

Até que  $\bar{\gamma}^j \leq \pi_0$

**II Parte**

**Fazer**  $a := \gamma^{j-1};$

$$b := \gamma^j;$$

$$x^* := x^s;$$

Repita

**Fazer**  $\gamma := (a+b)/2;$

$$u := 0; \xi^0 := x^s$$

Repita

**Direção** calcular  $\bar{h}(\xi^u, k^i, \gamma)$  e  $\bar{\delta}(\xi^u, k^i, \gamma)$ ;

**Busca** resolver aproximadamente

$$\bar{\lambda} = \underset{\lambda > 0}{\text{ARGMIN}} \{ \varphi_{k^i, \gamma}(\xi^u + \lambda \bar{h}) : \xi^u + \lambda \bar{h} \in S^0(k^i, \gamma) \};$$

Novo ponto  $\xi^{u+1} := \xi^u + \bar{\lambda} \bar{h}$ ;

$u = u+1$

Até que  $\bar{\delta} \leq 0.1$

$x^{s+1} := \xi^u$ ;  $s := s + 1$ ;

Calcular  $\bar{\gamma} = (1 - \frac{1}{\ell}) \pi^T x^s + \frac{1}{\ell} \gamma$ ;

Se  $\bar{\gamma} \leq \pi_0$  então fazer  $b := \gamma$ ;  $x^* := x^s$ ;

Senão fazer  $a := \gamma$ ;

Calcular  $\Delta := \gamma - \pi^T x^s$

Se  $\Delta \geq 20\ell$  então fazer  $\rho := -\Delta + \sqrt{\Delta^2 + \frac{1}{4\ell}}$

Senão fazer  $\rho := \min\{-\Delta + \sqrt{\Delta^2 + \frac{1}{4\ell}}, \frac{\Delta^2}{20\ell - \Delta}\}$

Até que  $a - b \leq \rho$

### III.3.3.1 – Observação (Saída do Algoritmo)

Seja  $\bar{P}(k, \gamma)$  o problema:

Minimizar  $\varphi_{k, \gamma}^1(x)$

Sujeito a:  $Ax < b$

$c^T x < k$  (q-vezes)

$\pi^T x < \gamma$

Note que ao final do procedimento geral temos um ponto  $x^*$  perto da trajetória central associada a  $\bar{P}(k^i, \bar{\gamma}^j)$  (algum  $i$  e  $j$ ) e  $\bar{\gamma}^j$  é quase  $\pi_0$ ; i.e., está muito perto de  $\pi_0$ ; desta maneira que o processo de reotimização (ver III.1) pode ser agora completado ao fazer  $\bar{\gamma}^j = \pi_0$  e rodar o algoritmo início desde  $x^*$  e

resolver ( $\bar{P}$ ). É importante notar também que  $x^*$  é um novo ponto interior em  $S^0(k^i, \pi_0)$  com gap comparável ao gap de início. Isto é o que entendemos por uma boa forma de inicializar um processo de reotimização ponto interiormente.

### III.4 – COMPLEXIDADE DO ALGORITMO

#### III.4.1 – Lema

Seja  $J$  o número de passos internos em uma iteração do **passo mudança de  $k$**  (I-Parte). Então:

$$J \leq 3 + 22 \operatorname{Ln} \left( \frac{\beta + \sqrt{q}}{\beta(\sqrt{q}-1)} \right)$$

#### Prova

Denotemos a cota superior de uma iteração externa do passo mudança de  $k$  por  $k'$ , e a cota superior da iteração anterior por  $k$ .

As iterações durante esta iteração externa são denotadas por  $x^0, x^1, \dots, x^J$ ; onde  $x^0$  é o iterado ao começo da iteração exterior.

De II.6 em II.5.3.3 e pela definição de  $\varphi_{k, \gamma}(\cdot)$ :

$$\begin{aligned} \varphi_{k', \gamma}(x^J) &:= f_{k'}(x^J) - \ell \operatorname{Ln}(\gamma - \pi^T x^J) \leq \\ &\leq f_k(x^0) - \frac{1}{22} J - \ell \operatorname{Ln}(\gamma - \pi^T x^0) = \varphi_{k, \gamma}(x^0) - \frac{1}{22} J \end{aligned}$$

$$\varphi_{k',\gamma}(x^J) \leq \varphi_{k',\gamma}(x^0) - \frac{1}{22} J \quad (\text{III.18})$$

Como a cota superior é atualizada ao começo da iteração externa, temos que:

$$\begin{aligned} \varphi_{k',\gamma}(x^0) - \varphi_{k,\gamma}(x^0) &= f_{k'}(x^0) - f_k(x^0) = -q \operatorname{Ln}\left(\frac{k'-c^T x^0}{k-c^T x^0}\right) \\ &= -q \operatorname{Ln}\left(\frac{1}{\beta}\right) = q \operatorname{Ln}(\beta) \end{aligned} \quad (\text{III.19})$$

Mas ainda:

$$\varphi_{k',\gamma}(x^J) - \varphi_{k,\gamma}(x^J) = -q \operatorname{Ln}\left(\frac{k'-c^T x^J}{k-c^T x^J}\right)$$

Agora por definição de  $k'$ :

$$\begin{aligned} k - c^T x^J &= \beta(k'-c^T x^0) + (c^T x^0 - c^T x^J) \\ &\geq \beta(k'-c^T x^0) \end{aligned}$$

Portanto que

$$\begin{aligned} \frac{(k'-c^T x^J)}{(k-c^T x^J)} &\leq \frac{(k'-c^T x^J)}{\beta(k'-c^T x^0)} \\ -q \operatorname{Ln}\left(\frac{k'-c^T x^J}{(k'-c^T x^J)}\right) &\geq -q \operatorname{Ln}\left(\frac{k'-c^T x^J}{(k'-c^T x^0)}\right) \end{aligned}$$

Portanto

$$-\varphi_{k',\gamma}(x^J) \leq -\varphi_{k,\gamma}(x^J) + q \operatorname{Ln}\left(\frac{k'-c^T x^J}{\beta(k'-c^T x^0)}\right) \quad (\text{III.20})$$

Substituindo (III.19) e (III.20) em (III.18), temos que:

$$\frac{1}{22} J \leq \varphi_{k,\gamma}(x^0) - q \operatorname{Ln}\left(\frac{1}{\beta}\right) - \varphi_{k,\gamma}(x^J) + q \operatorname{Ln}\left(\frac{k'-c^T x^J}{\beta(k'-c^T x^0)}\right).$$

Como  $x^0$  é aproximadamente central, então:

$$\varphi_{k,\gamma}(x^0) - \varphi_{k,\gamma}(x(k,\gamma)) < 0.1$$

Portanto que:

$$\varphi_{k,\gamma}(x^0) - \varphi_{k,\gamma}(x^J) < \frac{1}{10} + \varphi_{k,\gamma}(x(k,\gamma)) - \varphi_{k,\gamma}(x^J) < \frac{1}{10}$$

E agora temos que:

$$\frac{1}{22} J \leq \frac{1}{10} - q \operatorname{Ln}(1/\beta) + q \operatorname{Ln}\left(\frac{k'-c^T x^J}{\beta(k'-c^T x^0)}\right) \quad (\text{III.21})$$

Note que por II.5.2.4 tem-se:

$$k'-c^T x^0 > k'-c^T x^0 > \left(1 - \frac{1}{\sqrt{q}}\right) (k'-c^T x(k))$$

E portanto:

$$\frac{1}{\beta(k'-c^T x^0)} \leq \frac{1}{\beta(1 - \frac{1}{\sqrt{q}})(k-c^T x(k))} \quad (\text{III.22})$$

Mas ainda,

$$\begin{aligned} (k'-c^T x^J) &= (k-c^T x^J) + (k'-k) = (k-c^T x^J) + \frac{(1-\beta)}{\beta} (k-c^T x^0) \\ &\leq (k'-c^T x^J) + \frac{(1-\beta)}{\beta} (k-c^T x(k)) \\ &\leq (1 + \frac{1}{\sqrt{q}}) (k'-c^T x(k')) + \frac{(1-\beta)}{\beta} (k-c^T x(k)) \\ &\leq (1 + \frac{1}{\sqrt{q}}) (k-c^T x(k)) + \frac{(1-\beta)}{\beta} (k-c^T x(k)) \\ &= (\frac{(1-\beta)}{\beta} + (1 + \frac{1}{\sqrt{q}})) (k-c^T x(k)) \end{aligned} \quad (\text{III.23})$$

De (III.22) e (III.23) tem-se:

$$\frac{k'-c^T x^J}{\beta(k'-c^T x^0)} \leq \frac{(\frac{(1-\beta)}{\beta} + (1 + \frac{1}{\sqrt{q}}))}{\beta(1 - \frac{1}{\sqrt{q}})}$$

Portanto que (III.21) é:

$$\frac{1}{22} J \leq \frac{1}{10} - q \operatorname{Ln}(1/\beta) + q \operatorname{Ln}\left(\frac{\left(\frac{1-\beta}{\beta}\right) + \left(1 + \frac{1}{\sqrt{q}}\right)}{\beta\left(1 - \frac{1}{\sqrt{q}}\right)}\right)$$

$$\frac{1}{22} J \leq \frac{1}{10} + \operatorname{Ln}\left(\frac{\beta + \sqrt{q}}{\beta(\sqrt{q}-1)}\right)$$

ou

$$J \leq 3 + 22 \operatorname{Ln}\left(\frac{\beta + \sqrt{q}}{\beta(\sqrt{q}-1)}\right)$$

### III.4.2 – Proposição

O número de iterações tipo mudança de  $k$  é estritamente menor que:

$$O\left(\frac{1}{1+\beta} L\right)$$

Note que no pior caso a condição (III.14) em III.2.4.1 é válida até que o algoritmo início retorne a seu  $k$  inicial.

### III.4.3 – Proposição

O número total de iterações tipo mudança de  $k$  é estritamente menor que:

$$\frac{1}{1+\beta} \left(3 + 22 \operatorname{Ln}\left(\frac{\beta + \sqrt{q}}{\beta(\sqrt{q}-1)}\right)\right) O(L)$$

### III.4.4 – Observação

Note que o número total de iterações externas desta primeira parte do algoritmo é estritamente menor que o número total de iterações externas do algoritmo início aplicado ao problema:

$$\begin{aligned} &\text{Minimizar } \pi^T x \\ &\text{Sujeito a: } \quad Ax \leq b \\ &\quad \quad \quad c^T x \leq k \text{ (q-vezes)} \end{aligned}$$

Portanto que o número total de iterações externas na primeira parte do algoritmo é menor que  $O\left(\frac{1}{1+\beta} L\right)$ .

Note também que um passo tipo Renegar é idêntico a um passo geral do algoritmo início; por isso, o número total de iterações tipo passo Renegar é estritamente menor que o número total de iterações do algoritmo início (ver II.5.3.4).

A segunda parte do algoritmo reiniciação gera uma seqüência de intervalos  $I_j$ ,  $J = 0, 1, 2, \dots$ , tal que:

$$\ell(I_j) = \text{comprimento de } I_j = (a-b)/2^J$$

onde:

$$b = \gamma^j < \gamma^{j-1} = a$$

Então:

$$\ell(I_j) = \left(\frac{1}{2}\right)^J (a-b).$$

Note que esta parte termina se:

$$\left(\frac{1}{2}\right)^J (a-b) < \rho$$

ou

$$j \ln\left(\frac{1}{2}\right) + \ln(a-b) < \ln(\rho)$$

Portanto que:

$$J \leq \ln(a-b) - \ln(\rho)$$

Agora pela definição de  $\rho$ ,  $\rho$  é de ordem  $(\gamma - \pi^T x)$ ; s.p.g. Suponha que  $\gamma - \pi^T x > 2^{-L}$ , então  $\ln(\rho) > -L$ ; ou  $-\ln(\rho) < L$ ; finalmente:

$$J < L + \ln(a-b)$$

O número total de iterações externas da segunda parte do algoritmo reiniciação é menor que  $L$ .

### III.5 – AGREGAÇÃO DE UM CONJUNTO DE RESTRIÇÕES

Considere de novo o problema:

$$\text{Minimizar } c^T x \tag{P}$$

Sujeito a:  $Ax \leq b$

Suponha conhecido o ponto  $x^0 \in S_k^0$  tal que:

$$\delta(x^0, k) \leq 0.1 \text{ para algum } k \in \mathbb{R}.$$

Também suponha que perturbamos o problema (P) quando agregamos o conjunto de restrições:

$$\pi_j^T x \leq \pi_j^0 \quad j = 1, \dots, p \quad (p > 1)$$

onde  $\pi_j \in \mathbb{R}^n$ ;  $\pi_j^0 \in \mathbb{R}$ ;  $j = 1, \dots, p$ .

O problema que agora se quer resolver é:

$$\text{Minimizar } c^T x \tag{P̄}$$

$$\text{Sujeito a: } \begin{aligned} Ax &\leq b \\ \pi_j^T x &\leq \pi_j^0 \quad j = 1, \dots, p \end{aligned}$$

Concretamente, desejamos fazer a reinicialização do problema (P̄).

Tal reinicialização pode ser realizada fazendo uso do algoritmo reinicialização como segue:

$j = 1$

Repita

**Reinicializar o problema**

Minimizar  $c^T x$

Sujeito a:  $Ax \leq b$   
 $\pi_j^T x \leq \pi_j^0$

**Fazer**

$A = \begin{pmatrix} A \\ \pi_j \end{pmatrix}$  e  $b = \begin{pmatrix} b \\ \pi_j^0 \end{pmatrix}$

$j := j+1$

Até que  $j = p + 1$ .

Note que a complexidade deste procedimento é igual a  $p$  vezes a complexidade do algoritmo reiniciação (ver III.3).

## III.6 – AGREGAÇÃO DE UMA VARIÁVEL NO PROBLEMA ORIGINAL

### III.6.1 – Observação

Consideremos agora o caso de agregar variáveis ao problema (P). A razão desta consideração é que em princípio estamos interessados na resolução de problemas combinatórios, mais especificamente, no problema de Matching Perfeito (PMP). Este problema pode ser formulado com uma variável por cada aresta do grafo; para grafos completos com  $n$  vértices tem-se  $O(n^2)$  variáveis.

Pensando na eficiência computacional o que se faz é considerar só um sub-conjunto de arestas do grafo (pequeno) e para provar que a solução ótima da relaxação é solução ótima do PMP.

É necessário checar que nenhuma das variáveis omitidas aparecem na relaxação. Este checamento envolve revisar o custo das variáveis omitidas. Se o custo se reduz então é necessário agregar a correspondente variável à relaxação.

### III.6.2 – Observação

De novo considere o problema  $(P_{\underline{\quad}})$ :

$$\text{Minimizar } c^T x \tag{P_{\underline{\quad}}}$$

$$\text{Sujeito a: } \begin{aligned} Ax + z &= b \\ z &\geq 0 \end{aligned}$$

Suponha que perturbamos  $(P)$  ao agregar uma nova variável  $x_0 \in \mathbb{R}$ ; i.e., considere o novo problema:

$$\text{Minimizar } c'^T x \tag{P'}^{\tilde{}}$$

$$\text{Sujeito a: } Ax + \pi x_0 \leq b$$

onde  $\pi \in \mathbb{R}^m$ .

$(P')^{\tilde{}}$  pode ser escrito como:

$$\begin{aligned} &\text{Minimizar } c^T x \\ \text{Sujeito a: } & Ax + \pi x_0 + \bar{z} = b \\ & \bar{z} \geq 0 \end{aligned}$$

Suponha que depois de uma aplicação do algoritmo início a (P), obtemos um ponto  $\hat{x} \in S_k^0$  (algum  $k$ ) tal que:

$$\delta = \delta(\hat{x}, k) \leq 0.1 \quad (\text{III.24})$$

igual que em III.2. O problema central é a reiniciação eficiente de  $(\tilde{P})$ . A visão que nós apresentamos aqui para resolver este problema não é única, mas é imediata, usando os resultados de dualidade dados nas seções anteriores, reduzimos o problema a um problema de agregação de restrições a (D) (ver III.2).

### III.6.3 – Observação

Lembramos que o problema dual (D) associado a (P) é:

$$\begin{aligned} &\text{Maximizar } -b^T y \\ & \hspace{15em} (\text{D}) \end{aligned}$$

$$\begin{aligned} \text{Sujeito a: } & A^T y + c = 0 \\ & y \geq 0 \end{aligned}$$

e o dual associado a  $(\tilde{P})$  é:

$$\begin{aligned} &\text{Maximizar } -b^T y \\ \text{Sujeito a: } & A^T y + c = 0 \quad (\tilde{\text{D}}) \\ & \pi^T y = 0 \\ & y \geq 0 \end{aligned}$$

Note que:

$$\begin{aligned} \text{GAP} &= c^T x - (-b^T y) = (-A^T y)^T x + b^T y = -y^T Ax + b^T y \\ &= -y^T (b - \tilde{z} - x_0 \pi) + b^T y = -b^T y + \tilde{z}^T y + x_0 \pi^T y + b^T y \\ &= \tilde{z}^T y. \end{aligned}$$

Se (III.24) é válida, então por III.2.3.4 existe um  $y$  viável em (D); mais ainda, por III.2.3.5 o  $\text{GAP} = \tilde{z}^T y$  é pequeno.

s.p.g. Suponha então conhecido um ponto  $y$  perto da trajetória central associada ao problema:

Minimizar  $b^T y$

( $\hat{P}$ )

Sujeito a:  $\hat{A}y \leq \hat{c}$

onde:

$$\hat{A} = \begin{bmatrix} A^T \\ -A^T \\ -I \end{bmatrix} \in \mathbb{R}^{(2n+m) \times n}; \quad \hat{c} = \begin{bmatrix} -c \\ c \\ 0 \end{bmatrix} \in \mathbb{R}^{2n+m}$$

Agora podemos aplicar duas vezes o algoritmo reiniciação a ( $\hat{P}$ ) agregando as restrições:  $\pi^T y \leq 0$  e  $-\pi^T y \leq 0$  respectivamente, e assim ter resolvido o problema de reiniciação de ( $\tilde{D}$ ). De novo pelos resultados III.2.3.4 e III.2.3.5, obtemos o desejado; i.e., um ponto dual viável (agora ponto viável de ( $\tilde{P}$ )) com um gap pequeno; portanto que o problema inicial de reiniciação fica

resolvido. Todo o resto do processo de reotimização, ao igual que a agregação de um conjunto de variáveis é análogo ao tratado de III.2 e III.5.

## CAPÍTULO IV

### ALGORITMOS DE PLANOS CORTANTES PARA RESOLVER O PROBLEMA DE MATCHING PERFEITO

#### IV.1 – INTRODUÇÃO

O problema de Matching Perfeito (PMP) é um dos problemas fundamentais da otimização combinatória. J. EDMONDS [12] provou em 1965 que o problema é polinomial, mostrando um algoritmo combinatório de ordem cúbica; mais ainda, foi o primeiro a falar sobre um "bom algoritmo"; lembre que é ao começo da década dos anos 70 que se dá início a teoria de complexidade computacional. Edmonds encontra uma descrição poliédrica do PMP, dando uma descrição completa e não redundante das facetas (faces maximais) do poliedro. O número de facetas é exponencial no número de vértices do grafo; no entanto, Edmonds usa a estrutura do conjunto das facetas para obter um algoritmo que roda em tempo polinomial no número de vértices.

É bem sabido que na prática o algoritmo de Edmonds e suas variantes resultam pouco aceitáveis; isso devido à grande complicação dos mesmos. Melhoras neste sentido são conhecidas nos trabalhos de LAWLER [41], BALL-DERIGS [3].

Uma outra metodologia para resolver o PMP é baseada em Programação Linear (fazendo uso da boa caracterização poliedral dada por EDMONDS [12]). O primeiro trabalho deste tipo foi publicado por GRÖTSCHHELL-HOLLAND [35] em 1984. Eles apresentam a implementação de um algoritmo de planos cortantes que usa o conhecido método Simplex.

Fundamentalmente, um algoritmo de planos cortantes resolve em cada iteração uma relaxação do problema e um problema de separação. Um algoritmo polinomial para resolver o problema de separação no PMP foi desenvolvido por PADBERG—RAO [46] em 1982. Tal algoritmo não é bom na prática e Grötschell—Holland usam um conjunto de heurísticas antes de fazer uso dele.

Temos então que o algoritmo de Grötschell—Holland é um algoritmo não-polinomial que resolve o problema relaxado com o Simplex e o problema de separação com boas heurísticas em lugar do algoritmo de Padberg—Rao. Na prática este método funciona muito bem em comparação com os algoritmos combinatórios de Matching.

Fazendo uso do método dos elipsóides para resolver a relaxação do problema e do algoritmo de Padberg—Rao para resolver o problema de separação, obtém-se um algoritmo polinomial de tipo planos cortantes que resolve o PMP.

Infelizmente na prática tal algoritmo é ruim. É importante notar que em cada iteração de uma metodologia de planos cortantes tem-se um programa linear que é a relaxação do Problema de Otimização Combinatória (POC). A maneira standard de resolver estes programas lineares é usando o método Simplex. Cada ponto que é viável no POC é viável na relaxação.

Geralmente entre uma iteração e outra são agregados planos cortantes; e a solução viável dessa iteração, não é mais viável na seguinte relaxação; mas como a agregação de restrições significa a agregação de variáveis no problema dual associado a tal relaxação; se agora damos o valor zero a tais variáveis,

então obtemos um novo ponto dual viável e podemos resolver o problema fazendo uso do SIMPLEX-DUAL [6]. Depois do auge dos métodos de pontos interiores na resolução do problema de programação linear, é natural a seguinte pergunta: É possível construir uma boa metodologia de planos cortantes para resolver o PMP que use um algoritmo polinomial tipo ponto interior para resolver os problemas relaxados?

Note que a maior dificuldade da aplicação de pontos interiores numa metodologia de planos cortantes está no processo de reotimização, isto é, depois de agregar um corte; no dual, nossa solução é viável mas não é interior (i.e., não é estritamente positiva), portanto é necessário desenvolver uma metodologia para obter um novo ponto interior dual viável quando um corte é agregado.

As respostas da pergunta anterior começam com o trabalho de J. MITCHELL [44]; ele usa uma variante KARMARKAR [37] para resolver o problema dual associado à relaxação do POC. Também apresenta uma metodologia para obter um ponto interior depois de agregar novas restrições ao problema relaxado.

Neste capítulo apresentamos uma metodologia de planos cortantes para resolver o PMP usando um algoritmo de pontos interiores que segue a trajetória central (algoritmo início no Capítulo II) para resolver o problema relaxado, e o algoritmo reiniciação (Capítulo III) para o problema de reotimização.

## IV.2 – O PROBLEMA DE MATCHING PERFEITO

### IV.2.1 – Formulação do Problema

Considere um grafo não dirigido  $G = (V, E)$  e um conjunto de arestas  $M \subseteq E$ , diz-se que  $M$  é um **Matching em  $G$**  se não existem em  $M$  duas arestas com um extremo comum. Seja  $m = |V|$  e  $n = |E|$ ,  $m$ -par. Um matching  $M \subseteq E$  de cardinalidade  $m/2$  se diz um **matching perfeito em  $G$** .

Note que se  $M$  é um matching perfeito então cada vértice em  $V$  é extremo de exatamente uma aresta de  $M$ .

Seja  $c: E \rightarrow \mathbb{R}$  uma função de custo sobre as arestas do grafo  $G$ . Para um matching  $M$ , o custo de  $M$  está dado por:

$$c(M) = \sum_{e \in M} c_e$$

O Problema de Matching Perfeito (PMP) consiste em achar um matching perfeito em  $G$  com custo mínimo.

### IV.2.2 – Formulação do Problema como Programa Linear

Seja  $A \in \mathbb{R}^{m \times n}$  a matriz de incidência associada a  $G$ . Seja  $c \in \mathbb{R}^n$  um custo

dado sobre as aretas de  $G$ , e  $e = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^m$ .

O problema de matching sobre  $G$  pode ser formulado como:

$$\text{Minimizar } c^T x$$

$$(PM)$$

$$\text{Sujeito a: } Ax \leq e$$

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

onde  $\{0, 1\}^n := \{x \in \mathbb{R}^n; x_j = 0 \text{ ou } x_j = 1 \ \forall j = 1, \dots, n\}$ .

Para o PMP basta considerar o sistema:

$$Ax = e$$

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

Note que o problema relaxado associado a (PM) é dado por:

$$\text{Minimizar } c^T x$$

$$\text{Sujeito a: } Ax \leq e$$

$$x \geq 0$$

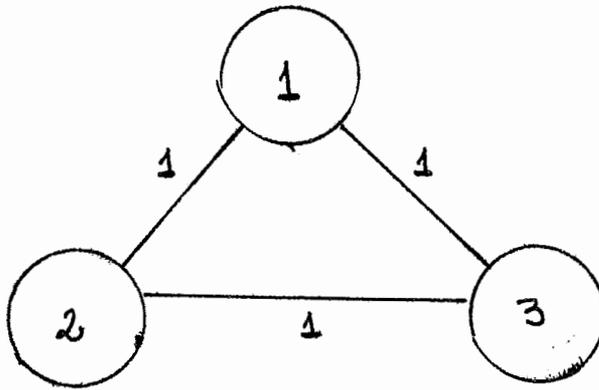
Se o grafo é bipartido então  $A$  é totalmente unimodular e os pontos extremos do poliedro

$$S := \{x \in \mathbb{R}^n: Ax \leq e, x \geq 0\}$$

representam matchings.

Para grafos em geral, isto não é verdade, como mostra o seguinte exemplo:

Considere o grafo completo  $k_3$ :



A relaxação do PM é:

maximizar  $x_1 + x_2 + x_3$

Sujeito a:  $x_1 + x_2 \leq 1$

$x_1 + x_3 \leq 1$

$x_2 + x_3 \leq 1$

$x_1, x_2, x_3 \geq 0$

A única solução ótima deste problema é:

$x := (1/2, 1/2, 1/2)$ .

#### IV.2.3 – Caracterização da Envoltória convexa das Soluções do Problema

Considere

$P_M(G) := \text{CONV}\{x \in \mathbb{R}^n: Ax \leq e, x_j = 0 \text{ ou } x_j = 1; j = 1, \dots, n\}$  e

$$P(G) := \text{CONV}\{x \in \mathbb{R}^n: Ax = e, x_j = 0 \text{ ou } x_j = 1; j = 1, \dots, n\}$$

Existe uma bijeção entre os vértices de  $P(G)$  e  $(P_M(G))$  os matchings perfeitos (matchings) em  $G$ .

O teorema central de J. EDMONDS [12] pode ser enunciado como segue:

**Teorema** (J. Edmonds, 1965)

Para cada grafo  $G = (V, E)$ ,  $P(G)$  é o conjunto de soluções do sistema:

$$(1) \quad Ax = e$$

$$(2) \quad \pi_w^T x \leq \frac{1}{2} (|w| - 1) \forall w \subseteq V: |w| \text{ é ímpar}$$

$$(3) \quad x \geq 0$$

onde

$$\pi_w \in \mathbb{R}^n$$

e,

$$\pi_{w_j} = 1 \text{ se } j \in \{j \in E: j = (v, v'); v, v' \in w\}$$

= 0 caso contrário.

Analogamente Edmonds provou que  $P_M(G)$  é o conjunto de soluções do

sistema:

$$Ax \leq e$$

$$\pi_w^T x \leq \frac{1}{2} (|w| - 1) \quad \forall w \subseteq V: |w| \text{ é ímpar}$$

$$x \geq 0$$

A restrição (2) é equivalente a restrição:

$$(4) \quad \pi_{V-w}^T x \geq 1 \quad \forall w \subseteq V: |w| \text{ é ímpar}$$

onde

$$\pi_{V-w} \in \mathbb{R}^n$$

e

$$(\pi_{V-w})_j := 1 \text{ se } j \in E: j = (v, v'); v \in w, v' \in V - w$$

$$= 0 \text{ caso contrário}$$

Na literatura as restrições (2) são conhecidas como restrições de cortes ímpares e (4) como restrições cliques.

Agora podemos escrever o PMP como:

Minimizar  $c^T x$

$$\begin{aligned} \text{Sujeito a:} \quad & Ax = e \\ & \pi_{\bar{w}}^T x \leq \frac{1}{2} (|w| - 1) \quad \forall w \subseteq V: |w| \text{ ímpar} \\ & x \geq 0 \end{aligned}$$

ou equivalente ao problema:

Minimizar  $c^T x$

$$\begin{aligned} \text{Sujeito a} \quad & Ax = e \\ & -\pi_{\bar{V}-w}^T x \leq -1, \quad \forall w \subseteq V: |w| \text{ ímpar} \\ & x \geq 0 \end{aligned}$$

Também o PM pode ser escrito como:

Minimizar  $c^T x$

$$\begin{aligned} \text{Sujeito a:} \quad & Ax \leq e \\ & \pi_{\bar{w}}^T x \leq \frac{1}{2} (|w| - 1) \quad \forall w \subseteq V: |w| \text{ ímpar} \\ & x \geq 0 \end{aligned}$$

ou equivalente ao problema:

Minimizar  $c^T x$

$$\begin{aligned} \text{Sujeito a:} \quad & Ax \leq e \\ & -\pi_{\bar{V}-w}^T x \leq -1 \quad \forall w \subseteq V: |w| \text{ ímpar} \\ & x \geq 0 \end{aligned}$$

Note que fazendo  $\bar{A} = \begin{bmatrix} A \\ -I \end{bmatrix}$  e  $\bar{b} = \begin{bmatrix} e \\ 0 \end{bmatrix}$  o PM pode ser escrito como:

Minimizar  $c^T x$

Sujeito a:  $\bar{A}x \leq \bar{b}$

$$\pi_w^T x \leq \frac{1}{2} (|w| - 1)$$

$$\forall w \subseteq V: |w| \text{ ímpar.}$$

**Lema** (Relação entre o PMP e o PM)

Seja  $B$  uma cota superior da função objetivo do PMP; i.e.,  $c^T x \leq B$ ,  $\forall x \in P(G)$  defina  $\bar{c}_e = B - c_e$ ,  $\forall e \in E$ . Então qualquer solução ótima do problema:

Minimizar  $\bar{c}^T x$

Sujeito a:  $x \in P_M(G)$

é solução ótima do problema:

Minimizar  $c^T x$

Sujeito a:  $x \in P(G)$

**Prova**

Se  $x \in P(G)$  então:

$$\bar{c}^T x = B e^T x - c^T x = B \frac{m}{2} - c^T x$$

ou

$$-c^T x = c^T x - B \frac{m}{2}$$

portanto qualquer solução ótima do problema

Minimizar  $c^T x$

Sujeito a:  $x \in P(G)$

é uma solução ótima do problema:

Minimizar  $-c^T x$

Sujeito a:  $x \in P(G)$

e vice-versa.

Se  $x$  é uma tal solução ótima, então:

$$-c^T x = c^T x - B \frac{m}{2} < B - B \frac{m}{2} = B(1 - \frac{m}{2})$$

Pela hipótese.

Seja  $x^M$  o vetor incidente de um matching  $M$  em  $G$ . Se  $M$  não é perfeito então

$$-c^T x > B(1 - \frac{m}{2})$$

e portanto  $x^M$  pode não ser ótimo do problema:

$$\begin{aligned} &\text{Minimizar } -\bar{c}^T x \\ &\text{Sujeito a: } x \in P_M(G) \end{aligned}$$

Agora todos os pontos extremos de  $P_M(G)$  são vetores incidentes de matchings; portanto, qualquer solução ótima de

$$\begin{aligned} &\text{Minimizar } -\bar{c}^T x \\ &\text{Sujeito a: } x \in P_M(G) \end{aligned}$$

é uma combinação convexa de vetores incidentes de matchings perfeitos e portanto é uma solução ótima de:

$$\begin{aligned} &\text{Minimizar } c^T x \\ &\text{Sujeito a: } x \in P(G). \end{aligned}$$

#### IV.2.4 – O Problema de Separação

##### Problema

Dado um vetor  $x \in \mathbb{R}^n$  decidir se  $x \in P(G)$ .

Se  $x$  não está em  $P(G)$ , encontrar um hiperplano que separe  $x$  de  $P(G)$ .

$P(G)$  consiste de todos os  $x \in \mathbb{R}^n$  que satisfazem (1)–(2)–(3). É muito fácil verificar se  $x$  satisfaz (1) e (3). Se  $x$  não satisfaz (1) e (3) então diretamente temos o hiperplano separador.

Note que o número de restrições tipo (2) (ou tipo (4)) é exponencial em  $|E| = n$ ; portanto é proibido verificar todas uma a uma. No entanto, o problema de separação para (4) pode ser encaixado num problema de otimização e pode ser resolvido em tempo polinomial.

Suponha que  $x$  satisfaz (1) e (3). Considere os valores  $x_e$  ( $e \in E$ ) como as capacidades dos arcos de  $G$  e agora encontre um  $w^* \subseteq V$ ,  $|w^*|$ —ímpar que minimiza  $\pi_{V-w}^T x$ ; é claro que se  $\pi_{V-w^*}^T x \geq 1$  então como  $\pi_{V-w}^T x \geq \pi_{V-w^*}^T x \geq 1$  para todo  $w \subseteq V$ ,  $|w|$ —ímpar,  $x$  satisfaz todas as desigualdades (4); caso contrário  $\pi_{V-w^*}^T x \geq 1$  deve ser agregada como restrição.

Este algoritmo de Padberg—Rao para resolver o problema de corte ímpar (*odd cut problem*) é uma variante do algoritmo de GOMORY—HU [16] para determinar um corte mínimo num grafo. A ordem deste algoritmo é  $|V|^4$ .

### IV.3 – ALGORITMO PARA O PROBLEMA DE MATCHING PERFEITO

#### IV.3.1 – Características Gerais

Considere  $G = (V, E)$  um grafo não—dirigido dado e  $|V|$  par. Seja  $c$  uma função de custo sobre as arestas de  $G$ . Suponha que o Grafo  $G$  é completo, senão complete com arestas de custo zero.

Em cada iteração do algoritmo, se resolve um programa linear usando um algoritmo polinomial tipo trajetória central, o qual é uma variante do algoritmo de Hertog—Ross—Terlaky (ver Capítulo II); depois, é analisada a solução obtida e novos cortes ou novas variáveis são agregadas. A análise se faz com a solução do problema relaxado e uma solução conhecida do PMP.

Uma solução viável inicial do PMP pode ser encontrada usando uma heurística tipo Greedy (mostraremos adiante).

É claro que os novos cortes serão agregados quando a solução do problema relaxado não for inteira; no algoritmo, chamamos as rotinas de cortes se as soluções do relaxado e do PMP são diferentes. A necessidade de agregar novas variáveis pode acontecer, pois não se trabalha com todo o conjunto de variáveis. A restrição de um subconjunto de variáveis é computacionalmente necessária nos problemas grandes. A relaxação inicial envolve só um subconjunto  $\bar{E} \subseteq E$  de arestas. Para cada vértice  $v \in V$  achamos o subconjunto formado pelos  $N$  arcos ( $1 \leq N \leq n-1$ ) de menor custo incidente a  $v$ ; então é a união desses subconjuntos.

Para garantir que existe um matching perfeito no grafo modificado, fazemos  $\bar{E} = \bar{E} \cup M$  onde  $M$  é o conjunto de arcos de um matching perfeito encontrado com um procedimento tipo Greedy.

Em [35], Grötschel e Holland escolhem  $5 \leq N \leq 10$ . A otimalidade global do problema é testada através do critério de custos reduzidos. Se temos uma solução ótima do problema relaxado, então fazemos uma revisão dos custos das variáveis omitidas. Se a função objetivo se reduz, então é necessário agregar as variáveis correspondentes à relaxação.

A determinação de cortes é feita com duas heurísticas de Grötschel–Holland; o algoritmo de Padberg–Rao só é usado no caso em que ambas heurísticas falham (isto garante a convergência finita).

Note também que por estar trabalhando com metodologia de pontos interiores, cada vez que agregamos novos cortes ou novas variáveis ao problema relaxado devemos fazer uma reiniciação do problema (encontrar um novo e bom ponto interior) antes de aplicar de novo o algoritmo início (ver Capítulo II e III).

#### IV.3.2 – Solução viável Inicial do PMP

Usamos um procedimento tipo Greedy para encontrar um matching perfeito no subgrafo  $(V, \bar{E})$  como segue:

- 1) Encontrar uma ordenação (crescente) das arestas em  $\bar{E}$  com respeito aos custos. (Use por exemplo o algoritmo Quicksort).
- 2) Encontrar um matching  $M \subseteq \bar{E}$  usando a ordenação como segue:
  - 2.1 **Escolher** uma ordenação  $\sigma$  dos vértices.
  - 2.2 **Marcar** todos os vértices como não-casados  $M := \phi$ .
  - 2.3 **Enquanto**  $i \leq m$  **fazer**

**Se**  $i$  **é** não-casados **então**

**Encontre**  $j$  com:

$$c_{ij} = \min\{c_{ij}; j' \text{ são não-casados}\}$$

**Agregar**  $(i, j)$  a  $M$ ;  $M = M \cup \{(i, j)\}$ .

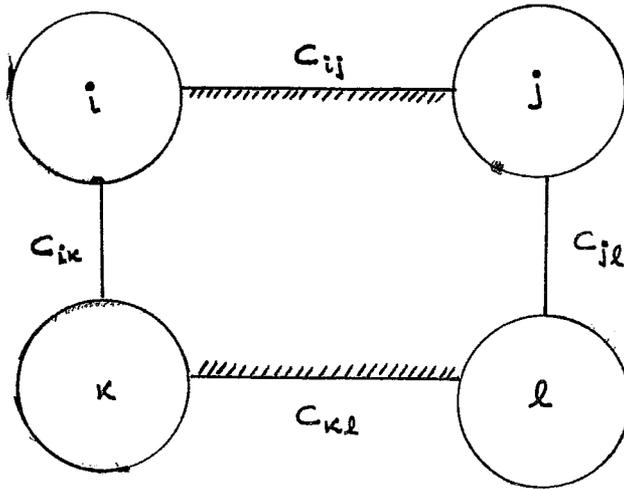
**Marcar**  $i$  e  $j$  como casados

$$i = i + 1.$$

## 2.4 Otimização do matching local

Enquanto existam  $i, j, k, \ell \in V$  com  $(i, j); (k, \ell) \in M$  e  $c_{ij} + c_{k\ell} >$   
 $> c_{ik} + c_{j\ell}$

Fazer  $M := (M \cup \{(i, k); (j, \ell)\}) - \{(i, j); (k, \ell)\}$



## 2.5 Completamento

Se  $M$  não é perfeito então

Para cada par de vértices  $i, j \in V$  tal que:

$(i, j) \in E - \bar{E}$  e  $i, j$  são não-casados

Fazer  $M := M \cup \{(i, j)\}$ .

## 2.6 (Viabilidade garantida)

Para garantir a viabilidade do sub-problema definido sobre  $\bar{E}$ ,  
 fazer  $\bar{E} = \bar{E} \cup M$ .

Note que ao final deste procedimento temos um matching perfeito  
 $M$  sobre o subgrafo  $(V, \bar{E})$ . No que segue, denotamos com  $x^M$  o

vetor incidente de  $M$ .

### IV.3.3 – Relaxação Inicial do PMP

Seja  $M \subseteq \bar{E} \subseteq E$  o matching perfeito inicial e  $x^M$  seu vetor incidente. Note que  $c^T x^M$  dá uma cota superior do valor de:

$$\begin{aligned} &\text{Minimizar } c^T x \\ & \hspace{20em} (\text{PMP}) \end{aligned}$$

$$\begin{aligned} \text{Sujeito a:} \quad & Ax = e \\ & \pi_w^T x \leq \frac{1}{2} (|w| - 1) \quad \forall w \subseteq V: |w| \text{ ímpar} \\ & x \geq 0; x \in \mathbb{R}^{\bar{E}} \end{aligned}$$

Seja  $B = c^T x^M$  e defina a função objetivo modificada  $\bar{c}^T x$  onde  $\bar{c}_e = B - c_e$  ( $e \in \bar{E}$ ); então pela relação entre o PMP e PM, qualquer solução ótima do problema:

$$\begin{aligned} &\text{Minimizar } -\bar{c}^T x \\ \text{Sujeito a:} \quad & Ax \leq e \\ & \pi_w^T x \leq \frac{1}{2} (|w| - 1) \quad \forall w \subseteq V: |w| \text{ ímpar} \\ & x \geq 0, x \in \mathbb{R}^{\bar{E}} \end{aligned}$$

é solução ótima do PMP.

No que segue, procura-se resolver o problema anterior.

Note que o problema pode ser escrito como:

Minimizar  $-\bar{c}^T x$

Sujeito a:  $\bar{A}x \leq \bar{b}$

$$\pi_{\bar{w}}^T x \leq \frac{1}{2} (|w| - 1) \quad \forall w \subseteq V: |w| \text{ ímpar}$$

onde:

$$\bar{A} = \begin{bmatrix} A \\ -I \end{bmatrix} \in \mathbb{R}^{(m+\bar{n}) \times \bar{n}}; \bar{b} = \begin{bmatrix} e \\ 0 \end{bmatrix} \in \mathbb{R}^{m+\bar{n}}; \bar{n} = |\bar{E}|$$

e o problema relaxado que consideramos é:

Minimizar  $-\bar{c}^T x$

Sujeito a:  $\bar{A}x \leq \bar{b}$

O ponto inicial para resolver este problema com o algoritmo início é escolhido como no Capítulo II.

#### IV.3.4 – Arredondamento de Soluções

O algoritmo melhora a solução  $x^M$  do problema combinatório com um arredondamento da solução ótima do problema relaxado  $x$  a uma solução vizinha do problema combinatório.

Com o arredondamento das soluções, temos que:

- i) Como a solução ótima do problema relaxado pode não corresponder a um ponto viável do problema combinatório, o arredondamento assegura que a

solução obtida ao final do algoritmo é uma solução ótima do problema combinatório e não um ponto próximo na envoltória convexa perto desta.

(ii) As rotinas de separação podem não ter que se chamadas.

É claro que  $k^M := -c^T x^M = c^T x^M - B \frac{m}{2} = B - B \frac{m}{2} = B(2-m)/2$  é

uma cota superior do valor de

$$\text{Minimizar } z = -c^T x$$

$$\text{Sujeito a: } Ax \leq e$$

$$\pi_w^T x \leq \frac{1}{2} (|w| - 1) \quad \forall w \subseteq V: |w| \text{ ímpar}$$

$$x \geq 0$$

$$x \in \mathbb{R}^{|E|}$$

e também de sua relaxação.

Se  $x \neq x^M$  então arredondamos  $x$  para encontrar um matching perfeito  $\bar{M}$  cujo valor comparamos com  $k^M$ . Seja  $x^{\bar{M}}$  o vetor de incidência de  $\bar{M}$ . Seja  $k^{\bar{M}} := -c^T x^{\bar{M}}$ . Se  $k^{\bar{M}} < k^M$  então atualizamos  $x^M$  com  $x^{\bar{M}}$  e  $k^M$  com  $k^{\bar{M}}$ . Se  $k^{\bar{M}} < z$  então fazemos  $x = x^{\bar{M}}$ ,  $z = k^{\bar{M}}$ .

### Determinação de $\bar{M}$

1) Inicialização  $\bar{M} := \phi$

2) Para  $e \in \bar{E}$

. Se  $x_e > 0.5$  fazer  $\bar{M} = \bar{M} \cup \{e\}$

3) Se  $|\bar{M}| < m/2$  então

. Para cada par de vértices  $i, j \in V$  tal que:  $(i,j) \in E - (\bar{E} \cup \bar{M})$  e  $i, j$  são não-casados

Fazer  $\bar{M} = \bar{M} \cup \{(i,j)\}$ .

4) Enquanto existam  $i, j, k, \ell \in V$  com  $(i,j), (k,\ell) \in \bar{M}$  e  $c_{ij} + c_{k\ell} > c_{ik} + c_{j\ell}$ ; fazer  
 $\bar{M} = (\bar{M} \cup \{(i,k), (j,\ell)\}) - \{(i,j), (k,\ell)\}$

#### IV.3.5 – Rotinas de Separação

Suponha no que segue que a solução ótima do problema relaxado  $x$  é fracionária. Agora construímos um grafo capacitado  $G_x$  o qual represente a solução do PL.  $G_x$  será a entrada da fase de reconhecimento de cortes (ou planos cortantes) no algoritmo da forma seguinte:

Seja  $G_x = (V, E_x)$

onde:

$E_x := \{e \in \bar{E}; x_e > 0\}$

Defina as capacidades dos arcos como:

$\rho : E_x \rightarrow \mathbb{R}_+$

tal que

$$\rho(e) := x_e$$

A solução ótima do PL  $x$  é fracionária e portanto não pertence a  $P(G)$ . Desta forma temos que achar um plano cortante.

As rotinas de separação envolvem três estados:

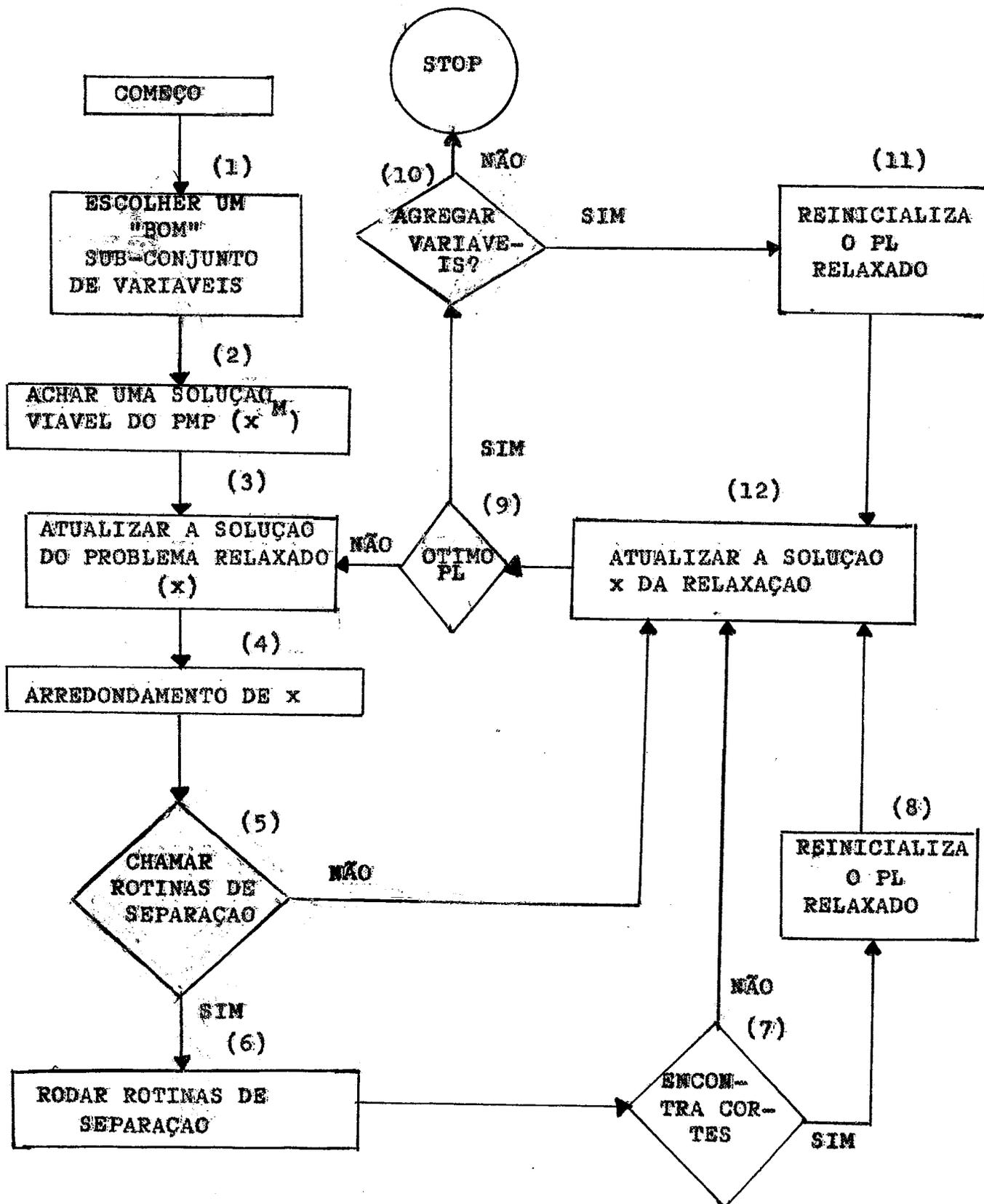
- 1) Defina  $E_x$  como antes e use DFS (Depth First Search) para encontrar as componentes do grafo  $(V, E_x)$ . Se qualquer das componentes ímpares (componente com um número ímpar de nós) viola a correspondente restrição de corte ímpar então agregar esta restrição à formulação.
- 2) Se no primeiro passo não encontramos nenhuma restrição violada, então definimos  $E'_x := \{e \in \bar{E}; x_e > 0.3\}$ . Usamos DFS para achar as componentes do grafo  $(V, E'_x)$ . Se qualquer das componentes ímpares viola a correspondente restrição de corte ímpar então agregamos esta restrição à formulação.
- 3) Se nenhuma restrição de corte ímpar é encontrada nos dois passos anteriores. Então chamar o procedimento de PADBERG-RAO [46].

### Nota

Agora temos todas as condições para apresentar uma descrição esquemática do algoritmo. As possíveis variações do esquema geral dependem

das implementações do mesmo.

#### IV.3.6 – Descrição Esquemática do Algoritmo



## Comentários do Algoritmo

### Caixa 1

Em alguns problemas o trato com o número de variáveis e/ou restrições é proibido. É por isto que se faz necessário tirar os valores de algumas delas antes de formar o PL relaxado inicial. No caso do problema de matching perfeito temos uma variável por cada aresta do grafo; no grafo completo com  $m$  vértices temos então  $O(m^2)$  variáveis. Mas é claro que muitas destas arestas não aparecem no matching perfeito ótimo e portanto que suas correspondentes variáveis podem tomar valor zero.

O PL inicial que se escolhe envolve um subconjunto de arestas  $\bar{E} \subseteq E$  formado por exemplo da seguinte forma: para cada nó, encontrar as  $N$  arestas de menor custo que tem a ele como extremos.  $\bar{E}$  é a união destes subconjuntos.

O  $N$  que escolhem Grötschel–Holland é

$$5 \leq N \leq 10$$

note que em geral  $1 \leq N \leq n-1$ .

### Caixa 2

Usamos um algoritmo Greedy para determinar uma solução viável  $(x^M)$  do PMP (ver IV.3.2).

**Caixa 3**

Usamo o algoritmo início (Capítulo II) para atualizar a solução do problema relaxado ( $x$ ).

**Caixa 4**

Se  $x \neq x^M$  então fazemos o arredondamento de  $x$  para melhorar  $x^M$  (ver IV.3.4).

**Caixa 5**

Chamamos as rotinas de separação se  $x \neq x^M$ .

**Caixa 6**

Ver IV.3.5.

**Caixa 7**

Se encontramos planos cortantes na caixa 6 então vamos à caixa 8, caso contrário vamos à caixa 12.

**Caixa 8**

Agregamos as restrições encontradas na caixa 6 e aplicamos o algoritmo reiniciação generalizado. Ao final deste, temos um bom ponto interior desde o qual podemos aplicar o algoritmo início para atualizar a solução do PL relaxado.

**Caixa 9**

Nesta caixa checamos se o GAP é suficientemente pequeno. Se é assim vamos à caixa 10, caso contrário vamos à caixa 3.

**Caixa 10**

Aplicamos o critério de custos reduzidos e determinamos se existem novas variáveis a serem agregadas. Se é assim, vamos à caixa 11, senão paramos.

**Caixa 11**

Agregamos as variáveis obtidas na caixa 10 e obtemos um bom ponto interior (como no Capítulo III) desde o qual podemos aplicar o algoritmo início para atualizar a solução do PL relaxado.

**Caixa 12**

A solução do PL relaxado é atualizada. Se chegamos a ela desde a caixa 5 ou 7, então calculamos um novo ponto. Note que o algoritmo início (Capítulo II) pode ser aplicado neste caso, posto que o ponto  $x = x^M$  satisfaz as condições dadas em II.6. A folga  $z$  associada a  $x$  pode ser escrita como combinação convexa das folgas associadas as soluções básicas viáveis, i.e.,  $z = \sum_k \lambda_k z^k$ ,  $\lambda_k \geq 0$ ,

$\sum_k \lambda_k = 1$ ; e as coordenadas de cada  $z^k$  satisfaz:

$$z_j^k \geq 2^{-L} \quad (j = 1, \dots, n)$$

Portanto que, para cada  $j = 1, \dots, n$

$$z_j = \sum_k \lambda_k z_j^k \geq \sum_k \lambda_k 2^{-L} = 2^{-L}$$

## CAPÍTULO V

### CONCLUSÕES

Depois do auge dos métodos pontos interiores para resolver o problema de programação linear, foi natural o planteamento sobre a aplicação desta metodologia em áreas como: otimização combinatória, programação quadrática, complementariedade linear, programação convexa, etc. A aplicação das idéias de pontos interiores a problemas combinatórios é limitada. Isto deve-se fundamentalmente as sérias dificuldades do problema de reiniciação que tem que ser resolvido numa metodologia de planos cortantes para resolver o problema inteiro.

Cada vez que um novo corte é encontrado, o ponto dessa iteração se faz não-viável e uma inicialização "eficiente" do algoritmo ponto interior que seja usado é difícil. Nesta dissertação tais dificuldades foram resolvidas. Os resultados obtidos tanto a nível teórico como a nível de algoritmos, devem-se as reflexões sobre uma inicialização eficiente. Na classe dos algoritmos de trajetória central (usados aqui) uma inicialização eficiente significa a obtenção de um ponto interior perto da trajetória central associada ao problema com gap baixo.

Até agora (julho de 1991) não conhecemos outro trabalho a não ser o de J. MITCHELL [44] (tese de doutorado defendida em 1988 na Universidade de Cornell (USA)) que incursione no problema da introdução dos algoritmos de pontos interiores na metodologia de planos cortantes para resolver problemas combinatórios. Nesse trabalho apresenta-se uma variante Karmarkar para resolver o problema dual associado à relaxação do problema combinatório. Em geral, a maneira dele pensar no problema de reotimização é a seguinte: a

agregação de planos cortantes na relaxação significa a agregação de variáveis ao problema dual, as quais com valor igual a zero permitem obter um ponto dual viável mesmo que não seja interior; depois desde este ponto ele mostra como obter uma direção para achar um novo ponto interior dual viável; mesmo que este ponto possa estar muito longe da solução ótima do problema perturbado, o qual não é nada bom.

Pela nossa parte podemos dizer que os objetivos centrais da tese foram todos plenamente cumpridos.

### **Primeiro**

Levantamos uma teoria consistente sobre o processo de reotimização na metodologia de planos cortantes com uso de pontos interiores.

### **Segundo**

Apresentamos pela primeira vez um algoritmo de pontos interiores tipo trajetória central de  $O(\sqrt{m} L)$  ( $m$  - número de restrições do PL e  $L$  o tamanho do PL) para a reiniciação do problema.

### **Terceiro**

Descrevemos em detalhe uma aplicação de todos os resultados obtidos, através de um algoritmo de planos cortantes que usa pontos interiores para resolver o bem conhecido problema de matching perfeito.

Os resultados obtidos sobre reotimização fazendo uso de métodos pontos

interiores não encerram o estudo; ainda faltaria:

- i) Programar e testar os algoritmos início (Capítulo II) e reinício (Capítulo III).
- ii) Programar e testar as heurísticas para a determinação de cortes no problema de matching perfeito (Capítulo IV).
- iii) A extensão dos resultados centrais de reotimização com uso dos métodos de redução potencial.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [ 1 ] I. ADLER, N. KARMARKAR, M. RESENDE and G. VEIGA – An Implementation of Karmarkar's Algorithm for Linear Programming. *Mathematical Programming*, 44, (1989).
- [ 2 ] K. ANSTREICHER – Analysis of a Modified Karmarkar Algorithm for Linear Programming. Technical Report Series B84, Yale School of Organization and Management, New Haven, CT, (1985).
- [ 3 ] M. BALL and U. DERIGS – An Analysis of Alternate Strategies for Implementing Matching Algorithms. *Networks*, 13: 517–549, (1983).
- [ 4 ] E. R. BARNERS – A Variation of Karmarkar's Algorithm for Solving Linear Programming Problems. *Mathematical Programming*, 36: 174–182, (1986).
- [ 5 ] D. BAYER and J. C. LAGARIAS – The Non-Linear Geometry of Linear Programming, I. Affine and Projective Scaling Trajectories, II. Legendre Transform Coordinates, III. Central Trajectories. Preprints, AT&T Bell Laboratories, Murray Hill, NY, (1986).
- [ 6 ] P. BREGALDA A. DE OLIVEIRA and C. BORNSTEIN – Introdução à Programação Linear. 3ª. Edição, Editora Campus, (1988).
- [ 7 ] T. M. CAVALIER and A. L. SOYSTER – Some Computational Experience and a Modification of the Karmarkar Algorithm. Presented at the 12<sup>th</sup> Symposium on Mathematical Programming,

Cambridge, MA, (1985).

- [ 8] G. DANTZIG - Linear Programming and Extensions. Princeton University Press, Princeton, N.J. (1963).
  
- [ 9] G. DE GHELLINCK and J. P. VIAL - A Polynomial Newton Method for Linear Programming. *Algorithmica*, 1: 425-453, (1986).
  
- [10] D. CLEN HERTOOG, C. ROOS and T. TERLAKY - A Potential Reduction Variant of Renegar's Short-Step Path - Following Method for Linear Programming. Report 90-14 Faculty of Technical Mathematics and Informatics. Delft. (1990).
  
- [11] I. I. DIKIN - Iterative Solution of Problems of Linear and Quadratic Programming. *Soviet Mathematics Doklady*, 8: 674-675, (1967).
  
- [12] J. EDMONDS - Maximum Matching and a Polyhedron with 0,1 Vertices. *Journal of Research National Bureau of Standards*, 69B: 125-130, (1965).
  
- [13] R. M. FREUND - A Potential - Function Reduction Algorithm for Solving a Linear Program Directly from an Infeasible 'Warm Start'. Technical Report 3079-89-MS, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, (1989).
  
- [14] D. GAY - A Variant of Karmarkar's Linear Programming Algorithm for Problems in Standard Form. *Mathematical Programming*, 37: 81-89, (1987).

- [15] G. H. GOLUB and C. F. VAN LOAN – Matrix Computations. Johns Hopkins University Press, Baltimore, (1983).
  
- [16] R. E. GOMORY and T. C. HU – Multi Terminal Network Flows. Journal of the Society for Industrial and Applied Mathematics, 9: 551–571, (1961).
  
- [17] R. E. GOMORY – Outline of an Algorithm for Integer Solutions to Linear Programs. Bulletin of the American Mathematical Society, 64: 275–278, (1958).
  
- [18] C. GONZAGA – Convergence of the Large Steps Primal Affine Scaling Algorithms For Primal Non-Degenerate Linear Programming. Internal Report, COPPE – Federal University of Rio de Janeiro, Brasil, (1990).
  
- [19] C. GONZAGA – Interior Point Algorithms for Linear Programming with Inequality Constraints. Internal Report ES-140/88, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro, Brasil, January 1988. To appear in Mathematical Programming.
  
- [20] C. GONZAGA – Large-Steps Path-Following Algorithms for Linear Programming: Barrier Function Method. Internal Report, COPPE/UFRJ, Brasil, (1989). To appear in SIAM Journal on Optimization.

- [21] C. GONZAGA – Large–Steps Path–Following Algorithms for Linear Programming: Potential Reduction Method. Internal Report, COPPE/UFRJ, Brasil, (1989). To appear in SIAM Journal on Optimization.
- [22] C. GONZAGA – On Lower Bound Updates in Primal Potential Reduction Methods for Linear Programming. Internal Report, COPPE/UFRJ, Brasil, (1990). To appear in Mathematical Programming.
- [23] C. GONZAGA – Polynomial Affine Algorithms for Linear Programming. Mathematical Programming, 49 (1990), 7–21.
- [24] C. GONZAGA – Search Directions for Interior Linear Programming Methods. Algorithmica (1991) 6: 153–181.
- [25] C. GONZAGA and T. J. TODD – An  $O(\sqrt{n}L)$  – Iteration Large–Step Primal–Dual Affine Algorithm for Linear Programming. Technical Report 862. School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, (1989).
- [26] C. GONZAGA – A Simple Presentation of Karmarkar's Algorithm. Preprint, COPPE – Federal University of Rio de Janeiro, (1988).
- [27] C. GONZAGA – An Algorithm for Solving Linear Programming Problems in  $O(n^2L)$  Operations. In N. Megiddo, Editor, Progress in Mathematical Programming Interior Point and Related Methods, Chapter 1, Springer Verlag, Berlin, (1989).

- [28] C. GONZAGA – Conical Projection Algorithms for Linear Programming. *Mathematical Programming*, 43: 151–173, (1988).
- [29] C. GONZAGA – Path Following Methods for Linear Programming. Report, COPPE/UFRJ, Brasil, (1990). To appear in *SIAM Review*.
- [30] C. GONZAGA – Algoritmos de Pontos Interiores para Programação Linear. 17<sup>o</sup>. Colóquio Brasileiro de Matemática, CNPq-IMPA, (1988).
- [31] M. GRÖTSCHEL, O. HOLLAND – A Cutting Plane Algorithm for Minimum Perfect 2 – Matchings. *Computing* 39, 327–344, (1987).
- [32] M. GRÖTSCHEL, L. LOVÁSZ, A. SCHRIJVER – The Ellipsoid Method and its Consequences in Combinatorial Optimization. *Combinatorica* 1, 169–197, (1981).
- [33] M. GRÖTSCHEL, L. LOVÁSZ, A. SCHRIJVER – *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, (1988).
- [34] M. GRÖTSCHEL, M. W. PADBERG – On the Symmetric Travelling Salesman Problem I: Inequalities. *Math. Programming* 16, 265–280, (1979).
- [35] M. GRÖTSCHEL and O. HOLLAND – Solving Matching Problems with Linear Programming. *Mathematical Programming*, 33: 243–259,

(1985).

- [36] N. KARMAKAR, M. RESENDE and K. RAMAKRISHMANN – An Interior Point Algorithm to Solve Computationally Difficult Set Covering Problems. Manuscript, AT&T Bell Laboratories, Murray Hill, NY, (1989).
  
- [37] N. KARMAKAR – A New Polynomial Time Algorithm for Linear Programming. *Combinatorica*, 4: 373–395, (1984).
  
- [38] L. KHACHIYAN – A Polynomial Algorithm for Linear Programming. *Soviet Mathematics Doklady*, 20: 191–194, (1979).
  
- [39] M. KOJIMA, S. MIZUNO and A. YOSHISE – A Primal–Dual Interior Point Method for Linear Programming. In N. Megiddo, Editor, *Progress in Mathematical Programming, Interior Point and Related Methods*, Chapter 2, Springer Verlag, Berlin, (1989).
  
- [40] A. H. LAND and A. G. DOIG – An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, 28: 497–520, (1960).
  
- [41] E. L. LAWLER – *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, New York, (1976).
  
- [42] N. MEGIDDO and M. SHUB – Boundary Behaviour of Interior Point Algorithms in Linear Programming. *Mathematics of Operations Research*, 14(1): 97–146, (1989).

- [43] N. MEGIDDO — Pathways to the Optimal Set in Linear Programming. In M. Megiddo, Editor, Progress in Mathematical Programming. Interior Point and Related Methods, Chapter 8, Springer-Verlag, Berlin, (1989).
- [44] J. E. MITCHELL — Karmarkar's Algorithm and Combinatorial Optimization Problems. Thesis Doctor of Philosophy Cornell University, USA, August (1988).
- [45] R. MONTEIRO, I. ADLER and M. RESENDE — A Polynomial-Time Primal-Dual Affine Scaling Algorithm for Linear and Convex Quadratic Programming and its Power Series Extension. Mathematics of Operations Research, 15: 191-214, (1990).
- [46] M. W. PADBERG and M. R. RAO — Odd Minimum Cut-Sets and b-Matchings. Mathematics of Operations Research, 7: 67-80, (1982).
- [47] C. H. PAPADIMITRIOU and K. STEIGLITZ — Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall, Englewood Cliffs, New Jersey, (1982).
- [48] J. RENEGAR — A Polynomial-Time Algorithm Based on Newton's Method for Linear Programming. Mathematical Programming, 40: 59-94, (1988).
- [49] R. T. ROCKAFELLAR — Convex Analysis. Princeton University Press, Princeton, NY, (1970).

- [50] A. SCHRIJVER – Theory of Linear and Integer Programming. John Wiley, Chichester, (1986).
- [51] M. TODD and B. BURRELL – An Extension of Karmarkar's Algorithm for Linear Programming Using Dual Variables. *Algorithmica*, 1: 409–424, (1986).
- [52] M. J. TODD and Y. YE – A Centered Projective Algorithm for Linear Programming. Technical Report 763, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, (1987).
- [53] P. VAIDYA – An Algorithm for Linear Programming Which Requires  $O(((m+n)n^2 + (m+n)^{1.5}n)L)$  Arithmetic Operations. Preprint, AT&T Bell Laboratories, Murray Hill, NJ, (1987).
- [54] R. J. VANDERBEI, M. J. MEKTON and B. A. FREEDMAN – A Modification of Karmarkar's Linear Programming Algorithm, *Algorithmica*, 1: 395–407, (1986).
- [55] R. VANDERBEI and J. C. LAGARIAS – I. I. Dikin's Convergence Results for the Affine-Scaling Algorithm. Preprints, AT&T Bell Laboratories, Murray Hill, NY, 1988.
- [56] Y. YE and M. KOJIMA – Recovering Optimal Dual Solutions in Karmarkar's Polynomial Algorithm for Linear Programming. *Mathematical Programming*, 39: 305–317, (1987).

- [57] Y. YE — An  $O(n^3L)$  Potential Reduction Algorithm for Linear Programming. Manuscript, Dept. of Management Sciences, the University of Iowa, Iowa City, Iowa, (1988). To appear in Mathematical Programming.
- [58] Y. YE — A Class of Projective Transformations for Linear Programming. Manuscript, Dept. of Management Sciences, the University of Iowa, Iowa City, Iowa, (1989).