

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E SISTEMAS

UM MÉTODO HEURÍSTICO PARA RESOLUÇÃO DE PROBLEMAS DE
PROGRAMAÇÃO LINEAR ATRAVÉS DO MÉTODO DE
PROJEÇÕES ORTOGONAIS

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

CLAUDIO LOESCH

FLORIANÓPOLIS
SANTA CATARINA - BRASIL
NOVEMBRO DE 1981

UM MÉTODO HEURÍSTICO PARA RESOLUÇÃO DE PROBLEMAS DE
PROGRAMAÇÃO LINEAR ATRAVÉS DO MÉTODO
DE PROJEÇÕES ORTOGONAIS

CLAUDIO LOESCH

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO
TÍTULO DE

"MESTRE EM ENGENHARIA"

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

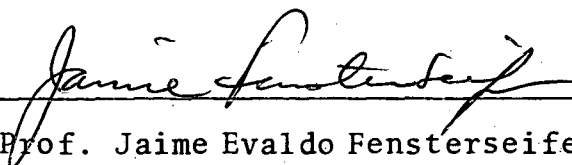


Prof. John Robert Mackness, Ph.D.
Coordenador do Programa
de Pós-Graduação em Engenharia de Produção

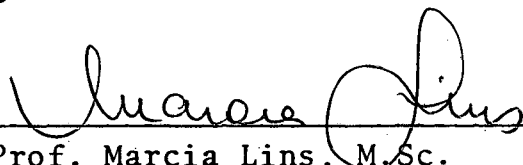
BANCA EXAMINADORA:



Prof. Wilhelm Rödder, Ph.D.
Presidente



Prof. Jaime Evaldo Fensterseifer, Ph.D.



Prof. Marcia Lins, M.Sc.



0.249.296-9

UFSC-BU

À minha esposa

Sueli

Aos meus pais

Walter e Amanda

A G R A D E C I M E N T O S

Manifesto meus sinceros agradecimentos às seguintes pessoas e instituições:

- Ao Prof. WILHELM RÖDDER, pela orientação e críticas dadas no transcorrer de todo este trabalho;
- À ACAFE, pela concessão da bolsa de estudos;
- À FURB, pelas facilidades que concedeu no sentido de tornar possível a realização da Pós-Graduação;
- Aos Professores integrantes da Banca Examinadora, pelos valiosos comentários e sugestões, que permitiram aperfeiçoar este trabalho;
- A todas as pessoas que, direta ou indiretamente, contribuíram para a realização deste trabalho.

RESUMO

Atualmente muitas das pesquisas científicas na área da Pesquisa Operacional estão voltadas à procura de novos métodos não convencionais que levem à solução do tradicional problema de programação linear. Desde 1947, ano da criação do método simplex, o mesmo tem sido o centro polarizador de desenvolvimento na programação linear e inteira.

O presente trabalho visa desenvolver um modelo de resolução do mesmo problema fundamentado na Álgebra Linear, no entanto através de um enfoque totalmente diverso do método simplex. Utiliza a noção de projeções ortogonais sobre hiperplanos de restrições violadas com a intenção de aproximar-se da solução ótima e assim pesquisar o conjunto de restrições que compõe o vértice da solução ótima.

Um programa computacional foi elaborado sobre o método proposto e os resultados obtidos são comentados, assim como aspectos de sua complexidade-tempo.

Finalmente, críticas são feitas sobre aspectos positivos e negativos, assim como sugestões sobre como o método poderia ser aperfeiçoado por pesquisas posteriores.

ABSTRACT

Most of the scientific research being conducted nowadays in the area of Operations Research reveal a tendency to look for new and non-conventional methods leading to the solution of the traditional problem of linear programming.

From 1947, when the simplex-method was created, this method has become the polarizing center of developments of integer and linear programming.

The present work aims at developing a model for the solution of the same problem, a model based on Linear Algebra, but conducted through a focus totally opposite to the simplex-method. It uses the notion of orthogonal projections on hyperplanes of violated restrictions intending to come close to the best solutions and consequently to research a set of the restrictions that compound the vertex of the optimal solution.

A computing program was elaborated involving the suggested method and the results obtained are analysed as well as the aspects of its time-complexity.

Finally, criticisms are made about the positive and negative aspects, as well as suggestions about how the method can be improved by further research.

S U M Á R I O

	pag.
LISTA DE FIGURAS E TABELAS	x
CAPÍTULO I	
1. INTRODUÇÃO	1
1.1. Origem do Trabalho	1
1.2. Objetivo do Trabalho	1
1.3. Importância do Trabalho	2
1.4. Estrutura do Trabalho	2
1.5. Limitações do Trabalho	5
1.6. Metodologia do Trabalho	5
CAPÍTULO II	
2. HISTÓRICO DA PROGRAMAÇÃO LINEAR	8
2.1. Surgimento	8
2.2. Esforços e Resultados Iniciais	8
2.3. Ampliação dos Horizontes	10
2.4. Tendências Não Ortodoxas	11
CAPÍTULO III	
3. PROGRAMAÇÃO LINEAR E MÉTODO SIMPLEX	14
3.1. O Problema da Programação Linear	14
3.2. Exemplo com Duas Variáveis	15
3.3. Resolução Geométrica	18
3.4. O Método Simplex de Resolução	19
3.5. Teoria do Método Simplex	23
3.6. Exemplo de Aplicação do Algoritmo	26

CAPÍTULO IV

4.	TEOREMAS SOBRE PROJEÇÕES EM HIPERPLANOS	28
4.1.	Terminologia Usada	28
4.2.	Definições, Lemas e Teoremas	29

CAPÍTULO V

5.	O PROBLEMA EQUIVALENTE PRIMAL-DUAL	39
5.1.	O Problema Primal e o Problema Dual	39
5.2.	O Problema Equivalente	41
5.3.	Um Exemplo de Problema Equivalente Primal-Dual e sua Solução Geométrica	44
5.4.	A Submatriz Quadrada da Solução do Problema de Programação Linear	45
5.4.1.	Base ótima e submatriz quadrada	45
5.4.2.	Um exemplo numérico	47
5.4.3.	Lema da univocidade	49
5.5.	Submatriz Quadrada e o Par Primal-Dual	50

CAPÍTULO VI

6.	O ALGORITMO APROXIMATIVO DE PROGRAMAÇÃO LINEAR .	55
6.1.	Metodologia do Algoritmo	55
6.2.	Descrição do Algoritmo	62
6.3.	Um Exemplo Completo	65
6.4.	Complexidade-Tempo de suas Partes	67

CAPÍTULO VII

7.	RESULTADOS DA APLICAÇÃO DO ALGORITMO	70
7.1.	Resultados Computacionais Obtidos	70

7.2.	Interpretação dos Resultados	73
------	------------------------------------	----

CAPÍTULO VIII

8.	CONCLUSÃO	75
----	-----------------	----

8.1.	Análise dos Resultados	75
------	------------------------------	----

8.2.	Considerações Finais	76
------	----------------------------	----

	REFERÊNCIAS BIBLIOGRÁFICAS	78
--	----------------------------------	----

ANEXO 1	Especificação dos Cartões de Entrada	81
---------	--	----

ANEXO 2	Programa Computacional do Algoritmo Aproximativo de Programação Linear	85
---------	--	----

ANEXO 3	Listagem de Saída do Programa	93
---------	-------------------------------------	----

LISTA DE FIGURAS E TABELAS

	pag.
FIGURA 1	Gráfico de um problema de PL 18
FIGURA 2	Gráfico de um problema equivalente primal-dual . 45
FIGURA 3	Fluxograma do algoritmo aproximativo de programação linear 64
TABELA 1	Dados estatísticos sobre os problemas testados 72
TABELA 2	Ocorrência de resoluções de sistemas lineares 74

CAPÍTULO I

1. INTRODUÇÃO

1.1. ORIGEM DO TRABALHO

A programação linear pode ser considerada o pivô do desenvolvimento da Pesquisa Operacional, que, apesar de constituir um ramo de conhecimentos relativamente novo, desenvolveu-se de maneira muito rápida. Preocupada com os processos de otimização, mantém um interrelacionamento muito profundo com outras ciências, assim como encontra campos de aplicação dos mais diversificados.

Nestas aplicações, normalmente lida-se com um grande volume de informações que constituem os dados de entrada do problema, o que impossibilita ou torna inconveniente o seu tratamento manual, exigindo a realização do processamento eletrônico.

Assim, em virtude do alto custo por tempo de processamento do computador, procuram-se processos de cálculo mais rápidos e algoritmos mais eficientes.

1.2. OBJETIVO DO TRABALHO

No estágio atual de desenvolvimento das pesquisas em torno da programação linear, não se determinou ainda nenhum método de resolver seus problemas diretamente através de aplicações de

fórmulas matemáticas que forneçam sua solução. Outrossim, o emprego de algoritmos tem-se mostrado mais eficaz nestes casos.

Os objetivos do presente trabalho são:

- a. aprimorar bases teóricas que conduzam a um algoritmo de resolução do problema de programação linear por métodos não convencionais;
- b. conduzir a conceitos novos relacionados ao aspecto da teoria da dualidade, de modo a encarar o par primal - dual como um único problema a ser resolvido;
- c. propor um algoritmo de resolução do problema de programação linear baseado nos dois itens precedentes, bem como discutir seus pontos positivos e negativos.

1.3. IMPORTÂNCIA DO TRABALHO

Longe de pretender esgotar o assunto, o presente trabalho poderá servir como fonte para pesquisas posteriores que aprimorem tanto a parte teórica quanto o algoritmo apresentado, de modo que se consiga chegar a um instrumento matemático que demonstre eficiência relativa ou absoluta em comparação aos demais algoritmos competitivos. Neste caso, a implantação de programas computacionais nele baseados, desde que apresentem economia em termos de tempo de execução justificariam a pesquisa desenvolvida.

1.4. ESTRUTURA DO TRABALHO

O presente trabalho foi desenvolvido em oito capítulos.

O primeiro capítulo visa definir os objetivos do trabalho apresentado, bem como sua importância e limitações.

O capítulo seguinte, denominado "Histórico da Programação Linear" tem por objetivo familiarizar o leitor com o desenvolvimento histórico e a situação atual das pesquisas em torno da programação linear. Apresenta tanto os desenvolvimentos teóricos realizados como também as dificuldades encontradas e vencidas em termos de processamento eletrônico.

O terceiro capítulo apresenta o problema de programação linear e sua modelagem matemática. Uma resolução geométrica é apresentada, para um problema de duas variáveis. A seguir, apresentam-se as propriedades algébricas válidas para a programação linear, que servem de suporte ao método simplex de resolução.

Devido à sua importância histórica e sua funcionalidade, o algoritmo simplex é apresentado neste capítulo e um exemplo é resolvido com o uso do mesmo.

O quarto capítulo introduz a terminologia utilizada para o desenvolvimento do restante da dissertação. A seguir, desenvolve definições, lemas e teoremas sobre projeções ortogonais em hiperplanos. Aspectos de distância euclidiana e convergência em uma seqüência de projeções ortogonais sobre hiperplanos constituem os resultados essenciais deste capítulo.

O quinto capítulo, denominado "O Problema Equivalente Primal-Dual" começa por apresentar o par primal-dual da programação linear e revisa algumas de suas propriedades mais importantes. A seguir, apresenta o problema equivalente primal-dual, cujo ve-

tor de variáveis resulta da concatenação do vetor de variáveis do primal com o vetor de variáveis do dual, e possui um conjunto de restrições que une as restrições do primal com as do dual, juntamente com a igualdade de suas funções-objetivas.

Propriedades do problema equivalente primal-dual são apresentadas, bem como sua relação com os problemas primal e dual. O capítulo termina por apresentar a relação entre as bases das soluções ótimas limitadas do problema primal e do dual com a submatriz quadrada do problema equivalente primal-dual que o resolve.

No capítulo seis é apresentado o algoritmo aproximativo de programação linear, precedido pela descrição metodológica do mesmo e de dois teoremas auxiliares. A seguir, aspectos de sua complexidade-tempo são discutidos.

Nos dois últimos capítulos, resultados positivos e negativos de sua funcionalidade são apresentados e discutidos, assim como sugestões para o seu aprimoramento.

Seguem ao último capítulo três anexos. Todos são relativos ao programa computacional do algoritmo apresentado, em linguagem FORTRAN IV, desenvolvido pelo autor deste trabalho. O primeiro fornece instruções quanto à entrada de dados, o segundo apresenta o programa, e o último apresenta um exemplo de saída dos resultados.

É apresentada uma bibliografia que faz referência às obras relacionadas ao presente trabalho.

1.5. LIMITAÇÕES DO TRABALHO

Não foi realizada nenhuma comparação de tempo real de execução de problemas, através do algoritmo proposto e outros já conhecidos.

O programa computacional apresentado admite problemas de programação linear de até 99 restrições e 99 variáveis. Trabalha com precisão simples e nenhuma pesquisa sobre a influência da precisão computacional no tratamento do problema foi realizada pelo autor.

Os problemas testados pelo autor deste trabalho foram gerados com coeficientes aleatórios de tal modo que garantam a existência de solução ótima limitada. Os coeficientes da matriz das restrições foram gerados segundo uma distribuição uniforme entre -100 e +100.

No desenvolvimento teórico do algoritmo, nenhum método para detectar a não-viabilidade e a não-limitação da solução ótima foi elaborado.

1.6. METODOLOGIA DO TRABALHO

O presente trabalho de pesquisa originou-se com a verificação de que uma seqüência de projeções ortogonais sobre hiperplanos de restrições violadas, em um problema de programação linear de duas dimensões, apresentavam tendência de convergir para algum ponto da fronteira do conjunto de suas soluções viáveis.

Então, o autor deste trabalho elaborou um programa computacional prévio que permitisse estudar melhor o comportamento deste tipo de convergência, assim como verificar se este comportamento se mantinha para problemas em espaços de dimensões maiores.

Como, em muitos casos, a sequência de projeções era infinita, o autor pesquisou uma forma de determinar diretamente o ponto-limite desta sequência de projeções. No caso em que havia tendência de estabelecer-se uma sequência de projeções entre dois hiperplanos apenas, seu ponto-limite pôde ser determinado com o uso de uma equação algébrica. Foram feitas tentativas de generalizar esta equação para um caso geral, porém sem resultado positivo.

A intenção de apresentar um exemplo de problema de programação linear com duas variáveis, conforme descrito na seção 3.2., é a de aplicar os passos do algoritmo simplex conforme a seção 3.6., e justificar a utilização de lemas e teoremas de projeções ortogonais no algoritmo apresentado pelo autor.

Após este trabalho preliminar, onde prevaleciam a intuição e os resultados estatísticos, como diretrizes da pesquisa, o autor preocupou-se em fundamentar o que parecia constituir leis matemáticas de relevância, originando desta forma o capítulo quatro.

Restava ainda contornar o problema de obtenção da solução ótima. O conhecimento de certas propriedades da teoria da

dualidade permitiram estruturar o problema matemático de tal forma que, se uma solução viável a ele fosse encontrada, obrigatoriamente forneceria o ótimo limitado do problema original. Assim, surgiu o capítulo cinco.

Colecionando os conhecimentos adquiridos por estes dois capítulos, foi possível ao autor elaborar um novo programa computacional, o qual é apresentado neste trabalho.

Novos testes com problemas de coeficientes aleatórios foram testados, e as observações de seus resultados geraram o capítulo final.

CAPÍTULO II

2. HISTÓRICO DA PROGRAMAÇÃO LINEAR

2.1. SURGIMENTO

Durante a 2^a Guerra Mundial a United States Air Force elaborou um grupo de pesquisadores, sob a direção de Marshall K. Wood, para tentar solucionar o problema de alocação de recursos limitados de modo a maximizar ou minimizar algum objetivo quantitativamente expresso. George B. Dantzig era um dos membros deste grupo, e embora nenhum método de expressivo sucesso fosse descoberto durante a guerra, ele formulou o problema de programação linear geral e inventou o método simplex de soluções em 1947. Contudo, o seu trabalho só foi amplamente avaliado em 1951, quando a Cowles Commission Monograph n°13 o publicou¹.

2.2. ESFORÇOS E RESULTADOS INICIAIS

Métodos práticos de solução para o problema de PL, baseados no método simplex foram desenvolvidos entre 1948 e 1952. Durante estes anos, algumas atenções foram feitas para com a utilização dos primitivos equipamentos computacionais, limitados em

¹ T. C. Koopmans, Ed., "Activity Analysis of Production and Allocation", New York, Wiley, (1951).

alcance e sucesso. Para que se tenha uma idéia da situação em 1952, era duvidoso se uma matriz geral de ordem 100 podia ser invertida ou não. Hoje em dia, um problema de 100 variáveis e/ou 100 restrições é considerado pequeno.

Os quatro maiores obstáculos para resolver problemas de PL foram enfrentados em 1952 e durante seus dois ou três anos posteriores. Foram eles:

- a. Achar uma solução básica viável inicial, ponto de partida do algoritmo.
- b. Resolver o problema de guardar a situação de degeneração.²
- c. Redução da área de memória e do número de operações aritméticas requeridas sem causar limitações de uso.
- d. Manter precisão numérica suficiente para a obtenção de resultados significativos.

Foi prontamente reconhecido que achar uma solução viável básica inicial era o mesmo problema que achar uma solução ótima partindo daquela. Problemas de degeneração e semi-degeneração foram amplamente discutidos em vários trabalhos publicados na época, e as dificuldades computacionais associadas foram resolvidas. Com relação à redução de operações aritméticas e espaço de armazenagem, a forma produto da inversa foi desenvolvida em 1953 e bem aceita, exceto em algumas circunstâncias muito especiais co-

² casos de troca de base onde periodicamente a mesma se repete sem que se tenha atingido a solução ótima.

mo a forma standard. O método simplex revisado³, desenvolvido por Dantzig e outros para a RAND Corporation foi publicado neste ano⁴. Em 1954, o uso da dupla precisão numérica veio a resolver o último dos obstáculos mencionados, se bem que o uso da dupla precisão numérica aumente o tempo de operação aritmética, e muita controvérsia sobre precisão numérica versus tempo computacional empregado tenha surgido.

2.3. AMPLIAÇÃO DOS HORIZONTES

A parametrização do lado da mão direita do problema do PL, bem como a dos coeficientes da função objetiva são extensões importantes do problema, e estas técnicas foram inicialmente usadas em 1954 e 1955 e de amplo uso em 1957. Os algoritmos de parametrização foram desenvolvidos simultaneamente.

Em 1957, outros desenvolvimentos ocorreram. O aperfeiçoamento das técnicas de inversão de matrizes, a manobra de variáveis implicitamente limitadas e a introdução do algoritmo dual, cuja teoria já havia sido lançada em 1953⁵. Nenhum desses desenvolvimentos foram de grande uso durante muitos anos; entretanto

³ trata-se de um procedimento computacional eficiente para resolver problemas de PL do mesmo modo como o método simplex, porém evitando computações desnecessárias que o último efetua.

⁴ G. B. DANTZIG, 'Computational Algorithm of the Revised Simplex Method', RM-1266, The RAND Corp., Oct., 1953.

⁵ G. B. DANTZIG and A. ORDEN, 'Duality Theorems', RM-1265, The RAND Corp., Oct. 1953.

eles são agora características padrões, em versões melhoradas, de sistemas de programas computacionais para PL.

Desde 1957 todos os aspectos da PL foram desenvolvidos em um passo veloz. Técnicas de decomposição foram investigadas desde 1953, porém tais técnicas eram na época impraticáveis ou ineficientes. Porém, com a publicação, em 1959, do algoritmo Dantzig-Wolfe⁶, o interesse nestas técnicas cresceu muito⁷. Por volta de 1963, os desenvolvimentos se voltaram para os programas computacionais e considerações sistêmicas, especialmente em técnicas algébricas. Grandes passos também foram dados em discernir novas aplicações para a PL.

2.4. TENDÊNCIAS NÃO ORTODOXAS

Embora hajam publicações de métodos e algoritmos competitivos com o simplex, o sucesso dos mesmos ou é restrito a classes particulares de problemas, ou não-melhor que o simplex, ou objeto ainda de estudos. Citaremos como exemplos o algoritmo Hungariano⁸

⁶ G. B. DANTZIG and P. WOLFE, 'The Decomposition of Mathematical Programming Problems', P-1818, The RAND Corp., May, 1959.

⁷ Muito úteis quando aplicados a matrizes esparsas, quando o número de elementos não nulos é da ordem de 5% ou menos, que tendem a se acumular em submatrizes, e no caso em que a área de armazenagem de dados, tempo de operações aritméticas ou precisão exigida tornam-se críticos para o sucesso da computação.

⁸ KUHN, H. W., 'The Hungarian Method for the Assignment Problem', Naval Research Logistics Quarterly, 2, 83-97, (1955).

e o algoritmo "Out-of-Kilter"⁹ que foram propostos, porém generalizações destes procedimentos foram abandonados. Outra tentativa foi a de produzir um algoritmo semelhante ao simplex, porém com troca de duas variáveis básicas a cada iteração¹⁰, a fim de reduzir o número de iterações até a solução ótima.

L. G. Khachiyan publicou¹¹ em 1979 um algoritmo polinomial limitado para encontrar um ponto interior do conjunto de soluções viáveis do conjunto de restrições de um problema de PL desde que os coeficientes das restrições e do lado da mão direita sejam números inteiros. Se, após certo número de iterações previamente calculado não for encontrado um ponto que satisfaz às desigualdades estritas do problema, conclui-se que não existe o ponto procurado. A limitação imposta por Kachiyan ao modelo de coeficientes inteiros, para efeitos práticos não constitui problema, visto que normalmente trabalhamos com aritmética finita tanto em número de dígitos à esquerda como à direita do ponto decimal. Assim, se multiplicarmos todos os coeficientes de um sistema por um fator adequado, transformamos todos os valores em inteiros. Apesar de não haver inversão de matrizes no algoritmo, surgem dois problemas computacionais sérios para sua implementação:

⁹ FULKERSON, D.R., 'An Out-of-Kilter Method for Minimal-Cost Flow Problems', Journal of the Society of Industrial and Applied Mathematics, 9 (1), 18-27, (1961).

¹⁰ PARANJAPÉ, S. R., 'The Simplex Method: Two Basic Variables Replacement', Management Science, 12, 135-141, (1965).

¹¹ KHACHIYAN, L.G., 'A Polynomial algorithm in linear programming', Doklady Akademiia Nauk SSSR Novaia Seriia, 244 (5), 1093-1096, (1979).

- a. É requerida uma precisão de cálculos muito elevada para que a validade da teoria e do algoritmo se mantenham.
- b. A seqüência de matrizes recursivamente definida é composta de números excessivamente elevados.

Ambas as dificuldades aumentam com o porte do problema e o valor absoluto dos coeficientes. Naturalmente, a teoria deve ser adaptada e o algoritmo modificado para fazer desaparecer os erros de arredondamento e ser computado dentro dos limites razoáveis devido à restrição de armazenagem por número. Mesmo que estes propósitos sejam atingidos, fica ainda a dúvida de como obter a solução ótima do problema sabendo-se que a mesma não é ponto interior do conjunto de soluções viáveis.

Outro método recente para resolver o clássico problema de programação linear é o denominado PROJECT, que combina duas idéias: a de múltipla troca básica, e a pesquisa do ótimo na direção da projeção do vetor gradiente da função objetiva sobre um hiperplano ou o subespaço da intersecção de vários hiperplanos (referências 13. e 14. na bibliografia).

CAPÍTULO III

3. PROGRAMAÇÃO LINEAR E MÉTODO SIMPLEX

3.1. O PROBLEMA DA PROGRAMAÇÃO LINEAR

Alocar recursos de maneira conveniente dentro de certas limitações encontradas é uma situação com a qual nos confrontamos frequentemente. Estas limitações sempre surgem e são devidas à finitude de elementos mensuráveis envolvidos ao nosso dispor, como dinheiro, espaço físico, energia, etc.

Esta alocação poderia ser feita, e muitas vezes o é, sem grandes dificuldades por uma pessoa habituada pela prática e bem envolvida com o problema. Neste caso, o bom senso, na maioria das vezes, tem servido como decisivo na escolha das alocações, bastando posteriormente comprovar sua fidelidade às restrições existentes. Caso algumas restrições ainda não sejam verificadas, novamente o bom senso, aliado à lógica, pode reformular a alocação de modo a adaptá-la a estas restrições sem destruir a viabilidade já alcançada com as demais. Este método pode ser, até certo ponto, funcional, mas não é científico pois não garante conduzir a uma solução, se houver.

Além de tudo, somente a alocação conveniente dos recursos às restrições geralmente não é um objetivo por si só, pois se a mesma pode ser efetuada de múltiplas maneiras, parece conve-

niente estabelecer-se uma ordem de preferência entre elas e almejar-se alcançar a melhor delas. Esta ordem de preferência pode ser justificada por custos, trabalho requerido, lucro, outras razões ou uma combinação entre elas. Admitimos, por isso, que esta preferência possa ser quantitativamente expressa por uma função, a qual denominaremos de função objetiva.

Assim, torna-se por demais exaustivo a utilização do método do bom senso aliado à seleção da alocação mais conveniente, sobretudo se o número de soluções viáveis é muito grande ou infinito, justificando a criação de um tratamento matemático aplicado a este tipo de problema.

Levaremos em conta, de agora em diante, que as variáveis envolvidas possam assumir valores contínuos e a linearidade das restrições e da função objetiva, para que o problema possa ser chamado de programação linear. Devido a ordem de preferência quantitativamente expressa pela função objetiva, sempre desejaremos maximizar ou minimizar a mesma.

Durante a formulação do modelo matemático de uma situação real, devemos em primeiro lugar estarmos certos de que o modelo retrate o mais fielmente possível a realidade, evitando esquecimento de informações e interpretações errôneas.

3.2. EXEMPLO COM DUAS VARIÁVEIS

No problema que segue, vamos admitir que todas as informações necessárias para a formulação do modelo matemático estão descritas.

"Um agricultor deseja cultivar duas variedades de cereais, digamos tipo A e B, em uma área restrita a um hectare, sendo que cada are cultivado pelo cereal A produz 8 sacas, e por B 10 sacas. Para o plantio, cada are cultivado de cereais A precisa de 3 homens-hora, e o cultivado de cereais B precisa de 2 homens-hora, sendo que 240 é a disposição máxima de homens-hora para o cultivo, a um custo de Cr\$ 200,00 por homem-hora. A demanda máxima é limitada pelo mercado consumidor a 480 sacas do cereal A, a um preço de Cr\$ 150,00 por saca, e a 800 sacas do cereal B, a um preço de Cr\$ 120,00 por saca. O agricultor deseja planejar a sua produção de modo a maximizar o lucro obtido."

As variáveis decisórias podem ser consideradas de duas maneiras: em termos de quantos ares serão cultivados por cada tipo de cereal, ou então em quantas sacas de cada tipo deverão ser recolhidas. Por conveniência, vamos considerar o modelo matemático construído a partir da primeira. Assim, sejam

x_1 = quantidade de ares cultivado para o cereal A;

x_2 = quantidade de ares cultivado para o cereal B.

Por ordem, as restrições são:

$$x_1 + x_2 \leq 100$$

$$3x_1 + 2x_2 \leq 240$$

$$x_1 \leq 480/8$$

$$x_2 \leq 800/10$$

$$x_1, x_2 \geq 0$$

e a função objetiva deverá incorporar as receitas e os custos variáveis. Será denominada lucro e denotada por Z. Assim,

$Z = \text{receitas} - \text{custos variáveis}$

$$Z = (150 \times 8)x_1 + (120 \times 10)x_2 - ((3 \times 200)x_1 + (2 \times 200)x_2)$$

$$Z = 600x_1 + 700x_2$$

e o nosso modelo matemático será

$$\text{Max } Z = 600x_1 + 700x_2$$

sujeito a

$$x_1 + x_2 \leq 100$$

$$3x_1 + 2x_2 \leq 240$$

$$x_1 \leq 60$$

$$x_2 \leq 80$$

$$x_1, x_2 \geq 0$$

É importante observar que todo problema de programação linear pode ser representado matricialmente. Neste caso, façamos

$$A = \begin{bmatrix} 1 & 1 \\ 3 & 2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 100 \\ 240 \\ 60 \\ 80 \end{bmatrix}, \quad c = \begin{bmatrix} 600 \\ 700 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

e então o modelo pode ser descrito como

$$\text{Max } Z = c^T x$$

sujeito a

$$A x \leq b$$

$$x \geq 0$$

3.3. RESOLUÇÃO GEOMÉTRICA

É possível resolver problemas de programação linear de duas variáveis geometricamente desde que a configuração geométrica do modelo matemático não vá além das limitações naturais da exatidão dos nossos instrumentos de desenho. A configuração geométrica do modelo descrito na seção anterior será:

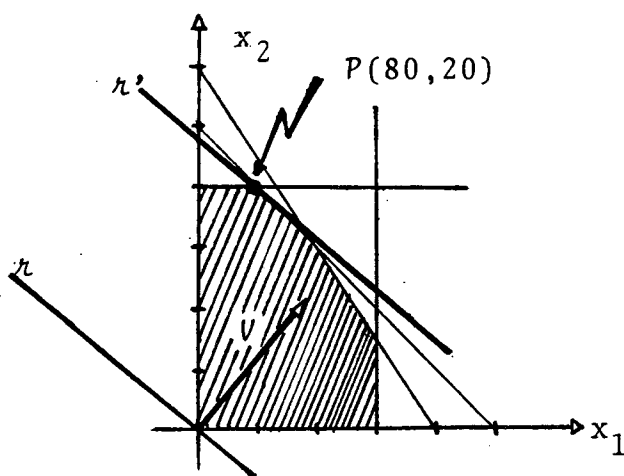


Fig. 1 Gráfico de um problema de PL

v é um vetor que indica a direção do máximo crescimento da função objetiva, cujas projeções horizontal e vertical são magnitudes proporcionais aos coeficientes da função objetiva, 600 e 800, respectivamente. A região hachuriada representa o conjunto de soluções viáveis do problema. A reta n perpendicular a v , é o lugar geométrico de todos os pontos em que $Z = 0$, e qualquer paralela a n é o lugar geométrico de pontos de Z constante. Deste modo, a reta n' é a paralela a n mais distante desta que contém algum ponto do conjunto de soluções compatíveis, o vértice P . Logo, P será o ponto de ótimo com

$$x_1^* = 20$$

$$x_2^* = 80$$

$$Z^* = 68000$$

ou seja, se o agricultor tomar a decisão ótima de plantar 20 ares de terra com cereais A e 80 ares de terra com cereais B, obterá um lucro de Cr\$ 68.000,00. Note-se ainda que

$$3 x_1^* + 2 x_2^* = 220 < 240$$

o que significa que, dos 240 homens-hora disponíveis, necessitaremos apenas 220. A diferença de 20 homens é denominada de folga na restrição.

3.4. O MÉTODO SIMPLEX DE RESOLUÇÃO

O método simplex faz uso de importantes propriedades algébricas aliadas à estrutura dos problemas de programação linear, como:

- a. O conjunto de soluções compatíveis é convexo e fechado.
- b. Se existe solução limitada ótima, ela será ponto fronteira do conjunto de soluções compatíveis;
- c. Se existe solução limitada ótima, haverá pelo menos um vértice do conjunto de soluções viáveis que a forneça.

Todo problema de programação linear envolvendo desigualdades, pode ser transformado em um problema equivalente em que

apareçam apenas igualdades nas restrições desde que adicionemos ou subtraímos variáveis de folga a cada uma delas. Assim, o problema da seção 3.2. poderia ser escrito como

$$\text{Max } Z = 600 x_1 + 700 x_2 + 0 x_3 + 0 x_4 + 0 x_5 + 0 x_6$$

sujeito a

$$x_1 + x_2 + x_3 = 100$$

$$3x_1 + 2x_2 + x_4 = 240$$

$$x_1 + x_5 = 60$$

$$x_2 + x_6 = 80$$

$$x_1, \dots, x_6 \geq 0$$

onde x_3, \dots, x_6 são as variáveis de folga incorporadas ao modelo.

Podemos representar matricialmente o problema com as variáveis de folga incorporadas na forma

$$(A, I) \cdot \begin{bmatrix} x \\ x_S \end{bmatrix} = b$$

ou ainda $Ax + Ix_S = b$,

e se considerarmos a i -ésima coluna de A como A_i , e a j -ésima coluna de I como e_j , teremos

$$\sum_{i=1}^n A_i x_i + \sum_{j=1}^m e_j x_{Sj} = b.$$

Como encontramos um máximo de m colunas independen-

tes em (A, I) , pois as colunas de I constituem uma base canônica do problema, qualquer matriz quadrada B formada por m colunas linearmente independentes de (A, I) constituem uma matriz básica, pois suas colunas geram o espaço R^m . Assim, B é não-singular.

Chamando de N a matriz formada pelas demais colunas de (A, I) , não em B , poderemos reescrever o sistema como

$$B x_B + N x_N = b$$

onde x_B , x_N são vetores de variáveis correspondentes às colunas em B e N , respectivamente. As variáveis de x_B são chamadas de básicas, e as de x_N de não-básicas. Toda solução ao sistema acima em que $x_N = 0$ e $x_B = B^{-1}.b$ é chamada de solução básica. Vale portanto a propriedade

- d. A cada solução básica viável corresponde um vértice do conjunto de soluções viáveis do problema.

Assim, na circunstância de haver uma solução ótima limitada, haverá também uma solução básica ótima. Portanto, é natural que se pesquisem as soluções básicas na procura do ótimo, pois são em número finito. O método que assim procede é o da enumeração exaustiva.

Se o problema tiver m restrições estruturais e n variáveis, então após a incorporação das variáveis de folga, uma por restrição estrutural, teremos um total de $m+n$ variáveis, e assim o método da enumeração exaustiva irá pesquisar um número total de

$$\binom{m+n}{m} = \frac{(m+n)!}{m! n!}$$

soluções básicas para selecionar a melhor viável.

O método simplex evita a exploração exaustiva de soluções básicas tomando como idéia:

- a. Só pesquisar soluções viáveis básicas;
- b. Melhorar a cada etapa o valor da função objetiva;
- c. Utilizar uma regra de parada caso a solução ótima tenha sido alcançada, ou em outras circunstâncias como solução ótima ilimitada ou não-existência de solução viável.

A aplicação do método simplex na sua forma originalmente concebida exige que o problema esteja na forma 'standard', ou seja:

- a. Com suas variáveis de folga e/ou artificiais;
- b. Restrição de não-negatividade de todas variáveis;
- c. Valores da mão-direita nas restrições não-negativos.

Com o problema assim colocado, o algoritmo procede nos seguintes passos:

Passo 1: partir de uma solução básica viável inicial.

Passo 2: verificar se a solução atual é ótima. Se é, pare. Senão, vá ao passo 3.

Passo 3: determinar a variável não-básica que deve entrar na base atual.

Passo 4: determinar qual variável básica deve sair da base atual.

Passo 5: obter a nova solução viável básica e retornar ao passo 2.

3.5. TEORIA DO MÉTODO SIMPLEX

De acordo com a notação estabelecida na seção precedente, seja B uma base de um problema de PL. Assim, a solução básica será

$$x_B = B^{-1}b.$$

Como a função objetiva pode ser escrita como

$$Z = c_B^T x_B + c_N^T x_N,$$

e tendo em vista que $x_N = 0$ na solução básica, concluímos que seu valor presente é

$$Z = c_B^T x_B.$$

Queremos introduzir na base uma coluna A_j de N , trocando-a por uma coluna B_r em B , de modo a produzir uma nova base \hat{B} , a qual produz outra solução básica $\hat{x}_B = \hat{B}^{-1}b$ também viável. Devemos estabelecer, a princípio, qual é a coluna B_r que deve sair sem violar a viabilidade da nova solução. Como as colunas de \hat{B} geram o espaço \mathbb{R}^m , existirá um vetor y_j tal que

$$A_j = B y_j$$

e assim

$$y_j = B^{-1}A_j.$$

Vamos definir o número z_j como

$$z_j = c_B^T y_j = c_B^T B^{-1}A_j.$$

Se A_j deve substituir um vetor B_r em B , é necessário

que $y_{rj} \neq 0$ e assim o vetor a sair da base pode ser escrito como

$$B_r = \frac{1}{y_{rj}} A_j - \sum_{\substack{i=1 \\ i \neq j}}^m \frac{y_{ij}}{y_{rj}} B_i$$

e como

$$\sum_{i=1}^m x_{Bi} B_i = b$$

temos, pela substituição na base,

$$\sum_{\substack{i=1 \\ i \neq r}}^m (x_{Bi} - x_{Br} \frac{y_{ij}}{y_{rj}}) B_i + \frac{x_{Br}}{y_{rj}} A_j = b .$$

A nova solução deve ser viável. Isto requer que

$$(*) \quad x_{Bi} - x_{Br} \frac{y_{ij}}{y_{rj}} \geq 0 \text{ para } i \neq r; \text{ e}$$

$$(**) \quad \frac{x_{Br}}{y_{rj}} \geq 0 .$$

Se $x_{Br} = 0$, a nova solução será a mesma que a anterior. Se x_{Br} não é nulo, então $y_{rj} > 0$ para satisfazer (**), e assim (*) é satisfeito sempre que $y_{ij} \leq 0$. Se $y_{ij} > 0$, teremos que ter

$$\frac{x_{Bi}}{y_{ij}} - \frac{x_{Br}}{y_{rj}} \geq 0 \text{ para } i \neq r; \text{ e assim}$$

$$\frac{x_{Br}}{y_{rj}} \leq \frac{x_{Bi}}{y_{ij}} \text{ para } i \neq r;$$

e então a escolha da coluna que sai da base deve recair sobre o vetor B_r em que

$$\frac{x_{Br}}{y_{rj}} = \min_i \left\{ \frac{x_{Bi}}{y_{ij}} , y_{ij} > 0 \right\} = \theta$$

fornecendo a nova solução em que

$$\hat{x}_{Bi} = x_{Bi} - x_{Br} \frac{y_{ij}}{y_{rj}}, \text{ para } i \neq r, \text{ e}$$

$$\hat{x}_{Br} = \frac{x_{Br}}{y_{rj}}.$$

A substituição da nova solução na função objetiva nos dará

$$\hat{Z} = Z - \theta(z_j - c_j).$$

Se $\theta > 0$, obteremos uma solução em que $\hat{Z} > Z$ quando $z_j - c_j < 0$. Portanto, quando o problema é de maximização, o critério de escolha da coluna que entra na base recai sobre alguma em que $z_j - c_j < 0$. Usualmente, escolhe-se a coluna de mínimo $z_j - c_j < 0$ ^{1,2}.

Quando os $z_j - c_j \geq 0$, para todas as variáveis, teremos atingido a solução ótima limitada, pois qualquer troca na base não aumenta o valor da função objetiva, no caso específico de maximização.

Se ocorrer $\theta = 0$, estaremos diante do problema de degeneração, e como evitá-lo foge dos interesses do presente trabalho.

Se existirem $z_j - c_j < 0$, todos com $y_{ij} \leq 0$, então teremos uma solução ótima ilimitada.

^{1,2} Esta regra é puramente heurística. Não garante, em absoluto, que a sua adoção reduza o número de soluções básicas pesquisadas pelo algoritmo.

3.6. EXEMPLO DE APLICAÇÃO DO ALGORITMO

A presente seção vai mostrar uma aplicação dos passos do algoritmo simplex, segundo a teoria exposta na seção anterior ao problema exemplo. Como o problema já se encontra na forma canônica padronizada, encontramos facilmente uma solução básica inicial.

Assim, a base inicial será a matriz identidade I , e portanto

$$x_B = B^{-1}b = I^{-1}b = Ib = b$$

$$x_B = (x_{S1}, x_{S2}, x_{S3}, x_{S4}) = (100, 240, 60, 80)$$

$$Z = c_B^T x_B = (0, 0, 0, 0) \cdot (100, 240, 60, 80)^T$$

$$Z = 0$$

$$y_j = B^{-1}A_j \text{ para } j = 1, 2$$

$$z_j = c_B^T B^{-1}A_j = 0 \text{ para } j = 1, 2$$

e portanto,

$$z_1 - c_1 = 0 - 600 = -600$$

$$z_2 - c_2 = 0 - 700 = -700.$$

Como $z_j - c_j < 0$, para $j = 1, 2$, escolhemos x_1 ou x_2 como nova variável básica. Seja $j=2$. Então

$$y_2 = A_2 = (1 , 2 , 0 , 1)^T$$

$$\min_i \left\{ \frac{x_{Bi}}{y_{ij}} , y_{ij} > 0 \right\} = \min \left\{ \frac{x_{B1}}{y_{12}} , \frac{x_{B2}}{y_{22}} , \frac{x_{B3}}{y_{42}} \right\}$$

$$= \min \{ 100/1 , 240/2 , 80/1 \} = 80 = x_{B4}/y_{42}$$

e assim $r = 4$, ou seja, a 4ª coluna da base é removida e trocada, entrando A_2 em seu lugar, constituindo a nova base \hat{B} .

A nova solução será

$$\hat{x}_B = (\hat{x}_{S1} , \hat{x}_{S2} , \hat{x}_{S3} , \hat{x}_2)$$

onde

$$\hat{x}_{S1} = 100 - 80 \times 1 / 1 = 20$$

$$\hat{x}_{S2} = 240 - 80 \times 2 / 1 = 80$$

$$\hat{x}_{S3} = 60 - 80 \times 0 / 1 = 60$$

$$\hat{x}_2 = 80 / 1 = 80$$

$$\hat{Z} = 0 - 80 \times (-700) = 56000.$$

Retorna-se ao cálculo de $y_j - c_j$ para as variáveis não básicas $\hat{x}_1 = \hat{x}_{S4} = 0$, para testar a otimalidade da presente solução. O algoritmo prossegue até que se encontre a solução ótima conforme a resolução geométrica na seção 3.3. . Em termos práticos, estas operações são efetuadas sobre um 'tableaux' adequado.

CAPÍTULO IV

4. TEOREMAS SOBRE PROJEÇÕES EM HIPERPLANOS

4.1. TERMINOLOGIA USADA

Durante o transcorrer deste trabalho, iremos referir-nos sempre ao clássico problema de PL geral:

$$(1) \quad \begin{aligned} \text{Max } Z &= c^T x \\ Ax &\leq b \\ x &\geq 0 \end{aligned}$$

onde:

- . A é uma matriz de números reais $m \times n$ sem linha ou coluna de zeros;
- . x é um vetor de dimensão n;
- . b é um vetor de dimensão m;
- . c é um vetor de dimensão n.

Todo problema de PL pode ser modelado de modo a apresentar-se na forma (1).

Dado um vetor v qualquer, sua norma euclidiana será denotada por $|v|$ e calculada como

$$|v| = (v^T \cdot v)^{1/2}$$

A matriz A de (1) pode também ser representada como:

$$A = (a_{ij})_{mn} = \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_m \end{bmatrix}$$

onde a_i ($1 \leq i \leq m$) é a matriz linha dos coeficientes da i -ésima linha da matriz A . Como $a_i x = b_i$ define um hiperplano no espaço \mathbb{R}^n , o mesmo será denotado por H_i . Assim,

$$H_i = \{x \in \mathbb{R}^n / a_i x = b_i\} .$$

Representaremos por $a_i x : b_i$ uma relação onde $:$ poderá ser \leq ou \geq . Esta relação define um semi-espaço em \mathbb{R}^n denotado por R_i .

$$\bar{R}_i = \{x \in \mathbb{R}^n / a_i x : b_i\} .$$

Dado um conjunto qualquer C em \mathbb{R}^n , denotaremos por \bar{C} o seu complementar.

4.2. DEFINIÇÕES, LEMAS E TEOREMAS

Definição 1: Sejam $x_1, x_2 \in \mathbb{R}^n$ e $x_1 \notin H_i$. Diremos que x_2 é a projeção ortogonal de x_1 sobre H_i , e denotaremos por $x_2 = p(x_1, H_i)$ se, e somente se:

- (a) $x_2 \in H_i$
- (b) $(x_2 - x_1)$ é ortogonal a H_i , isto é, perpendicular a qualquer vetor contido em H_i .

Lema 1: Seja $x_1 \in \mathbb{R}^n$ e $x_1 \notin H_i$. Se calcularmos x_2 como segue:

$$(2) \quad x_2 = x_1 + \frac{(b_i - a_i x_1)}{|a_i|^2} a_i$$

então

$$x_2 = p(x_1, H_i)$$

Prova :

$$\begin{aligned} (a) \quad a_i \cdot x_2 &= a_i \cdot \left(x_1 + \frac{(b_i - a_i x_1)}{|a_i|^2} a_i \right) \\ &= a_i x_1 + \frac{(b_i - a_i x_1)}{|a_i|^2} a_i^2 \\ &= a_i x_1 + b_i - a_i x_1 = b_i \end{aligned}$$

e logo, $x_2 \in H_i$

$$(b) \quad x_2 - x_1 = \frac{(b_i - a_i x_1)}{|a_i|^2} a_i \quad \text{onde} \quad \frac{b_i - a_i x_1}{|a_i|^2} \neq 0 \quad \text{e}$$

logo $(x_2 - x_1)$ possui a mesma direção de a_i , perpendicular a H_i .

Lema 2: Sejam $x_1, x_2, y \in \mathbb{R}^n$ tais que:

a. $x_1 \in \bar{R}_i$

b. $x_2 = p(x_1, H_i)$

c. $y \in R_i$

então $|y - x_2| < |y - x_1|$ se $x_2 \neq y$

Prova : Mostraremos a validade deste lema através da relação $a_i x \leq b_i$ apenas. Para a relação $a_i x \geq b_i$, a prova é semelhante. Assim admitiremos que:

$$a. \quad a_i x_1 > b_i$$

$$b. \quad x_2 = x_1 + \frac{(b_i - a_i x_1)}{|a_i|^2} a_i$$

$$c. \quad a_i y \leq b_i \quad e \quad y \neq x_2$$

Sabemos que:

$$(x_1 - y) + (x_2 - x_1) = x_2 - y$$

logo, segue que

$$x_1 - y + \frac{(b_i - a_i x_1)}{|a_i|^2} a_i = x_2 - y$$

$$(x_1 - y) (x_2 - y) + \frac{(b_i - a_i x_1)}{|a_i|^2} a_i (x_2 - y) = (x_2 - y)^2$$

e além disso,

$$b_i - a_i x_1 < 0 \quad e \quad a_i (x_2 - y) = a_i x_2 - a_i y = b_i - a_i y \geq 0$$

logo

$$\frac{b_i - a_i x_1}{|a_i|^2} a_i (x_2 - y) \leq 0$$

Consideremos, agora, dois casos:

$$i) \quad a_i y = b_i. \quad \text{Então} \quad \frac{(b_i - a_i x_1)}{|a_i|^2} a_i (x_2 - y) = 0 \quad e$$

$$(x_1 - y) (x_2 - y) = (x_2 - y)^2$$

Como $y \neq x_2$, temos:

$$|x_1 - y| |x_2 - y| > (x_1 - y) (x_2 - y)$$

e, portanto

$$|x_1 - y| |x_2 - y| > (x_2 - y)^2 = |x_2 - y|^2$$

$$\text{e logo } |x_1 - y| > |x_2 - y|$$

ii) $a_i y < b_i$. Então $\frac{(b_i - a_i x_1)}{|a_i|^2} a_i (x_2 - y) < 0$ e

$$(x_1 - y) (x_2 - y) > (x_2 - y)^2$$

$$\text{Mas } |x_1 - y| |x_2 - y| \geq (x_1 - y) (x_2 - y)$$

e, portanto

$$|x_1 - y| |x_2 - y| > (x_2 - y)^2 = |x_2 - y|^2$$

$$\text{e logo } |x_1 - y| > |x_2 - y|.$$

Definição 3: Denotaremos por S o conjunto dos pontos viáveis do problema (1), ou seja:

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

Definição 4: Seja $y \in \mathbb{R}^n$. Representaremos por $d(y, S)$ a distância do ponto y ao conjunto S definido como

$$d(y, S) = \begin{cases} \min \{|y - x| \mid x \in S\} & \text{se } S \neq \{\} \\ \infty & \text{se } S = \{\} \end{cases}$$

Lema 3 : Seja $S \neq \{\}$, seja $x_1 \in \mathbb{R}^n$ e $x_1 \notin R_i$. Logo, se $x_2 = p(x_1, H_i)$, então $d(x_2, S) < d(x_1, S)$.

Prova: O conjunto convexo das soluções compatíveis de (1) pode ser escrito como:

$$S = \bigcap_{j=1}^m R_j$$

Seja $y \in S$. Logo $y \in R_i$ e como $x_1 \in \bar{R}_i$ segue, de acordo com o lema 2, que

$$|y - x_2| < |y - x_1|$$

e portanto,

$$\min \{|y - x_2| / y \in S\} < \min \{|y - x_1| / y \in S\}$$

pois S é fechado, e, de acordo com a definição 4, temos que

$$d(x_2, S) < d(x_1, S)$$

Os lemas que seguem até o final deste capítulo, referem-se à seguinte situação: seja $x_1 \in H_i$, e H_i e H_j dois hiperplanos tais que $a_i a_j \neq 0$ e $|a_i a_j| \neq |a_i| |a_j|$, e seja (x_1, x_2, x_3, \dots) uma seqüência de pontos projetados a partir do segundo, alternadamente, entre os hiperplanos H_j e H_i de modo que $x_1 \in H_i$, $x_2 = p(x_1, H_j)$, $x_3 = p(x_2, H_i)$, $x_4 = p(x_3, H_j)$, ...

Lema 4: A seqüência de distâncias euclidianas

$$(|x_{k+1} - x_k|, k = 1, 2, 3, \dots)$$

forma uma progressão geométrica decrescente cuja razão é o co-seno do ângulo agudo formado por a_i e a_j .

Prova : Sem perda de generalidade, podemos considerar:

$$|a_i| = |a_j| = 1^{13}.$$

Assim, através da expressão (2),

$$x_2 = x_1 + (b_j - a_j x_1) a_j$$

$$|x_2 - x_1| = |b_j - a_j x_1| |a_j|$$

$$|x_2 - x_1| = |b_j - a_j x_1|$$

e também

$$x_3 = x_2 + (b_i - a_i x_2) a_i$$

$$|x_3 - x_2| = |b_i - a_i x_2| \cdot |a_i|$$

substituindo x_2 na última expressão teremos:

$$|x_3 - x_2| = |b_i - a_i (x_1 + (b_j - a_j x_1) a_j)|$$

$$|x_3 - x_2| = |b_i - a_i x_1 - (b_j - a_j x_1) a_i a_j|$$

e como $x_1 \in H_i$, temos $b_i - a_i x_1 = 0$ e assim

$$|x_3 - x_2| = |b_j - a_j x_1| \cdot |a_i a_j|$$

$$|x_3 - x_2| = |x_2 - x_1| \cdot |a_i a_j|$$

temos ainda que

$$|a_i a_j| = ||a_i| \cdot |a_j| \cdot \cos \theta| = |\cos \theta|$$

onde θ é o ângulo entre a_i e a_j e $|\cos \theta|$ é o co-seno do ângulo agudo entre a_i e a_j . Como por hipótese H_i e H_j não

¹³ Se $|a_i| \neq 1$ ou $|a_j| \neq 1$, podemos dividir todos os coeficientes da restrição de norma não-unitária pela norma de seu gradiente a_i ou a_j sem alterar o conjunto-solução S , de modo que seu novo gradiente seja um vetor unitário.

são paralelos e nem perpendiculares,

$$0 < |\cos \theta| < 1$$

e portanto

$$|x_{k+1} - x_k| = |x_k - x_{k-1}| \cdot |\cos \theta| \quad (k=2,3,4,\dots)$$

ou ainda

$$|x_{k+1} - x_k| = |x_2 - x_1| \cdot (|\cos \theta|)^{k-1} \quad (k=2,3,4,\dots)$$

Lema 5: A seqüência de pontos (x_1, x_2, x_3, \dots) converge.

Prova : Sejam m, n inteiros positivos tais que $m > n$.

Então

$$\begin{aligned} |x_m - x_n| &\leq |x_m - x_{m-1}| + |x_{m-1} - x_{m-2}| + \dots + |x_{n+1} - x_n| \\ &< \sum_{i=n}^{\infty} |x_{i+1} - x_i| \\ |x_m - x_n| &< \sum_{i=n}^{\infty} |x_2 - x_1| \cdot (|\cos \theta|)^{i-1} = \\ &= \frac{|x_2 - x_1| \cdot (|\cos \theta|)^{n-1}}{1 - |\cos \theta|} \end{aligned}$$

onde $|\cos \theta|$ é o co-seno do ângulo agudo entre H_i e H_j .

Portanto, $\forall \xi > 0$, seja n tal que

$$\frac{|x_2 - x_1| \cdot (|\cos \theta|)^{n-1}}{1 - |\cos \theta|} \leq \xi$$

e assim

$$|x_m - x_n| < \xi$$

para todo $m > n$ e portanto a seqüência (x_1, x_2, x_3, \dots) é convergente.

Lema 6: Seja $\bar{x} = \lim_{k \rightarrow \infty} x_k$. Então existem $\lambda_i, \lambda_j \in \mathbb{R}$ tais que:

$$\bar{x} = x_1 + \lambda_i a_i + \lambda_j a_j.$$

Prova : Vamos definir uma translação T que leve a origem do sistema de coordenadas para x_1 ,

$$T(x) = x - x_1, \quad \forall x \in \mathbb{R}^n$$

e deste modo:

$$T(x_1) = x_1 - x_1 = 0$$

$$T(x_2) = x_2 - x_1 = \frac{(b_j - a_j x_1)}{|a_j|^2} a_j$$

$$T(x_3) = x_3 - x_1 = (x_3 - x_2) + (x_2 - x_1)$$

$$T(x_3) = \frac{(b_i - a_i x_2)}{|a_i|^2} a_i + \frac{(b_j - a_j x_1)}{|a_j|^2} a_j$$

.....

e assim $(T(x_1), T(x_2), T(x_3), \dots)$ é uma seqüência gerada por a_i e a_j e portanto pertence ao subespaço vetorial gerado por a_i e a_j . Seja V este subespaço,

$$V = \{x \in \mathbb{R}^n / \exists \lambda_i, \lambda_j \in \mathbb{R}, x = \lambda_i a_i + \lambda_j a_j\}.$$

Como V é fechado, significa que V contém todos os seus pontos de acumulação, e, particularmente, contém

$$T(\bar{x}) = \lim_{k \rightarrow \infty} T(x_k)$$

assim, existem $\lambda_i, \lambda_j \in \mathbb{R}$ tais que

$$T(\bar{x}) = \lambda_i a_i + \lambda_j a_j$$

e assim

$$\bar{x} = T^{-1} (\lambda_i a_i + \lambda_j a_j)$$

$$\bar{x} = \lambda_i a_i + \lambda_j a_j + x_1$$

Lema 7: $\bar{x} \in H_i \cap H_j$

Prova : Se \bar{x} não pertence a algum dos hiperplanos, suponhamos H_i , por exemplo, então se $0 < \xi < d(\bar{x}, H_i)$. Teremos sempre que $|x_k - \bar{x}| > \xi$ para $k = 1, 3, 5, 7, \dots$ pois $x_1, x_3, x_5, \dots \in H_i$.

Teorema 1: Se $\bar{x} = \lim_{k \rightarrow \infty} x_k$, então

$$(2) \quad \bar{x} = x_1 + (a_i^2 a_j - (a_i a_j) a_i) \cdot \frac{(b_j - a_j x_1)}{a_i^2 a_j^2 - (a_i a_j)^2}$$

Prova : Vamos escrever

$$\bar{x} = x_1 + \lambda_i a_i + \lambda_j a_j$$

e então, de acordo com o lema 7, $\bar{x} \in H_i \cap H_j$

$$\bar{x} a_i = x_1 a_i + \lambda_i a_i^2 + \lambda_j a_i a_j$$

$$b_i = b_i + \lambda_i a_i^2 + \lambda_j a_i a_j$$

$$\lambda_i a_i^2 + \lambda_j a_i a_j = 0 \quad (*)$$

e também

$$\bar{x} a_j = x_1 a_j + \lambda_i a_i a_j + \lambda_j a_j^2$$

$$b_j = x_1 a_j + \lambda_i a_i a_j + \lambda_j a_j^2$$

$$\lambda_i a_i a_j + \lambda_j a_j^2 = b_j - x_1 a_j \quad (**)$$

e, resolvendo o sistema (*) e (**), obteremos

$$\lambda_i = \frac{a_i a_j (x_1 a_j - b_j)}{a_i^2 a_j^2 - (a_i a_j)^2}$$

$$\lambda_j = \frac{a_i^2 (b_j - x_1 a_j)}{a_i^2 a_j^2 - (a_i a_j)^2}$$

e portanto,

$$\bar{x} = x_1 + \frac{(a_i a_j) a_i (x_1 a_j - b_j)}{a_i^2 a_j^2 - (a_i a_j)^2} + \frac{a_i^2 a_j (b_j - x_1 a_j)}{a_i^2 a_j^2 - (a_i a_j)^2}$$

$$\bar{x} = x_1 + (a_i^2 a_j - (a_i a_j) a_j) \frac{(b_j - x_1 a_j)}{a_i^2 a_j^2 - (a_i a_j)^2}$$

Teorema 2: Se $\bar{x} = \lim_{k \rightarrow \infty} x_k$ e $S \neq \{\}$, então

$$d(\bar{x}, S) < d(x_k, S) \text{ para } k = 1, 2, 3, \dots$$

Prova : Como $(d(x_k, S), k = 1, 2, 3, \dots)$ constitui uma seqüência monotonicamente decrescente e convergente, então $d(\bar{x}, S) < (d(x_k, S))$ para qualquer $k = 1, 2, 3, \dots$

CAPÍTULO V

5. O PROBLEMA EQUIVALENTE PRIMAL-DUAL

5.1. O PROBLEMA PRIMAL E O PROBLEMA DUAL

Como é bem conhecido através da literatura da PL, a cada problema do tipo (1)

$$\begin{aligned} \text{Max } Z &= c^T \cdot x \quad \text{sujeito à} \\ Ax &\leq b \\ x &\geq 0 \end{aligned}$$

podemos associar um outro problema de PL

$$\begin{aligned} (3) \quad \text{Min } W &= b^T \cdot y \quad \text{sujeito à} \\ A^T y &\geq c \\ y &\geq 0 \end{aligned}$$

onde y é um vetor de dimensão m . O problema (3) é chamado de problema dual do problema (1), sendo que este último é também denominado de problema primal¹⁴. A resolução do problema dual apresenta

¹⁴ Esta distinção entre primal e dual não implica em fixar o nome de um problema de PL em primal ou dual, visto que podemos transformar a função objetivo de minimização para maximização, e restrições do tipo 'maior ou igual' em 'menor ou igual'. O mesmo sucede ao contrário. Apenas significa que, se um problema for considerado o primal, o outro será o seu dual. O dual do dual é o primal.

muitas características de interesse relacionadas à resolução do problema primal, algumas das quais passaremos a enunciar a seguir.

- (a) Seja x uma solução viável de (1) e y uma solução viável de (3). Então,

$$c^T x \leq b^T y$$

ou ainda

$$Z \leq W$$

- (b) Se (1) e (3) possuírem soluções ótimas finitas, denotadas por x^* e y^* , então

$$c^T x^* = b^T y^*$$

ou ainda

$$Z^* = W^*$$

e, reciprocamente, se x^* é solução viável de (1) e y^* é solução viável de (3) tais que $c^T x^* = b^T y^*$, então x^* e y^* são soluções ótimas de (1) e (3), respectivamente.

- (c) Apenas uma das três alternativas que seguem é verdadeira:

- . nenhum dos problemas admite soluções viáveis;
- . um problema não admite solução viável e o outro admite soluções viáveis, porém seu ótimo não é finito;
- . os dois problemas admitem soluções ótimas finitas.

A última propriedade é também chamada de propriedade

fundamental da existência.

5.2. O PROBLEMA EQUIVALENTE

Queremos agora possibilitar a resolução do problema (1) sem recorrer às técnicas de tentar otimizar a função-objetiva, mas determinando a solução ótima (quando existir) do problema de PL como sendo uma solução viável de outro problema de PL, recorrendo às propriedades já conhecidas sobre a relação entre o problema primal e o problema dual. Para conseguir este objetivo, vamos juntar ambos os problemas e a propriedade (b) da seção precedente, de modo a criar um problema único, doravante denominado de problema equivalente primal-dual e referenciado por PEPD, como segue abaixo:

$$(4) \quad \begin{bmatrix} A & 0 \\ 0 & A^T \\ c^T & -b^T \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \begin{matrix} \leq \\ \geq \\ = \end{matrix} \begin{bmatrix} b \\ c \\ 0 \end{bmatrix} \quad \text{e} \quad \begin{bmatrix} x \\ y \end{bmatrix} \geq 0 \quad (\text{PEPD})$$

tal que

$\begin{bmatrix} x \\ y \end{bmatrix}$ = vetor de dimensão $(n+m)$ resultante da concatenação¹⁵ dos vetores x e y dos problemas (1) e (3).

$\begin{bmatrix} b \\ c \\ 0 \end{bmatrix}$ = vetor resultante da concatenação de b e c dos problemas (1) e (3) e do vetor 0 em R , de dimensão $(n+m+1)$.

¹⁵ Sejam $v_1^T = (w_1, \dots, w_n)$ e $v_2^T = (w_{n+1}, \dots, w_m)$ dois vetores quaisquer. Então a concatenação de v_1 e v_2 resultará no vetor $(v_1^T, v_2^T)^T = (w_1, \dots, w_n, w_{n+1}, \dots, w_m)$.

$$\begin{bmatrix} A & 0 \\ 0 & A^T \\ c^T & -b^T \end{bmatrix} = \text{matriz de dimensão } (m+n+1) \times (m+n) \text{ cujo particionamento é o da configuração indicada. Suas submatrizes } 0 \text{ são formadas por zeros.}$$

O problema (4) poderia ainda ser representado como:

$$(5) \quad \begin{bmatrix} A & 0 \\ 0 & A^T \\ c^T & -b^T \\ I & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \begin{matrix} \leq \\ \geq \\ = \\ \geq \\ \geq \end{matrix} \begin{bmatrix} b \\ c \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{PEPD}).$$

O lema que segue estabelece uma relação entre (1), (3) e (4).

Lema 8: O problema primal (ou dual) possui solução ótima limitada se, e somente se, o seu PEPD possui solução.

Prova: Decorre imediatamente das propriedades (a), (b) e (c) da seção precedente.

Além deste lema, podemos evidenciar mais duas propriedades ligadas ao PEPD, enunciadas nos lemas 9 e 10.

Lema 9: Duas restrições quaisquer do PEPD, sendo uma delas do primal e outra do dual (inclusive restrições de não-negatividade), definem hiperplanos em R^{n+m} perpendiculares entre si.

Prova: Seja uma restrição $a_i x \leq b_i$ de (1) e uma restrição $\bar{a}_j y \geq c_j$ de (3), onde a_i é uma linha de A ou não-negatividade, e \bar{a}_j é uma linha de A^T ou não-negatividade. Assim, em (5):

$$(*) \quad (a_i, 0) \cdot \begin{bmatrix} x \\ y \end{bmatrix} \begin{matrix} \leq b_i \\ \geq 0 \end{matrix} \quad \text{ou}$$

$$(**) \quad (0, \bar{a}_j) \cdot \begin{bmatrix} x \\ y \end{bmatrix} \begin{matrix} \geq c_j \\ \geq 0 \end{matrix} \quad \text{ou}$$

e então $(a_i, 0) \cdot (0, \bar{a}_j)^T = a_i \cdot 0 + 0 \cdot \bar{a}_j = 0$. Desta forma, os hiperplanos das restrições (*) e (**) são perpendiculares.

Definição 4: Denotaremos por \bar{S} o conjunto de pontos viáveis de:

$$(6) \quad \begin{bmatrix} A & 0 \\ 0 & A^T \\ I & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \begin{matrix} \leq b \\ \geq c \\ \geq 0 \\ \geq 0 \end{matrix}$$

Definição 5: Seja w um ponto fronteira de um conjunto convexo X ; então $cx = Z$ é chamado de hiperplano suporte por w se X está contido em um semi-espaço fechado determinado pelo hiperplano, e se $cw = Z$.

Lema 10: Seja $\bar{S} \neq \{\}$. Então $c^T x - b^T y = 0$ é hiperplano suporte de \bar{S} .

Prova : De fato, se $(x^T, y^T) \in \bar{S}$, então

$$c^T x \leq b^T y$$

ou ainda

$$c^T x - b^T y \leq 0$$

e além do mais, $\bar{S} \neq \{\}$ somente se os problemas (1) e (3) admitem solução ótima finita. Portanto, existem x^* e y^*

tais que

$$c^T x^* - b^T y^* = 0$$

e assim o hiperplano $c^T x - b^T y = 0$ é hiperplano suporte de \bar{S} por $((x^*)^T, (y^*)^T)$.

5.3. UM EXEMPLO DE PROBLEMA EQUIVALENTE PRIMAL-DUAL E SUA SOLUÇÃO GEOMÉTRICA

Consideremos o seguinte problema de PL:

$$(7.1) \quad \text{Max } Z = 2x \quad \text{sujeito a}$$

$$x \leq 3$$

$$x \geq 0$$

cujo dual será então

$$(7.2) \quad \text{Min } W = 3y \quad \text{sujeito a}$$

$$y \geq 2$$

$$y \geq 0.$$

Assim, a solução ótima para (7.1) será

$$x^* = 3$$

$$Z^* = 6$$

e para (7.2)

$$y^* = 2$$

$$W^* = 6.$$

E assim, temos

$$\bar{S} = \{(x, y) \in \mathbb{R}^2 / x \leq 3, x \geq 0, y \geq 2\}$$

e cuja representação geométrica será:

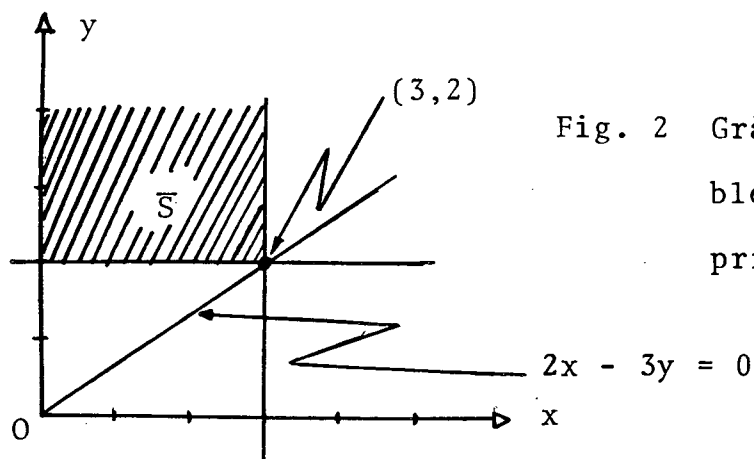


Fig. 2 Gráfico de um problema equivalente primal-dual

Vê-se que as restrições primais são perpendiculares às restrições duais, e que a reta $2x - 3y = 0$, é hiperplano suporte à \bar{S} por $(3,2)$. Como o único ponto comum ao hiperplano suporte e à \bar{S} é o ponto $(3,2)$, este ponto será solução do sistema (7.1) ou (7.2) e assim, de acordo com o lema 8, determina as soluções ótimas limitadas $x^* = 3$ e $y^* = 2$ para (7.1) e (7.2), respectivamente.

5.4. A SUBMATRIZ QUADRADA DA SOLUÇÃO DO PROBLEMA DE PROGRAMAÇÃO LINEAR

5.4.1. Base ótima e submatriz quadrada.

Sabemos, através da teoria da PL, que a todo problema do tipo (1) podemos associar um outro equivalente, no espaço \mathbb{R}^{n+m} , desde que adicionemos variáveis de folga às restrições $Ax \leq b$ de modo a obtermos igualdades. Assim, teremos:

$$\begin{aligned} \text{Max} &= c^T x \quad \text{sujeito a} \\ (8) \quad (A, I) &\begin{bmatrix} x \\ x_s \end{bmatrix} = b \\ &\begin{bmatrix} x \\ x_s \end{bmatrix} \geq 0 \end{aligned}$$

onde

$$x_S^T = (x_{S1}, x_{S2}, \dots, x_{Sm})$$

é o vetor das variáveis de folga. Se houver solução ótima limitada para (1), haverá solução ótima básica em (8) e vice-versa. Seja

$$B^* = \text{base da solução ótima e } r(B^*) = m$$

onde $r(B^*)$ é o rank de B^* , e seja

$$x_B^* = (x_{i_1}, \dots, x_{i_j}, x_{S1_1}, \dots, x_{S1_k})^T$$

o vetor da solução ótima tal que as j primeiras componentes de x_B^* sejam componentes de x do problema (1), e as restantes k sejam componentes de x_S . É claro que $j + k = m$. A existência de variáveis de folga em x_B^* significa que os hiperplanos das restrições das linhas l_1, \dots, l_k de $Ax \leq b$ não contêm o vértice da solução ótima, enquanto que os demais contêm.

Podemos eliminar, desta forma, as linhas l_1, \dots, l_k de $Ax \leq b$ sem modificar a solução ótima do problema (1). Além disso, as componentes de x do problema (1) que não se encontrarem entre as j primeiras componentes de x_B^* são nulas na solução ótima limitada de (1), por serem variáveis não-básicas. Sua eliminação também não modifica a otimalidade da solução.

Desta forma, podemos construir uma submatriz quadrada D^* a partir de A , pela eliminação das linhas l_1, \dots, l_k e das colunas i_1, \dots, i_j , com a finalidade de criar um sistema linear cuja solução é a solução básica ótima. Denominaremos D^* de subma-

triz quadrada da solução do problema de PL. Se x_B^* for totalmente formada de variáveis de folga, D^* é de ordem zero¹⁶, e se $k < m$, então a ordem de D^* será j .

A matriz D^* , por sua vez, determina a solução ótima x^* do problema (1) pela resolução do sistema linear reduzido

$$D^* \cdot \begin{bmatrix} x_{i_1} \\ \vdots \\ x_{i_1} \end{bmatrix} = \begin{bmatrix} b_{i_1} \\ \vdots \\ b_{i_1} \end{bmatrix}$$

5.4.2. Um exemplo numérico

Seja o problema

$$(9) \quad \left\{ \begin{array}{l} \text{Max } Z = 6x_1 + 7x_2 + x_3 \quad \text{sujeito à} \\ x_1 + x_2 + x_3 \leq 20 \\ x_1 + 2x_2 + 3x_3 \leq 30 \\ x_1 + x_2 + 2x_3 \leq 24 \\ x_1, x_2, x_3 \geq 0 \end{array} \right.$$

que se torna, com a adição das variáveis de folga em:

¹⁶ Pois então $\{l_1, \dots, l_k\} = \{1, \dots, m\}$, o que resulta na eliminação de todas as linhas e colunas de A .

$$(10) \quad \left\{ \begin{array}{l} \text{Max} = 6x_1 + 7x_2 + x_3 \quad \text{sujeito } \tilde{a} \\ x_1 + x_2 + x_3 + x_{S1} = 20 \\ x_1 + 2x_2 + 3x_3 + x_{S2} = 30 \\ x_1 + x_2 + 2x_3 + x_{S3} = 24 \\ x_1, x_2, x_3, x_{S1}, x_{S2}, x_{S3} \geq 0 \end{array} \right.$$

e cuja solução ótima é:

$$x_B^* = (x_1, x_2, x_{S3}) = (10, 10, 24)$$

e a base ótima B^* do sistema (10) é

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

de forma que

$$B^* \cdot x_B^* = b$$

e

$$Z^* = c^T \cdot x^* = (6, 7, 1, 0, 0, 0) \cdot (10, 10, 0, 0, 24)^T = 130$$

Como x_{S3} é componente de x_B^* e como x_3 não é componente de x_B^* , obtém-se a submatriz quadrada da solução do problema de PL pela eliminação da 3ª linha e 3ª coluna da matriz A de (9);

$$D^* = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

e assim a resolução do sistema

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 20 \\ 30 \end{bmatrix}$$

nos fornece a solução ótima de (9):

$$x_1^* = 10 \quad , \quad x_2^* = 10$$

e como x_3 não é variável básica de x_B^* de (10), temos

$$x_3^* = 0.$$

5.4.3. Lema da univocidade

Assim como construímos a submatriz quadrada D^* da solução do problema de PL, a partir do conhecimento das variáveis básicas que compõem a solução ótima, podemos construir submatrizes quadradas D pelo mesmo processo, para cada solução básica do problema.

A construção contrária também é possível. Dada uma submatriz quadrada D de A , podemos identificar, pelas linhas de A que se encontram, ausentes em D e colunas de A presentes em D , quais são as variáveis de folga e do problema que compõe a dada solução básica.

Deste modo, fica estabelecido o seguinte:

Lema 1.1: Para cada base possível B de (8) existe uma submatriz

quadrada D construída de acordo com a secção 5.4.3. e reciprocamente, a cada submatriz quadrada D de A corresponde uma base B de (8).

Portanto, o número de submatrizes quadradas de A do problema (1) é igual ao número máximo de bases possíveis para (8), ou seja,

$$\binom{m+n}{m}$$

Pode-se ainda obter este resultado pela soma do número de submatrizes quadradas das mais variadas ordens. Por exemplo, se $m \geq n$ temos

$$\begin{aligned} \text{número de submatrizes quadradas de ordem } i &= \binom{m}{i} \binom{n}{i} \\ \text{total} &= \sum_{i=0}^n \binom{m}{i} \binom{n}{i} = \binom{m+n}{n} \end{aligned}$$

5.5. SUBMATRIZ QUADRADA E O PAR PRIMAL-DUAL

Se particionarmos a matriz A do problema (1), de modo que D seja uma submatriz quadrada de A na forma

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & D \end{bmatrix}$$

e os vetores x e c na forma

$$x = \begin{bmatrix} x_N \\ x_D \end{bmatrix}, \quad c = \begin{bmatrix} c_N \\ c_D \end{bmatrix}$$

tal que o produto matricial seja possível, poderemos reescrever o problema (1) na forma

$$(11) \left\{ \begin{array}{l} Z = (c_N^T, c_D^T) \cdot \begin{bmatrix} x_N \\ x_D \end{bmatrix} \quad \text{sujeito a} \\ \begin{bmatrix} A_1 & A_2 \\ A_3 & D \end{bmatrix} \cdot \begin{bmatrix} x_N \\ x_D \end{bmatrix} \leq \begin{bmatrix} b_N \\ b_D \end{bmatrix} \\ x_N, x_D \geq 0 \end{array} \right.$$

e um particionamento em y do problema (3) nos permitirá reescrevê-lo como

$$(12) \left\{ \begin{array}{l} W = (b_N^T, b_D^T) \cdot \begin{bmatrix} y_N \\ y_D \end{bmatrix} \quad \text{sujeito a} \\ \begin{bmatrix} A_1^T & A_3^T \\ A_2^T & D^T \end{bmatrix} \cdot \begin{bmatrix} y_N \\ y_D \end{bmatrix} \geq \begin{bmatrix} c_N \\ c_D \end{bmatrix} \\ y_N, y_D \geq 0 \end{array} \right.$$

Vamos agora considerar o caso de que, dada a submatriz D , desejamos resolver

$$(13) \quad \begin{cases} x_N = 0 \\ y_N = 0 \\ Dx_D = b_D \\ D^T y_D = c_D \end{cases}$$

com D não singular, sem a preocupação de que a solução (13) satisfaça a (11) e (12). Então;

Teorema 3 : A solução do sistema (13), é um ponto do hiperplano $(c^T x - b^T y = 0)$ do PEPD

$$\begin{aligned} \text{Prova} \quad : \quad c^T \cdot x &= c_D^T \cdot x_D = (D^T \cdot y_D)^T \cdot x_D = (y_D^T \cdot D) \cdot x_D \\ &= y_D^T \cdot (D \cdot x_D) = y_D^T \cdot b_D = y^T \cdot b = b^T y \end{aligned}$$

e assim

$$c^T x - b^T y = 0$$

O teorema a seguir, estabelece a relação entre a submatriz quadrada D^* de A que determina a solução ótima limitada x^* para (1), quando existir, e a submatriz quadrada \bar{D}^* de A^T que determina a solução ótima limitada y^* para o seu dual (3). Veremos que $\bar{D}^* = (D^*)^T$.

Teorema 4 : Se o problema (1) possui solução ótima limitada, e se D^* for sua submatriz quadrada da solução do problema

de PL, então $(D^*)^T$ será a submatriz quadrada da solução do seu problema dual de PL.

Prova: Como estamos interessados apenas nas soluções básicas, podemos reescrever (11) como

$$A_1 x_N + A_2 x_D \leq b_N$$

$$A_3 x_N + D x_D = b_D$$

e então

$$x_D = D^{-1} \cdot b_D - D^{-1} \cdot A_3 \cdot x_N$$

$$Z = c_D^T \cdot x_D + c_N^T \cdot x_N$$

$$Z = c_D^T \cdot D^{-1} \cdot b_D + (c_N^T - c_D^T \cdot D^{-1} \cdot A_3) x_N \quad (*)$$

e se $Z = Z^*$, teremos $D = D^*$ e

$$c_N^T \leq c_D^T \cdot D^{-1} \cdot A_3$$

$$A_3^T ((D^{-1})^T \cdot c_D) \geq c_N$$

mas $(D^{-1})^T = (D^T)^{-1}$ e assim

$$A_3^T ((D^T)^{-1} \cdot c_D) \geq c_N \quad (**)$$

e de (13) temos

$$y_D = (D^T)^{-1} \cdot c_D$$

e assim, substituindo em (**)

$$A_3^T y_D \geq c_N$$

e portanto

$$Ay \geq c$$

Falta provar a não-negatividade de y . Quando considerarmos a não-negatividade das variáveis de folga acrescentadas ao sistema (11) de modo a transformá-lo em um sistema do tipo (8), obteremos, de acordo com (*):

$$c_D D^{-1} e_i - 0 \geq 0 \quad (1 \leq i \leq m)$$

$$\text{para } x_{Si} \geq 0 \quad (1 \leq i \leq m)$$

onde e_i é o vetor da base canônica da i -ésima coordenada.

Assim,

$$c_D \cdot D^{-1} \cdot (e_1, \dots, e_m) \geq 0$$

ou ainda

$$c_D \cdot D^{-1} \cdot I \geq 0$$

e portanto

$$c_D^T D^{-1} \geq 0$$

$$y_D^T \geq 0$$

$$y \geq 0$$

e assim (12) é satisfeito. Como a presente solução y satisfaz também a (13), temos, pelo teorema (4), que y é solução ótima limitada para o dual (3) determinada por $D^T = (D^*)^T$.

CAPÍTULO VI

6. O ALGORITMO APROXIMATIVO DE PROGRAMAÇÃO LINEAR

6.1. METODOLOGIA DO ALGORITMO

O Algoritmo Aproximativo de PL, que será apresentado na seção 6.2., baseia-se na teoria exposta nos capítulos 3 e 4, além de utilizar os resultados de mais dois teoremas expostos nesta seção.

Sua idéia principal é a de obter a submatriz quadrada D^* que determina a solução ótima para o problema (1), caso ela exista¹⁷. No entanto, embora o propósito seja o de resolver o clássico problema de PL tipo (1), o algoritmo irá trabalhar sobre o seu PEPD¹⁸, e tentar determinar a otimalidade de (1), caso exista, baseado na viabilidade de seu PEPD¹⁹. Para obtermos esta submatriz quadrada D^* da matriz A , ou então $(D^*)^T$ de A^T , devemos determinar quais são as n restrições de (1) ou quais são as m restrições de seu dual (3) que determinam um vértice de otimalidade do poliedro convexo de soluções compatíveis de (1) ou (3) através da intersecção de seus hiperplanos.

¹⁷ Baseado no teorema 4. Veja na seção 5.5.

¹⁸ Conforme definido por (4) e (5). Veja na seção 5.2.

¹⁹ Baseado no lema 8. Veja na seção 5.2.

Para obter estas restrições, o algoritmo efetuará estes procedimentos:

- a. partirá de um vetor x inicial qualquer (vetor zero, por exemplo);
- b. se não for solução compatível para o PEPD determina-se o conjunto de restrições violadas e faz-se a projeção ortogonal sobre o hiperplano da restrição violada mais distante;
- c. se o ponto não for solução compatível, continuará o processo descrito em (b). Caso exista tendência de se estabelecer um LOOP de projeções entre dois hiperplanos, determina-se diretamente o ponto limite deste LOOP de acordo com a equação (2)²⁰ e continua o processo (b).

Desta forma, os sucessivos pontos obtidos se aproximam da solução viável (caso exista) do PEPD²¹.

Provaremos agora que se um ponto estiver suficientemente próximo de uma solução viável do PEPD, ele poderá violar somente restrições de hiperplanos que contenham esta solução viável, porém nunca violará restrições de hiperplanos que não o contenham. Para tal, devemos ter em mente que toda solução ótima para (1) é um ponto fronteira de seu conjunto S de soluções compatíveis, e portanto, toda solução viável do conjunto \bar{S} de soluções compatíveis do PEPD será também um ponto fronteira de \bar{S} .

²⁰ Ver teorema 1 na seção 4.2.

²¹ Lema 3 aplicado ao PEPD e teorema 2 (veja na seção 4.2.).

Teorema 5: Seja o sistema linear abaixo

$$Ax \begin{cases} \leq \\ \geq \\ = \end{cases} b$$

e S o conjunto de soluções viáveis do mesmo. Se $v \in S$ e se $J = \{i / a_i x \neq b_i\}$, então existe uma vizinhança aberta $V(v, \xi)$ centrada em v e de raio $\xi > 0$ tal que todo $\bar{x} \in V(v, \xi)$ satisfaz às restrições em J .

Prova : Seja ξ definido como

$$\xi = \min \{d(v, H_i) / i \in J\}$$

Então, se $\bar{x} \in V(v, \xi)$, temos

$$|\bar{x} - v| < \xi$$

e para qualquer $i \in J$ teremos

$$\xi \leq d(v, H_i) = \frac{|b_i - a_i v|}{|a_i|}$$

e portanto

$$|\bar{x} - v| < \frac{|b_i - a_i v|}{|a_i|}$$

$$|a_i \bar{x} - a_i v| < |b_i - a_i v|$$

que conduz a duas alternativas

$$(a) - (b_i - a_i v) < a_i \bar{x} - a_i v < b_i - a_i v$$

$$(b) b_i - a_i v < a_i \bar{x} - a_i v < -(b_i - a_i v).$$

O desenvolvimento de (a) nos conduz à

$$\left\{ \begin{array}{l} a_i v - b_i < a_i \bar{x} - a_i v \\ a_i \bar{x} - a_i v < b_i - a_i v \end{array} \right.$$

$$\left\{ \begin{array}{l} a_i v < (a_i \bar{x} + b_i)/2 \\ a_i \bar{x} < b_i \end{array} \right.$$

$$\left\{ \begin{array}{l} a_i v < b_i \\ a_i \bar{x} < b_i \end{array} \right.$$

e um desenvolvimento semelhante nos conduz a

$$\left\{ \begin{array}{l} a_i v > b_i \\ a_i \bar{x} > b_i \end{array} \right.$$

Estes dois resultados obtidos, a partir de (a) e (b), nos mostram que \bar{x} satisfaz a restrição $i \in J$ do mesmo modo que v .

O PEPD apresenta-se exatamente na forma descrita pelo enunciado do Teorema 5. Sabemos que a execução dos procedimentos (b) e (c) descritos na página 56 convergem a seqüência de pontos projetados para a solução

$$v = \begin{bmatrix} x^* \\ y^* \end{bmatrix}$$

do PEPD, caso exista. Se

$$(\bar{x}_i) = \left(\begin{bmatrix} x_i \\ y_i \end{bmatrix} \right), \quad i = 1, 2, 3, \dots$$

for a seqüência de pontos projetados, então, dada a vizinhança $V(v, \xi)$ conforme o enunciado do Teorema 5, deve existir um inteiro positivo n tal que

$$\bar{x}_i \in V(v, \xi) \quad , \quad \forall_i \geq n.$$

As projeções ortogonais para $i \geq n$, portanto, serão efetuadas apenas sobre hiperplanos que contenham a solução ótima v , os quais são fundamentais para a formação de D^* .

Embora o algoritmo descrito neste trabalho não calcule ou determine quando a condição $\bar{x}_i \in V(v, \xi)$ é satisfeita, utiliza a idéia de colecionar hiperplanos atingidos por projeções sobre restrições violadas desde o princípio de sua execução, como descrito acima, e toda vez que D ou D^T é formado a partir de uma coleção de n hiperplanos primais ou m hiperplanos duais completada, resolvemos o sistema

$$\begin{aligned} D x_D &= b_D \\ D^T y_D &= c_D \end{aligned}$$

desde que D seja não-singular, e formamos o novo vetor $\bar{x} = (x^T, y^T)^T$, com as componentes de x que não são as de x_D e as componentes de y que não são as de y_D nulas, conforme (13), garantindo desta forma que

$$c^T x - b^T y = 0$$

sem garantir, no entanto, que o PEPD seja satisfeito com este novo ponto. Se tal acontecer, o algoritmo reinicia por uma nova série de repetições de projeções, obedecendo os procedimentos (b) e (c) até que nova coleção de restrições nos possibilite obter nova submatriz quadrada D ou D^T .

No procedimento (b), foi descrito que a projeção ortogonal é efetuada sobre o hiperplano da restrição violada mais distante. Tal idéia é baseada sobre o fato de apressar a aproximação dos sucessivos valores de \bar{x} ao seu conjunto de soluções viáveis \bar{S} , justificado pelo teorema abaixo.

Teorema 6: Seja $\{H_i, 1 \leq i \leq p\}$ uma coleção de hiperplanos tais que $\bigcap_{i=1}^k H_i \neq \{\}$. Seja $x \in \mathbb{R}^n$ e seja j um índice ($1 \leq j \leq p$) tal que

$$d(x, H_j) = \max \{d(x, H_i), 1 \leq i \leq p\}.$$

Então, para todo $z \in \bigcap_{i=1}^k H_i$ vale

$$|p(x, H_j) - z| \leq |p(x, H_i) - z| \quad (1 \leq i \leq p)$$

Prova : Vamos considerar $x_i = p(x, H_i)$ ($1 \leq i \leq p$).

Assim,

$$x - z = (x - x_i) + (x_i - z)$$

$$(x - z)^2 = (x - x_i)^2 + 2(x - x_i)(x_i - z) + (x_i - z)^2$$

mas

$$\begin{aligned} (x - x_i)(x_i - z) &= \frac{(b_i - a_i x)}{|a_i|^2} \cdot a_i \cdot (x_i - z) \\ &= \frac{(b_i - a_i x)}{|a_i|^2} (a_i x_i - a_i z) \end{aligned}$$

e como $x_i \in H_i$, $z \in H_i$ temos $a_i x_i = a_i z = b_i$

e assim

$$(x - z)^2 = (x - x_i)^2 + (x_i - z)^2$$

e portanto

$$(x - x_j)^2 + (x_j - z)^2 = (x - x_i)^2 + (x_i - z)^2 \quad (1 \leq i \leq p)$$

e como $d(x, H_i) = |x - x_i|$ $(1 \leq i \leq p)$

temos

$$(x - x_j)^2 - (x - x_i)^2 = (x_i - z)^2 - (x_j - z)^2 \geq 0$$

e então

$$|x_j - z| \leq |x_i - z|$$

Deste modo, para qualquer coleção de hiperplanos de restrições violadas pelo ponto \bar{x} , de tal maneira que haja um ponto comum z a todos eles, a projeção de maior comprimento nos aproximará mais de z . Como z satisfaz a todas as restrições da coleção de hiperplanos, z estará mais próximo de \bar{S} que qualquer uma das possíveis projeções sobre os hiperplanos considerados. Assim, espera-se que a projeção de maior comprimento, entre todas as restrições violadas, quer seus hiperplanos apresentem ponto comum ou não, nos aproxime mais de \bar{S} embora possa-se mostrar que isto nem sempre ocorre.

6.2. DESCRIÇÃO DO ALGORITMO

6.2.1. Notação utilizada

Os seguintes vetores e conjuntos integram o algoritmo aproximativo de PL:

- \bar{x} : vetor de $n+m$ componentes do PEPD
- K : vetor que armazena índices sucessivos das restrições violadas mais distantes
- PRIM : conjunto que armazena índices das restrições violadas mais distantes do primal
- DUAL : ídem, do dual

e além disso, teremos

- NUM : contador que controla o número de iterações e serve de índice ao vetor K
- $o(\text{PRIM})$: cardinalidade do conjunto PRIM
- $o(\text{DUAL})$: ídem, do conjunto DUAL
- $p(\bar{x}, H_i, H_j)$: projeção de \bar{x} sobre o ponto limite de um LOOP entre H_i e H_j , conforme equação (2)
- MARCA : variável controladora para teste de LOOP entre $K(\text{NUM})$ e $K(\text{NUM}-1)$
- k : índice da restrição violada mais distante; poderemos convencionar, por exemplo:

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \stackrel{=}{=} \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \quad \begin{array}{l} (k = 1) \\ \vdots \\ (k = m) \end{array}$$

$$\begin{bmatrix} x_{n+1} & \dots & x_{n+m} \end{bmatrix}$$

$$\stackrel{=}{\geq}$$

$$\begin{bmatrix} c_1 & \dots & c_n \end{bmatrix}$$

$$(k = m+1) \dots (k = m+n)$$

$$\sum_{j=1}^n c_j x_j - \sum_{j=1}^m b_j x_{n+j} = 0 \quad (k = m+n+1)$$

$$x_j \geq 0 \quad (j = 1, \dots, n+m) \quad (k=1000+j)^{22}$$

assumindo que $m + n + 1 < 1000$

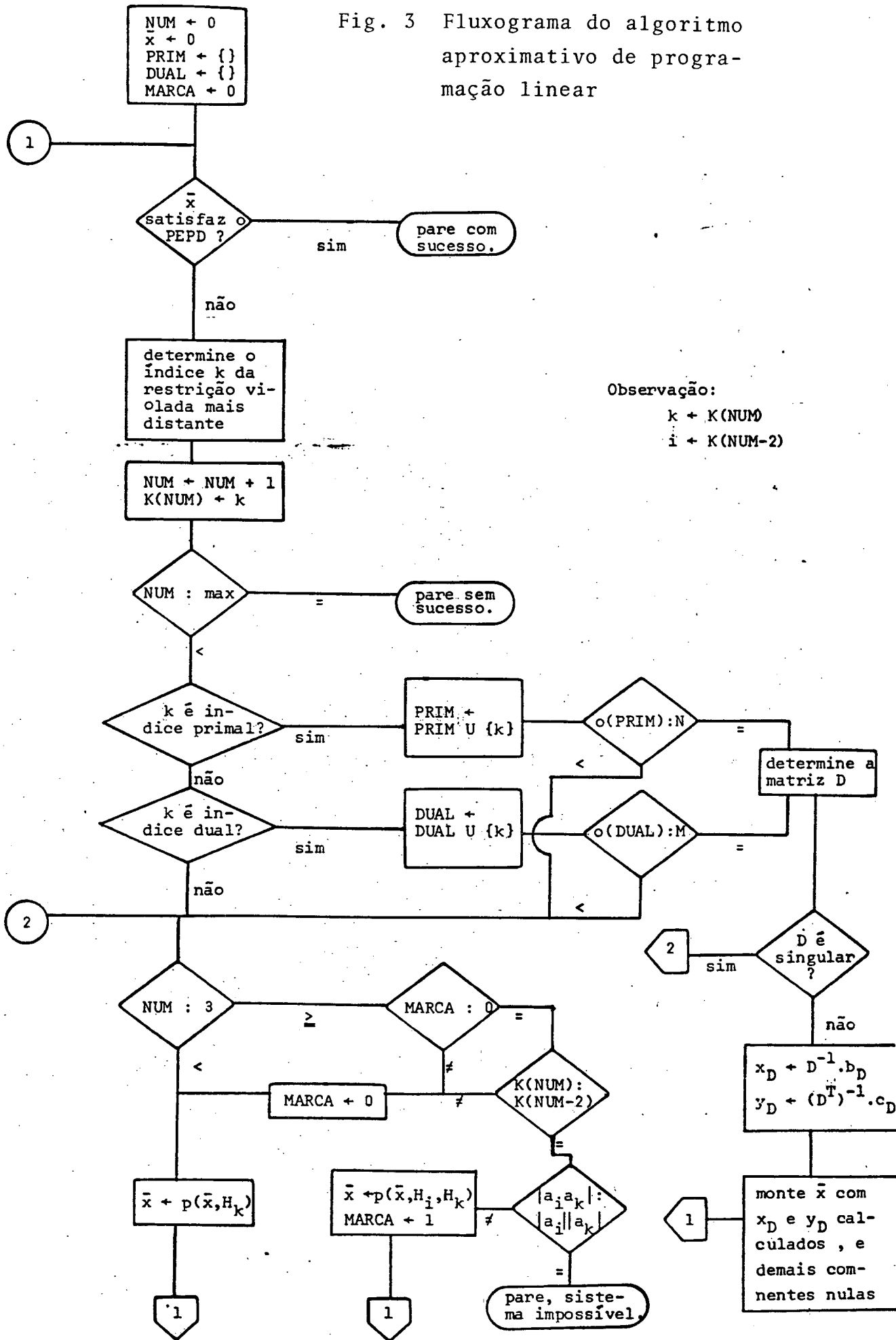
max : um certo número inteiro positivo, considerado como limite de utilização do algoritmo.

6.2.2. O algoritmo

Uma vez estabelecidas as notações acima, o algoritmo aproximativo de PL pode ser descrito como segue no fluxograma da página seguinte.

²² Esta convenção foi utilizada para o programa computacional desenvolvido pelo autor deste trabalho.

Fig. 3 Fluxograma do algoritmo aproximativo de programação linear



Observação:

$k \leftarrow K(\text{NUM})$
 $i \leftarrow K(\text{NUM}-2)$

6.3. UM EXEMPLO COMPLETO

Consideremos o problema de PL abaixo:

$$\text{Max } Z = 5382 x_1 + 7366 x_2 + 4176 x_3$$

sujeito à

$$42 x_1 + 66 x_2 - 53 x_3 \leq 3375$$

$$37 x_1 + 45 x_2 + 72 x_3 \leq 13380$$

$$x_1, x_2, x_3 \geq 0$$

o qual pode ser apresentado como

	x_1	x_2	x_3	b_i	
x_4	42	66	-53	3375	(k = 1)
x_5	37	45	72	13380	(k = 2)
c_j	5382	7366	4176		
	(k=3)	(k=4)	(k=5)		

(PEPD)

em que $k = \text{índice da restrição}$. Convencionaremos

$$k = 1000 + i \quad \text{para } x_i \geq 0 \quad (1 \leq i \leq 5)$$

e

$$k = 6 \quad \text{para } c^T x - b^T y = 0.$$

Inicialização do algoritmo:

$$\text{PRIM} = \{\}$$

$$\text{DUAL} = \{\}$$

$$\bar{x}_0 = (0, 0, 0, 0, 0).$$

De acordo com o fluxograma estabelecido para o algoritmo, teremos:

restrição violada mais distante = 3

$$\bar{x}_1 = p(\bar{x}_0, H_3) = (0,0, 0,0, 0,0, 72,15, 63,56)$$

$$\text{PRIM} = \{\}, \quad \text{DUAL} = \{3\}$$

restrição violada mais distante = 6

$$\bar{x}_2 = p(\bar{x}_1, H_6) = (20,23, 27,68, 15,69, 59,47, 13,28)$$

$$\text{PRIM} = \{\}, \quad \text{DUAL} = \{3\}$$

restrição violada mais distante = 5

$$\bar{x}_3 = p(\bar{x}_2, H_5) = (20,23, 27,68, 15,69, 48,76, 92,19)$$

$$\text{PRIM} = \{\}, \quad \text{DUAL} = \{3,5\}$$

e como $\#(\text{DUAL}) = m = 2$

temos

$$D = \begin{bmatrix} 42 & -53 \\ 37 & 72 \end{bmatrix}$$

e a resolução de

$$D x_D = b_D \quad \text{e} \quad D^T y_D = c_D$$

fornece

$$\bar{x}_4 = (191,0, 0,0, 87,68, 46,74, 92,40)$$

$$\text{PRIM} = \{\}, \quad \text{DUAL} = \{\}$$

restrição violada mais distante = 4

$$\bar{x}_5 = p(\bar{x}_4, H_4) = (191,0, 0,0, 87,68, 48,01, 93,27)$$

$$\text{PRIM} = \{\}, \quad \text{DUAL} = \{4\}$$

restrição violada mais distante = 6

$$\bar{x}_6 = p(\bar{x}_5, H_6) = (191,30, 0,40, 87,91, 47,83, 92,54)$$

$$\text{PRIM} = \{\}, \quad \text{DUAL} = \{4\}$$

restrição violada mais distante = 4

como a seqüência de hiperplanos 4, 6, 4 mostra tendência de estabelecimento de um LOOP entre os hiperplanos 4 e 6, determinamos x_7 de acordo com

$$\bar{x}_7 = p(\bar{x}_6, H_4, H_6) = (191,47, 0,63, 88,04, 48,46, 92,62)$$

restrição violada mais distante = 5

$$\text{PRIM} = \{\}, \quad \text{DUAL} = \{4,5\}$$

e como o (DUAL) = $m = 2$

temos

$$D = \begin{bmatrix} 66 & -53 \\ 45 & 72 \end{bmatrix}$$

e a resolução de

$$D x_D = b_D \quad \text{e} \quad D^T y_D = c_D$$

fornece

$$\bar{x}_8 = (0,0, 133,41, 102,45, 47,98, 93,32)$$

que satisfaz à todas as restrições do PEPD, e assim $\bar{x}^* = \bar{x}_8$, com

$$Z^* = W^* = 1410532$$

6.4. COMPLEXIDADE-TEMPO DE SUAS PARTES

Para determinar a complexidade-tempo de um procedimento computacional devemos determinar o número de multiplica-

ções e divisões efetuadas em função do tamanho da entrada de dados a este procedimento, visto que estas operações são as mais relevantes quando o tempo assume proporções de interesse prioritário. No caso de um problema de PL, os parâmetros de interesse são o número de variáveis n e o número de restrições m , e toda expressão de complexidade-tempo será dada em função de n e m .

Se examinarmos o algoritmo, veremos que os passos

- a. Testar \bar{x} no PEPD
- b. $\bar{x} \leftarrow p(\bar{x}, H_k)$
- c. $\bar{x} \leftarrow p(\bar{x}, H_i, H_k)$
- d. $x_D \leftarrow D^{-1} \cdot b_D$ e $y_D \leftarrow D^{-1} \cdot c_D$

são os únicos que necessariamente envolvem produtos e quocientes. Vamos avaliar a complexidade-tempo de cada um.

- a. Para testar $Ax \leq b$ necessitamos efetuar n produtos durante m vezes, e para testar $A^T y \geq c$, precisamos de m produtos n vezes. O teste $c^T x - b^T y = 0$ necessita de $n+m$ produtos. Os testes de não-negatividade não envolvem produtos. Total: $(2nm + m + n)$.
- b. Para efetuar $\bar{x} \leftarrow p(\bar{x}, H_k)$ teremos que efetuar

$$\bar{x} \leftarrow \bar{x} + \frac{(b_k - a_k \bar{x})}{|a_k|^2} a_k$$

e como $(b_k - a_k \bar{x})$ foi previamente computado em (a), não será necessário repetí-lo. Por outro lado, $|a_k|$ pode ser computado e armazenado (para cada k primal ou dual) na inicialização do algoritmo uma única vez. Temos, então, um quociente e

n produtos se k for decimal

m produtos se k for dual

(n + m) produtos se k representa a restrição

$$c^T x - b^T y = 0$$

Total: (n+1) ou (m+1) ou (n + m + 1).

c. Para efetuar $\bar{x} + p(\bar{x}, H_i, H_k)$ teremos que efetuar

$$\bar{x} + \bar{x} + (a_i^2 a_k - (a_i a_k) a_i) \cdot \frac{(b_k - \bar{x} a_k)}{a_i^2 a_j^2 - (a_i a_k)^2}$$

sendo que $b_k - a_k \bar{x}$ foi previamente computado. Assim,

Computar	Produtos	
	PRIMAL	DUAL
$a_i a_k$	n	m
$a_i^2 a_k$	n	m
$(a_i a_k) a_i$	n	m
Total	3n	3m

temos ainda que computar $a_i^2 \cdot a_k^2$, $(a_i a_k)^2$ e um único quociente. Total: $3n + 3$ ou $3m + 3$.

d. A matriz D poderá ter no máximo dimensão n ou dimensão m, e sua resolução pelo método da diagonalização nos dará um total de $(n^3 + n^2)/2$ ou $(m^3 + m^2)/2$ operações de multiplicação e quocientes, no máximo.

CAPÍTULO VII

7. RESULTADOS DA APLICAÇÃO DO ALGORITMO

7.1. RESULTADOS COMPUTACIONAIS OBTIDOS

O autor do presente trabalho elaborou um programa computacional do algoritmo apresentado no capítulo anterior, escrito em linguagem FORTRAN IV²³, e testou um conjunto de 70 problemas de PL de pequeno porte com o mesmo. Estes problemas foram gerados com os coeficientes da matriz A do problema clássico (1) de forma aleatória, distribuídos uniformemente entre -100 e +100, e os vetores b e c foram construídos de tal forma que todos os problemas tivessem solução ótima limitada.

Uma pequena inovação no programa computacional criado, que não consta na descrição do algoritmo, consiste em coleccionar os índices de restrições violadas em PRIM e DUAL quando (n + m) projeções foram efetuadas. Tal medida heurística deve-se à idéia de permitir, a partir das (n + m) primeiras projeções, uma maior aproximação ao conjunto das soluções do PEPD. Assim, poderia criar-se uma maior probabilidade de que a primeira coleção completa de índices que determinam a matriz D contenham os índices que determinam D*.

²³ Veja nos anexos

Verificou-se, a partir dos resultados, que:

- a. soluções ótimas foram encontradas;
- b. houve situações em que repetiram-se periodicamente as soluções de $D x_D = b_D$ e $D^T y_D = c_D$, portanto formando um laço infinito sem encontrar a solução ótima;
- c. ocorreram casos em que o número de projeções superou o limite previsto para 2000 projeções, mesmo sem a ocorrência do fenômeno descrito em (b).

A tabela estatística (1) apresenta a descrição dos resultados obtidos, de forma que:

- . as 3 primeiras colunas indicam a ordem sequencial dos problemas e seus tamanhos, em termos de número de restrições m (fora as não-negatividades) e número de variáveis n ;
- . a coluna 4 indica o número de projeções efetuadas até a parada de execução de cada problema;
- . a coluna 5 indica o número de sistemas $Dx_D = b_D$ e $D^T y_D = c_D$ resolvidos no caso (a). Indica LOOP se ocorreu o caso (b), e indica EXC se ocorreu o caso (c).

TABELA 1

DADOS ESTATÍSTICOS SOBRE OS PROBLEMAS TESTADOS

PROBLEMA NÚMERO	TAMANHO		NÚMERO DE PROJEÇÕES	NÚMERO DE SISTEMAS
	m	n		
1	3	5	16	3
2	3	5	13	1
3	3	5	77	1
4	3	5	66	3
5	3	5	40	2
6	3	5	262	2
7	3	5	16	2
8	3	5	57	2
9	3	5	59	2
10	3	5	271	3
11	3	5	167	3
12	5	3	53	1
13	5	3	79	2
14	5	3	22	1
15	5	3	11	1
16	5	3	81	2
17	5	3	107	3
18	5	3	192	3
19	5	3	124	1
20	5	3	141	2
21	5	3	14	1
22	4	6	273	2
23	4	6	1958	3
24	4	6	63	1
25	4	6	885	3
26	4	6	126	2
27	4	6	384	3
28	4	6	> 2000	EXC
29	4	6	50	1
30	4	6	60	1
31	4	6	> 2000	LOOP
32	6	4	> 2000	LOOP
33	6	4	101	2
34	6	4	37	2
35	6	4	> 2000	EXC
36	6	4	68	2
37	6	4	> 2000	LOOP
38	6	4	> 2000	LOOP
39	6	4	518	1
40	6	4	1716	4
41	6	4	514	0
42	4	8	> 2000	LOOP
43	4	8	> 2000	LOOP
44	4	8	895	3
45	4	8	62	2
46	4	8	85	2
47	8	4	145	4
48	8	4	515	4
49	8	4	> 2000	EXC
50	8	4	> 2000	EXC
51	8	4	45	1
52	8	4	24	1
53	10	6	45	1
54	10	6	91	2
55	10	6	> 2000	EXC
56	10	6	> 2000	EXC
57	10	6	> 2000	EXC
58	6	10	> 2000	LOOP
59	6	10	574	2
60	6	10	> 2000	EXC
61	6	10	84	1
62	6	10	613	3
63	3	6	49	1
64	7	3	44	2
65	5	5	1936	1
66	8	4	24	1
67	4	9	458	2
68	9	4	81	1
69	8	6	1998	1
70	12	6	96	2

7.2. INTERPRETAÇÃO DOS RESULTADOS

A verificação da tabela (1) permite concluir o seguinte:

- a. em 78,6% dos casos examinados foi encontrada a solução ótima, e nos demais 21,4% não;
- b. a proporção de problemas em que a solução ótima não foi encontrada aumenta sensivelmente com o tamanho do problema;
- c. apenas em um caso ocorreu não haver necessidade de resolver sistemas lineares, por encontrar-se o vetor x , após determinado número de projeções, suficientemente próximo da solução ótima, dentro dos limites da tolerância admitida pelo programa;
- d. nos demais casos em que foi encontrada a solução ótima, o número de sistemas do tipo

$$D x_D = b_D \quad \text{e} \quad D^T y_D = c_D$$

resolvidos ficou limitado entre 1 e 4, com a distribuição de frequências dada pela tabela 2.

Além disso, o autor do presente trabalho submeteu os casos de insucesso a outro programa computacional, via simplex. Constatou que nestes casos, a última solução do sistema $D x_D = b_D$ e $D^T y_D = c_D$, antes da parada pelo algoritmo apresentado, encontrava-se bastante próxima da solução ótima do problema, apesar de sua não-viabilidade.

Tabela 2

OCORRÊNCIA DE RESOLUÇÕES DE
SISTEMAS LINEARES

NÚMERO DE SISTEMAS	NÚMERO DE CASOS
1	20
2	20
3	11
4	3
TOTAL	54

CAPÍTULO VIII

8. CONCLUSÃO

8.1. ~~ANÁLISE DOS RESULTADOS~~

Para que um algoritmo seja competitivo com os demais, é necessário que além de possuir todas as qualidades dos demais já existentes, quer em termos de operacionalidade e análises efetuadas, quer em termos de facilidade de implantação computacional, ou ainda em precisão computacional dos resultados, o mesmo ofereça pelo menos alguma outra vantagem que os demais não possuam.

Conforme a tabela (1) e sua interpretação na seção 7.2. , podemos concluir que o algoritmo proposto neste trabalho não é um bom instrumento de resolução de problemas de PL, mesmo no caso de existência de solução ótima limitada, visto já existirem hoje programas computacionais implantados que resolvem problemas de PL de mais de 10.000 restrições e variáveis, indicando inclusive a inexistência de soluções viáveis e a não-existência de solução ótima limitada.

A deficiência do algoritmo proposto apresenta-se principalmente em dois pontos. O primeiro deles é a de que o algoritmo proposto não garante encontrar a solução ótima limitada, mesmo caso ela exista. Conclui-se desta forma que, mesmo que as projeções entre os hiperplanos aproximem o vetor x do problema de PL de sua so-

lução ótima limitada, a solução dos sistemas lineares podem afastar o vetor x da solução ótima limitada, pois senão a repetição do mesmo sistema linear não poderia ocorrer. O outro ponto está relacionado à indeterminação de sua complexidade-tempo. Mesmo que a solução ótima seja encontrada, não há meios de determinar o número de projeções e resoluções de sistemas lineares em função do tamanho do problema. Podemos calcular apenas a complexidade-tempo de seus passos.

8.2. CONSIDERAÇÕES FINAIS

Faz-se assim necessário um desenvolvimento da teoria envolvida, de modo a incorporar instrumentos ao algoritmo proposto ou então modificar o algoritmo.

Seria importante também que se conseguisse generalizar a equação de determinação do ponto limite de uma sequência de projeções ortogonais entre dois hiperplanos para uma situação idêntica envolvendo maior número de hiperplanos. Desta forma, o ponto-limite calculado satisfaria um maior número de restrições, em média.

Devido à ocorrência, em alguns casos, da repetição de sistemas lineares e suas soluções, o que cria um laço infinito, seria interessante pesquisar um modo de evitá-lo. Poderia talvez ser conseguido se for possível determinar qual dos hiperplanos da coleção que forma a matriz D do sistema repetido não contém o vértice da solução ótima, ou o mais provável neste sentido. Desta forma, poderia criar-se um método de eliminação e posterior subs-

tituição que levasse à solução ótima do problema. Outra causa possível da ocorrência destas repetições é a falta de precisão numérica computacional, o que pode acarretar erro decisório na escolha da restrição violada mais distante, devido à falta de precisão no cômputo das distâncias euclidianas.

Nos problemas testados pelo autor deste trabalho, nos casos em que ocorreu insucesso, a solução final encontrada através da resolução de sistemas lineares, antes da parada do algoritmo, é bem próxima da solução ótima. Isto sugere um ponto positivo em favor do algoritmo, se levarmos em conta que nos modelos reais raramente temos restrições rigidamente fixas. Isto é, uma restrição do tipo

$$a_i x \leq b_i$$

no modelo matemático não significa que a mesma restrição correspondente real não admita uma pequena tolerância em sua mão direita, de forma que

$$a_i x \leq b_i + \xi_i, \quad \xi_i \cong 0$$

expresse melhor a restrição considerada.

Assim, uma solução encontrada pelo presente algoritmo, próxima da solução ótima do problema (1), mesmo que inviável, poderá se tornar uma solução viável quando admitidas certas tolerâncias ξ_i incorporadas às restrições. Poderá ser tão boa quanto a solução ótima para (1).

Sob este prisma, uma boa sugestão é a de aperfeiçoar o algoritmo de modo que se crie uma regra de parada em função do conjunto de tolerâncias admitidas.

REFERÊNCIAS BIBLIOGRÁFICAS

01. BIRKHOFF, G. and MACLANE, S. A., "A survey of modern algebra". Macmillan, New York, 1941.
02. CHARNES, A. e outros, "An introduction to linear programming". Wiley, New York, 1958.
03. DANTZIG, G. B. and ORDEN, A., "Duality theorems". RM-1265, the RAND Corp., oct., 1953.
04. DANTZIG, G. B., "The dual simplex algorithm". RM-1270, the RAND Corp., jul., 1954.
05. DANTZIG, George B., "Thoughts on linear programming and automation". Management sci., vol. 3, jan., 1957, pp. 131-139.
06. GASS, S. I., "Linear programming". Mc Graw Hill Book Company, New York, 1958.
07. HADLEY, G., "linear programming". Addison - Wesley Publishing Company, 8ª ed., 1974.
08. HADLEY, G., "Algebra linear". Ed. Forense Universitária, RJ, 1979.
09. KACHIYAN, L. G., "A polynomial algorithm in linear programming (in russ)". Doklady Akademii Nauk SSSR, 1979, vol.244, nº5, pp. 1093 - 1096.

10. ORCHARD-HAYS, William, "Matrices elimination and the simplex method". CEIR, Inc., Arlington, Oct., 1961.
11. ORCHARD-HAYS, William, "Advanced linear-programming computing techniques". Mc Graw-Hill Book Company, New York, 1968 .
12. PUCCINI; Abelardo de L., "Introdução à programação linear". Livros Técnicos e Científicos Ed., RJ, 1978.
13. RÖDDER, W. e BLAETH, M., "Project - an alternative LP-code". Boletim de Produção e Sistemas - UFSC, Florianópolis, 2(2): 24-35, 1980.
14. RÖDDER, W., " A note on linear dependency in project". Boletim de Produção e Sistemas - UFSC, Florianópolis, 2(3):52-63, 1981.

A N E X O 1
= = = = = =

ESPECIFICAÇÃO DOS CARTÕES DE ENTRADA

CARTÃO TIPO 1

COLUNAS	CAMPO	DESCRIÇÃO	FORMATO
1 à 2	1	Número de problemas a resolver	I2

CARTÃO TIPO 2

COLUNAS	CAMPO	DESCRIÇÃO	FORMATO
1 à 52	1	Título do problema	13A4

CARTÃO TIPO 3

COLUNAS	CAMPO	DESCRIÇÃO	FORMATO
1 à 3	1	Número de restrições do problema (m)	I3
4 à 6	2	Número de variáveis do problema (n)	I3
7 à 9	3	Controle do formato da entrada da matriz dos coeficientes do problema 0 = forma não compacta (cartão tipo 4) 1 = forma compacta, lê apenas os coeficientes não nulos (cartão tipo 5)	I3
10 à 12	4	Controle da impressão do problema 0 = não imprime 1 = imprime	I3

O problema a resolver deve apresentar-se na forma

$$\text{Max } Z = c^T x \quad \text{sujeito à}$$

$$Ax \leq b$$

$$x \geq 0$$

e será lido segundo o formato do cartão tipo 4 ou cartão tipo 5. As m primeiras linhas contêm, cada uma, $(n + 1)$ coeficientes lidas na ordem

$$a_{i_1}, a_{i_2}, \dots, a_{i_n}, b_i \quad (1 \leq i \leq m)$$

e a última linha lida contêm os coeficientes da função objetiva

$$c_1, c_2, \dots, c_n.$$

CARTÃO TIPO 4

COLUNAS	CAMPO	DESCRIÇÃO	FORMATO
1 à 10	1	Coeficiente da 1 ^a coluna da i -ésima linha do problema	F10.2
11 à 22	2	Coeficiente da 2 ^a coluna da i -ésima linha do problema	F10.2
.....
71 à 80	8	Coeficiente da 8 ^a coluna da i -ésima linha do problema	F10.2

Observação: Se o número de colunas $(n + 1)$ exceder 8, é necessário adicionar mais cartões do tipo 4 para a i -ésima linha. Um segundo cartão tipo 4 contêm os coeficientes a partir da coluna 9, até a coluna 16, no máximo, e assim por diante.

Devem ser colocados cartões tipo 4 desde $i = 1$ até $i = m + 1$. O mesmo vale para cartões tipo 5, caso se deseje a forma compacta.

CARTÃO TIPO 5

COLUNAS	CAMPO	DESCRIÇÃO	FORMATO
1 à 2	1	Coluna do coeficiente não nulo da i -ésima linha	I2
3 à 12	2	Valor do coeficiente a i -ésima linha e coluna especificada no campo anterior	F10.2
13 à 14	3	Idem, campo 1	I2
15 à 24	4	Idem, campo 2	F10.2
25 à 26	5	Idem, campo 1	I2
27 à 36	6	Idem, campo 2	F10.2
.....
63 à 72	12	Idem, campo 2	F10.2

Observação: Em cada cartão tipo 5, cabem até 6 coeficientes não-nulos da i -ésima linha com as suas respectivas colunas no problema. Este cartão deverá ser repetido o número de vezes que se fizer necessário para a linha i . Se o último cartão da linha i estiver com os campos completos, deve ser inserido um cartão em branco.

A N E X O 2
= = = = = =

PROGRAMA COMPUTACIONAL DO ALGORITMO
APROXIMATIVO DE PROGRAMAÇÃO LINEAR

```

DIMENSION VAL(6),ITEMP(6),TITULO(13)
COMMON AB(100,100),X(200),XNCRM(201),KATINJ(2000),KPD(2,200)
COMMON D(100,100),A(100,100),M,N,M1,N1,NM
NI=1
NO=3
KTOT=0
READ(NI,60)NPROB
1 KTOT=KTOT+1
  IF(KTOT-NPROB)2,2,42
2 WRITE(NO,61)KTOT
  READ(NI,62)(TITULO(I),I=1,13)
  WRITE(NO,62)(TITULO(I),I=1,13)
  READ(NI,63)M,N,L,K
  M1=M+1
  N1=N+1
  NM=N+M
  IF(L-1)3,5,5
3 DO 4 I=1,M1
  READ(NI,64)(AB(I,J),J=1,N1)
4 CONTINUE
  GO TO 11
5 DO 10 I=1,M1
  DO 6 J=1,N1
    AB(I,J)=0.0
6 CONTINUE
7 READ(NI,65)(ITEMP(L),VAL(L),L=1,6)
  DO 9 J=1,6
8     L=ITEMP(J)
    AB(I,L)=VAL(J)
9 CONTINUE
  GO TO 7
10 CONTINUE
11 IF(K)14,14,12
12 WRITE(NO,66)M,N
  WRITE(NO,67)(J,J=1,N)
  DO 13 I=1,M1
    WRITE(NO,68)(AB(I,J),J=1,N1)
13 CONTINUE
  WRITE(NO,69)
14 DO 16 I=1,M
  T=0.0
  DO 15 J=1,N
    T=T+AB(I,J)**2
15 CONTINUE
  XNORM(I)=SQRT(T)
16 CONTINUE
  DO 18 J=1,N
  T=0.0
  DO 17 I=1,M
    T=T+AB(I,J)**2
17 CONTINUE
  XNORM(M+J)=SQRT(T)
18 CONTINUE
  T=0.0
  DO 19 J=1,N

```

```

      T=T+AB(M1,J)**2
19 CONTINUE
      DO 20 I=1,M
        T=T+AB(I,N1)**2
20 CONTINUE
      XNORM(NM+1)=SQRT(T)
      DO 21 I=1,NM
        X(I)=0.0
21 CONTINUE
      NM1=NM+1
      NUM=0
      IMPOSS=0
      NRES=0
22 CALL TESTE(II,0.001,DMAIOR,FAT,SINAL,Z,W)
      IF(IND.EQ.0)GO TO 23
      WRITE(NO,76)NUM,NRES
23 IF(II)24,24,25
24 WRITE(NO,70)NUM,NRES,Z
      WRITE(NO,71)(X(I),I=1,N)
      GO TO 1
25 NUM=NUM+1
      IF(NUM.LE.2000)GO TO 99
      WRITE(NO,75)
      GO TO 1
99 KATINJ(NUM)=II
      CALL SIST(II,NUM,LP,LD,NP,ND,IND,NNN)
      IF(IND.EQ.0)GO TO 95
      CALL RESOL(NNN,IMPOSS)
      NRES=NRES+1
      IF(IMPOSS)80,80,95
80 JJ=0
      DO 83 J=1,N
        IF(KPD(IND,J+M))81,82,81
81      JJ=JJ+1
        X(J)=A(JJ,NNN+1)
        GO TO 83
82      X(J)=0.
83 CONTINUE
      NN1=NNN+1
      DO 85 I=1,NN1
        DO 84 J=I,NN1
          TEMP=D(I,J)
          D(I,J)=D(J,I)
          D(J,I)=TEMP
84      CONTINUE
85 CONTINUE
      JJ=0
      CALL RESOL(NNN,IMPOSS)
      DO 88 J=1,M
        IF(KPD(IND,J))86,87,86
86      JJ=JJ+1
        X(J+N)=A(JJ,NN1)
        GO TO 88
87      X(J+N)=0.
88 CONTINUE

```



```

WRITE(NO,71)(X(I),I=1,NM)
GO TO 22
95 IF(NUM-3)31,26,26
26 IF(MARCA)27,27,31
27 IF(KATINJ(NUM)-KATINJ(NUM-2))31,28,31
28 CALL PROJ(FAT,NUM,IMPOSS,MARCA)
IF(IMPOSS)22,22,29
29 WRITE(NO,74)
GO TO 1
31 MARCA=0
IF(II-M)32,32,34
32 Y=DMAIOR/XNORM(II)
DO 33 J=1,N
X(J)=X(J)-Y*AB(II,J)
33 CONTINUE
GO TO 22
34 IF(II-NM)35,35,37
35 Y=DMAIOR/XNORM(II)
DO 36 I=1,M
X(N+I)=X(N+I)+Y*AB(I,II-M)
36 CONTINUE
GO TO 22
37 IF(II-NM-1)38,38,41
38 Y=SINAL*DMAIOR/XNORM(NM+1)
DO 39 J=1,N
X(J)=X(J)-Y*AB(M1,J)
39 CONTINUE
DO 40 I=1,M
X(N+I)=X(N+I)+Y*AB(I,N1)
40 CONTINUE
GO TO 22
41 X(II-1000)=0.0
GO TO 22
42 STOP
60 FORMAT(I2)
61 FORMAT(4(/),10X,'PROBLEMA NUMERO',I4,/,/, 'TITULO',/)
62 FORMAT(13A4)
63 FORMAT(4I3)
64 FORMAT(8F10.2)
65 FORMAT(6(I2,F10.2))
66 FORMAT(/,10X,'NUMERO DE RESTRICCOES=',I4,9X,'NUMERO DE VARIAVEIS=',
*I4,6(/),20X,'MATRIZ DE DADOS')
67 FORMAT(/,10(/,2X,10(9X,I2)))
68 FORMAT(/,10(5X,10(F10.2,1X)))
69 FORMAT(/,120('*'))
70 FORMAT(/,9X,'SOLUCAO OTIMA APOS',I5,3X,'PROJECCOES',9X,'NUMERO DE
*RESOLUCOES DE SISTEMAS=',I3,/,9X,'VALOR DA FUNCAC OBJETIVO=',F15.2
*,/,9X,'VETOR X')
71 FORMAT(/,5X,10F10.4)
73 FORMAT(/,10X,'NUM=',I4,2X,'VICL=',I5,2X,'DIST=',F8.3)
74 FORMAT(/,10X,'O PROBLEMA NAG POSSUI SOLUCAC OTIMA LIMITADA')
75 FORMAT(/,10X,'O NUMERO DE PROJECCOES SUPEROU O LIMITE PREVISTO')
76 FORMAT(/,'PROJ=',I4,3X,'RESOL=',I3)
END
SUBROUTINE TESTE(II,PREC,DMAICR,FAT,SINAL,Z,W)

```

```

COMMON AB(100,100),X(200),XNCRM(201),KATINJ(2000),KPC(2,200)
COMMON D(100,100),A(100,100),M,N,M1,N1,NM
II=0
DMAIOR=0.0
DO 4 I=1,M
  Y=0.0
  DO 1 J=1,N
    Y=Y+AB(I,J)*X(J)
1  CONTINUE
  Y=AB(I,N1)-Y
  IF(Y)2,4,4
2  DIST=-Y/XNORM(I)
  IF(DIST-DMAIOR)4,4,3
3  DMAIOR=DIST
  II=I
  FAT=Y
4  CONTINUE
DO 8 J=1,N
  Y=0.0
  DO 5 I=1,M
    Y=Y+AB(I,J)*X(N+I)
5  CONTINUE
  Y=AB(M1,J)-Y
  IF(Y)8,8,6
6  DIST=Y/XNORM(M+J)
  IF(DIST-DMAIOR)8,8,7
7  DMAIOR=DIST
  II=M+J
  FAT=Y
8  CONTINUE
Z=0.0
DO 9 J=1,N
  Z=Z+AB(M1,J)*X(J)
9  CONTINUE
W=0
DO 10 I=1,M
  W=W+AB(I,N1)*X(N+I)
10 CONTINUE
Y=Z-W
DIST=ABS(Y/XNORM(NM+1))
IF(DIST-DMAIOR)12,12,11
11 DMAIOR=DIST
SINAL=ABS(Y)/Y
II=NM+1
FAT=-Y
12 DO 15 J=1,NM
  IF(X(J))13,15,15
13  DIST=-X(J)
  IF(DIST-DMAIOR)15,15,14
14  DMAIOR=DIST
  II=1000+J
  FAT=-X(J)
15 CONTINUE
  IF(DMAIOR-PREC) 16,17,17
16 II=0

```

```

17 RETURN
END
SUBROUTINE PROJ(FAT,NUM,IMPOSS,MARCA)
COMMON AB(100,100),X(200),XNORM(201),KATINJ(2000),KPC(2,200)
COMMON D(100,100),A(100,100),M,N,M1,N1,NM
DIMENSION C(2,200),NN(2),XX(2)
MARCA=1
DO 2 I=1,2
  DO 1 J=1,NM
    C(I,J)=0.0
1  CONTINUE
2 CONTINUE
NN(1)=KATINJ(NUM)
NN(2)=KATINJ(NUM-1)
DO 13 I=1,2
  IF(NN(I)-M)3,3,5
3  DO 4 J=1,N
    C(I,J)=AB(NN(I),J)
4  CONTINUE
  XX(I)=XNORM(NN(I))**2
  GO TO 13
5  IF(NN(I)-NM)6,6,8
6  DO 7 J=1,M
    C(I,J+N)=AB(J,NN(I)-M)
7  CONTINUE
  XX(I)=XNORM(NN(I))**2
  GO TO 13
8  IF(NN(I)-NM-1)9,9,12
9  DO 10 J=1,N
    C(I,J)=AB(M1,J)
10 CONTINUE
  DO 11 J=1,M
    C(I,J+N)=-AB(J,N1)
11 CONTINUE
  XX(I)=XNORM(NN(I))**2
  GO TO 13
12 C(I,NN(I)-1000)=1.0
  XX(I)=1.0
13 CONTINUE
  AIAJ=0.0
  DO 14 J=1,NM
    AIAJ=AIAJ+C(I,J)*C(2,J)
14 CONTINUE
  DEN=XX(1)*XX(2)-AIAJ**2
  IF(DEN)16,15,16
15 IMPOSS=1
  RETURN
16 IMPOSS=0
  FAT=FAT/DEN
  DO 17 J=1,NM
    X(J)=X(J)+(XX(2)*C(1,J)-AIAJ*C(2,J))*FAT
17 CONTINUE
  RETURN
END
SUBROUTINE SIST(II,NUM,LP,LD,NP,ND,IND,NN)

```

```

COMMON AB(100,100),X(200),XNORM(201),KATINJ(2000),KPD(2,200)
COMMON D(100,100),A(100,100),M,N,M1,N1,NM
IND=0
IF(NUM-NM)27,1,4
1 DO 2 J=1,M
  KPD(1,J)=0
  KPD(2,J)=1
2 CONTINUE
DO 3 J=M1,NM
  KPD(1,J)=1
  KPD(2,J)=0
3 CONTINUE
NNN=0
LP=0
LD=0
4 IF(NNN.NE.0)GO TO 1
  IF(II-M)5,5,7
5 IF(KPD(1,II))27,6,27
6 KPD(1,II)=1
  LP=LP+1
  GO TO 16
7 IF(II-NM)8,8,10
8 IF(KPD(2,II))27,9,27
9 KPD(2,II)=1
  LD=LD+1
  GO TO 16
10 IF(II-1000)27,27,11
11 IF(II-1000-N)12,12,14
12 IF(KPD(1,II-1000+M))13,27,13
13 KPD(1,II-1000+M)=0
  LP=LP+1
  GO TO 16
14 IF(KPD(2,II-1000-N))15,27,15
15 KPD(2,II-1000-N)=0
  LD=LD+1
16 IF(LP-N)18,17,17
17 IND=1
  GO TO 20
18 IF(LD-M)27,19,19
19 IND=2
20 IL=0
  DO 24 I=1,M
    IF(KPD(IND,I))21,24,21
21  IL=IL+1
    JL=0
    DO 23 J=1,N
      IF(KPD(IND,J+M))22,23,22
22  JL=JL+1
      D(IL,JL)=AB(I,J)
23  CONTINUE
      D(IL,JL+1)=AB(I,N1)
24 CONTINUE
  NNN=IL
  JL=0
  DO 26 J=1,N

```

```

        IF(KPD(IND,J+M))25,26,25
25     JL=JL+1
        D(NNN+1,JL)=AB(M1,J)
26     CONTINUE
        NN1=NNN+1
        D(NN1,NN1)=0.
27     RETURN
        END
        SUBROUTINE RESOL(N,IMPOSS)
        COMMON AB(100,100),X(200),XNORM(201),KATINJ(2000),KPC(2,200)
        COMMON D(100,100),A(100,100)
        N1=N+1
        DO 2 I=1,N
            DO 1 J=1,N1
                A(I,J)=D(I,J)
1         CONTINUE
2         CONTINUE
        DO 14 K=1,N
            TEMP=ABS(A(K,K))
            ITEMP=K
            DO 4 I=K,N
                IF(TEMP-ABS(A(I,K)))3,4,4
3             TEMP=ABS(A(I,K))
                ITEMP=I
4             CONTINUE
            IF(TEMP-0.001)5,5,6
5             IMPOSS=1
            RETURN
6             IF(ITEMP-K)7,9,7
7             DO 8 J=1,N1
                TEMP=A(K,J)
                A(K,J)=A(ITEMP,J)
                A(ITEMP,J)=TEMP
8             CONTINUE
9             PIVO=A(K,K)
            DO 10 J=K,N1
                A(K,J)=A(K,J)/PIVO
10            CONTINUE
            DO 13 I=1,N
                IF(I-K)11,13,11
11             TEMP=A(I,K)
                DO 12 J=K,N1
                    A(I,J)=A(I,J)-TEMP*A(K,J)
12            CONTINUE
13            CONTINUE
14            CONTINUE
            RETURN
        END

```

A N E X O 3
= = = = =

LISTAGEM DE SAÍDA DO PROGRAMA

MAIRIZ DE DADOS

1	2	3	4	5	6
-15.00	36.00	12.00	-36.00	66.00	84.00
-49.00	95.00	21.00	74.00	-50.00	15.00
-11.00	51.00	-5.00	-90.00	-28.00	-57.00
30.00	90.00	-45.00	56.00	-59.00	24.00
58.00	4.00	54.00	-38.00	93.00	-34.00
52.00	-42.00	-40.00	-3.00	87.00	-34.00
6.00	30.00	-93.00	10.00	0.00	50.00
-95.00	65.00	42.00	65.00	-98.00	3.00
0.00	20.00	-55.00	50.00	34.00	-90.00
-13.00	-10.00	-35.00	-32.00	-50.00	88.00
5.00	45.00	38.00	-30.00	-23.00	16.00
-43.00	-30.00	59.00	54.00	-61.00	90.00
-5907.00	27782.00	-3074.00	-2453.00	-14526.00	11835.00

PROJ= 0 RESOL= 0 Z= 0.0 K= C.0
 75.8827 96.8093 57.2515 14.5206 15.6053 31.1122 -198.1512 C.0 179.1684 0.0
 C.0 0.0 628.1487 191.4672 -564.3572 -500.5297 C.0 0.0
 PROJ= 39 RESOL= 1 Z= 2089050.00 K= 2089050.00
 88.2333 100.4428 60.1541 14.2810 13.9941 22.8969 31.6130 141.1786 80.8485 0.0
 C.0 0.0 105.3185 0.0 0.0 40.6782 141.2424 C.0
 PROJ= 96 RESOL= 2 Z= 2141586.00 K= 2141594.00

SOLUCAC OTIMA APOS 96 PROJECES NUMERO DE RESOLUCES DE SISTEMAS= 2
 VALOR DA FUNCAC OBJETIVO= 2141586.00
 VETOR X

88.2333 100.4428 60.1541 14.2810 13.9941 22.8969