

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELETRICA  
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

Este exemplar corresponde à redação final da tese  
defendida por Maria Amelia Biagio

e aprovada pela Comissão  
Julgadora em 22 / 02 / 91.

Secundino Soares F.  
Orientador

ESTUDOS SOBRE MÉTODOS DE DECOMPOSIÇÃO VIA DUALIDADE NA  
RESOLUÇÃO DE PROBLEMAS LINEARES DINÂMICOS

AUTORA : MARIA AMELIA BIAGIO

ORIENTADOR: PROF. DR. SECUNDINO SOARES FILHO

Tese apresentada pela Faculdade de Engenharia Elétrica - FEE/  
UNICAMP como requisito parcial para obtenção do título de Doutora  
em Engenharia Elétrica.

fevereiro / 1991

Aos meus pais,  
Egídio (in memoriam)  
e Maria

Ao Pablo, compa-  
nheiro de todas as horas

Àqueles que acreditam e  
lutam pelo desenvolvimento da pes-  
quisa no Brasil e no mundo.

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

A elaboração deste trabalho contou com a participação de amigos e colegas de profissão que, de diversas formas, contribuíram para a conclusão do mesmo. Muitos foram os amigos que registraram sua colaboração pela presença agradável e indispensável durante todos esses anos. Colegas e professores das disciplinas de pós-graduação da Faculdade de Engenharia Elétrica marcaram sua colaboração antecipada através das discussões e estudos sobre Otimização de Sistemas. A dinâmica das discussões, o espírito de renovação, juntamente com a prontidão dos muitos colegas em ajudar no saneamento das dúvidas surgidas, encontrados no cotidiano da Universidade Estadual de Campinas, exerceram papel fundamental para o empreendimento deste trabalho.

Contou com presenças constantes como a do Professor Secundino, seu mentor e orientador, que depositou confiança em seu êxito e a de Pablo, colega e amigo, nas discussões sobre os diversos problemas surgidos. A presença do Akebo foi bastante importante por sua sempre disponibilidade em ajudar nas dúvidas dos trabalhos computacionais, tanto de "hardware" como de "software".

Dispuseram-se para discussões inerentes ao trabalho Celso e Raimundo, colegas do Departamento de Matemática da Universidade Federal do Paraná, Ana Friedlander e José Mario Martínez, colegas do Departamento de Matemática Aplicada do IMECC/Unicamp, e o colega Eurice, do Departamento de Engenharia Elétrica da UnB.

Contou com a preciosa presença amiga do Takaaki, Anésio, Kátia, Berenice, Evelin, Sueli e Toinho na sua fase final em Campinas.

Este trabalho teve sua parte computacional elaborada e testada em computador DEC-System 10, da Universidade Federal do Paraná, onde sua autora foi professora até março de 1990, e contou com o apoio técnico do Hamilton, Analista de Sistemas do Setor de Processamento de Dados daquela Universidade. A digitação de seu texto contou com a participação preciosa do Pablo e a colaboração indispensável do Departamento de Engenharia Elétrica da Universidade de Brasília, que concedeu a utilização de suas instalações para tal finalidade.

Este trabalho contou com o apoio financeiro da FAPESP .

## R E S U M O

Este trabalho versa sobre dois métodos de resolução para problemas lineares dinâmicos baseados em dualidade e que exploram possibilidades de decomposição temporal para esses problemas.

O primeiro, o Método do Lagrangeano Aumentado, obtém simultaneamente as soluções ótimas dual e primal através da adição de um termo de penalidade quadrático ao Lagrangeano simples. A solução do Lagrangeano Aumentado é adaptada idéia utilizada para o caso não-linear para obtenção da separabilidade temporal do problema.

O segundo, Decomposição via Lagrangeano Simples, determina a solução ótima dual via Aproximação Tangencial, explorando a decomposição temporal do Lagrangeano simples. A partir daí, a solução ótima primal é obtida através da resolução de subsistemas lineares sequenciais em decorrência da aplicação do Princípio de Otimalidade de Bellman.

Ambos os métodos são analisados, implementados e resultados computacionais são apresentados.

## A B S T R A C T

This work is concerned with two methods, based on duality, for solving linear dynamic problems which exploit temporal decomposition.

The first, the Augmented Lagrangean Method, obtains simultaneously both dual and primal optimal primal solutions by adding a quadratic penalty term to the ordinary Lagrangean. In order to obtain temporal decomposition of the problem, an idea from the nonlinear case is adapted.

The second, the Decomposition via Ordinary Lagrangean, determine the dual optimal solution via Tangencial Approximation, exploiting the temporal decomposition of the simple Lagrangean. Then, the primal optimal solution is obtained throught the resolution of sequential linear subsystems according to the application of Bellman's Optimality Principle.

Both methods are analysed, implemented and computacional results are presented.

# I N D I C E

CAPÍTULO I : INTRODUÇÃO . . . . .	1
CAPÍTULO II : O MÉTODO DO LAGRANGEANO AUMENTADO	
II.1 - INTRODUÇÃO . . . . .	18
II.2 - APRESENTAÇÃO DO (P.L.D.) . . . . .	19
II.3 - O PROBLEMA DUAL DO (P.L.D.) . . . . .	20
II.3.1 - A Função Dual . . . . .	20
II.3.2 - Interpretação Geométrica . . . . .	23
II.4 - O MÉTODO DO LAGRANGEANO AUMENTADO - RESOLUÇÃO DO (P.L.D.) . . . . .	27
II.4.1 - Interpretação Geométrica . . . . .	31
II.4.2 - O Algoritmo do Lagrangeano Aumentado . . . . .	35
II.5 - DECOMPOSIÇÃO DO PROBLEMA . . . . .	36
II.5.1 - Diferenciabilidade de (FV) . . . . .	42
CAPÍTULO III : IMPLEMENTAÇÃO DO LAGRANGEANO AUMENTADO - TESTES E CONCLUSÕES	
III.1 - INTRODUÇÃO . . . . .	45
III.2 - O MÉTODO DO GRADIENTE CONJUGADO . . . . .	46



III.3 - IMPLEMENTAÇÃO DO "1 <sup>o</sup> NÍVEL" . . . . .	53
III.3.1 - Algoritmo para Gradientes Conjugados (C-G-C) . . . . .	53
III.4 - IMPLEMENTAÇÃO DO "2 <sup>o</sup> NÍVEL" . . . . .	57
III.4.1 - Atualização com Busca Unidimensional - Utilização de Gradientes Conjugados . . . . .	59
III.4.2 - Algoritmo (C - G) . . . . .	59
III.5 - TESTES COMPUTACIONAIS . . . . .	62
III.5.1 - Problemas - Teste . . . . .	62
III.5.2 - Resultados Computacionais . . . . .	66
III.6 - CONCLUSÕES . . . . .	72

CAPÍTULO IV : DECOMPOSIÇÃO VIA LAGRANGEANO SIMPLES

IV.1 - INTRODUÇÃO . . . . .	77
IV.2 - MÉTODO DA APROXIMAÇÃO TANGENCIAL . . . . .	78
IV.2.1 - Adaptação para (P.L.D.) . . . . .	84
IV.2.2 - Geração das Restrições . . . . .	87
IV.2.3 - Inicialização do Processo . . . . .	89
IV.2.4 - A Técnica da Relaxação . . . . .	91
IV.3 - PROCEDIMENTO PARA OBTENÇÃO DA SOLUÇÃO ÓTIMA PRIMAL . . . . .	94
IV.3.1 - Preliminares sobre Programação Linear . . . . .	95
IV.3.2 - Obtenção das Variáveis Não-Básicas da Solução ótima de um (P.L.D.) . . . . .	98
IV.3.3 - Obtenção da Solução ótima Primal para (P.L.D.) . . . . .	99
IV.3.4 - Características do Sistema a ser Resolvido na "2 <sup>a</sup> FASE" . . . . .	100
IV.3.5 - Procedimento Iterativo . . . . .	111

CAPÍTULO V : IMPLEMENTAÇÃO DA DECOMPOSIÇÃO VIA LAGRANGEANO SIMPLES		
- TESTES E CONCLUSÕES		
V.1	- INTRODUÇÃO . . . . .	113
V.2	- IMPLEMENTAÇÃO DA "1ª FASE" . . . . .	114
V.2.1	- Algoritmo Dual-Simplex Canalizado . . . . .	115
V.2.2	- Relaxação em $(PM_{LD})^r$ . . . . .	128
V.3	- RESOLUÇÃO DA "2ª FASE" . . . . .	131
V.3.1	- Estimativa sobre a Ocorrência de Não - Degenerescência em $(P_{T-k})$ . . . . .	134
V.4	- TESTES COMPUTACIONAIS . . . . .	136
V.4.1	- Resultados Computacionais . . . . .	138
V.5	- CONCLUSÕES . . . . .	143
CAPÍTULO VI : CONCLUSÕES . . . . .		147
APÊNDICE 1	. . . . .	153
APÊNDICE 2	. . . . .	155
APÊNDICE 3	. . . . .	160
APÊNDICE 4	. . . . .	167
BIBLIOGRAFIA . . . . .		173

## CAPITULO I

### INTRODUCAO

Um problema de programação linear dinâmico (P.L.D) pode ser apresentado na forma

$$\begin{aligned} \text{(FO)} \quad & \text{Minimizar } \sum_{t=0}^{T-1} c(t+1) x(t+1) + d(t) u(t) \\ \text{s. a.} \quad & x(t+1) = A(t)x(t) + B(t)u(t) \quad (1) \\ & f(t) = C(t)x(t) + D(t)u(t) \quad (2) \\ & x(0) = x^0 \quad (3) \\ & x(t+1), u(t) \in S, t=0, \dots, T-1 \end{aligned}$$

onde  $S$  é um conjunto,  $x(t)$  representa a variável de estado e  $u(t)$  variável de controle no intervalo  $t$ . A equação (1) nos diz que o estado do sistema num determinado tempo depende do estado e de seu controle no tempo anterior; a equação (2) sugere outras restrições para o problema, caso existirem. A equação (3) fornece o estado inicial do sistema e (FO) a função objetivo do problema. O horizonte de otimização é dado por  $T$  intervalos de tempo.

Naturalmente, este problema pode ser visto como um problema linear cuja matriz de restrições é dada por

$$\begin{bmatrix}
 D(0) \\
 B(0) - I \\
 C(1) \quad D(1) \\
 A(1) \quad B(1) - I \\
 \cdot \\
 \cdot \\
 \cdot \\
 C(T-1) \quad D(T-1) \\
 A(T-1) \quad B(T-1) - I
 \end{bmatrix}$$

onde  $I$  representa a matriz identidade de ordem  $n$  e, como um problema linear, pode ser resolvido pelo método clássico do Simplex Revisado.

Todavia, esta matriz possui características bastante particulares: uma delas, a mais importante, é a sua estrutura apresentada em forma de uma "escada". Essa particularidade, aliada ao fato de que um problema linear dinâmico pode apresentar-se como um problema de grande porte devido a grande dimensão dos vetores de estado e controle e longo horizonte de otimização, faz com que esta matriz seja um forte atrativo para estudiosos que se interessam em apresentar técnicas de resolução para problemas de grande porte.

Dadas as proporções que a matriz de um (P.L.D.) pode assumir, a utilização simples do método do Simplex acarretaria um gasto excessivo de memória e tempo de operação.

A partir daí, surgiram duas linhas de pesquisa bastante distintas que procuram explorar, naturalmente, as características de um (P.L.D.): de um lado os métodos que, baseados no simplex revisado, exploram a estrutura escada da matriz de restrições do problema e que são aqui denominados metodos diretos (segundo LASDON (1970)) e, por outro lado, os métodos que buscam a separabilidade temporal do problema, obtendo subproblemas de porte pequeno e de fácil resolução aqui denominados, por sua vez, como

## metodos indiretos.

Ambas as linhas de pesquisa se preocupam em propor técnicas que viabilizem economia de memória e/ou gasto computacional (tempo de processamento).

Citaremos a seguir alguns desses estudos realizados até o momento.

### Métodos Diretos:

Entre os métodos diretos, um dos temas bastante explorado foi o de substituir o conceito de base global do método Simplex (ou seja, uma matriz básica para a matriz de restrições completa) pelo conceito de bases locais, que associa a cada intervalo de tempo uma base de dimensão menor. O primeiro trabalho divulgado foi de Krivonozhko e Chebotarev (1976), os quais desenvolveram estudos para problemas com variáveis de estado livres; em seguida, Propoi e Krivonozhko (1978) reduziram o número de bases locais para tratar o mesmo problema. Esta mesma abordagem, aperfeiçoada para problemas com variáveis de estado não negativas, foi desenvolvida por Wollmer (1977) e difere do método de Krivonozhko no processo de obtenção e atualização das bases locais.

As bases locais no método de Krivonozhko - Chebotarev - Propoi são obtidas através da decomposição da matriz básica num produto de duas matrizes, sendo a primeira uma matriz bloco triangular inferior contendo matrizes quadradas na sua diagonal principal (as bases locais) e a segunda uma matriz bloco triangular superior com elementos unitários na diagonal principal. Sempre que ocorre uma troca de colunas (básica por uma não-básica) algumas bases locais precisam ser atualizadas e é neste trabalho de atualização que se concentra o maior gasto computacional. Neste caso a troca de variáveis entre bases adjacentes, quando possível, é mais eficiente que outros esquemas de troca.

A "FASE I" desse método é análoga à do Simplex clássico ao

introduzir variáveis artificiais e utiliza também a metodologia das bases locais .

A fase de preparação da matriz para obtenção das bases locais é bastante trabalhosa , envolvendo várias multiplicações de matrizes e  $(2T-1)$  escolhas de colunas linearmente independentes , o qual geralmente é feito através do processo de eliminação de Gauss. Mas, este trabalho pode ser compensado pela economia de operações e de memória na FASE II , fase de obtenção da solução ótima do problema .

Na metodologia de Wollmer , as bases locais são contruídas através de arranjos adequados de colunas e pivoteamentos, apresentando um resultado análogo ao do método anterior .

Entretanto, ambos os procedimentos causam quebra da estrutura escada da matriz de restrições com o aparecimento de elementos não nulos substituindo alguns zeros durante o processo de decomposição.

Testes realizados mostraram que o método proposto por Krivonozhko - Propoi pode apresentar resultados melhores quando comparado ao método Simplex Revisado .

Muitos foram os trabalhos desenvolvidos para programação linear de grande porte que tiveram como meta a especialização do método do Simplex Revisado aproveitando a esparsidade da matriz de restrições através da fatoração L-U no armazenamento da matriz base ( Lasdon , 1968 ) e que obtiveram resultados bastante satisfatórios .

Sendo uma submatriz de uma matriz esparsa , a matriz base de um (P.L.D.) também deve ser esparsa . Como o método Simplex requer, a cada iteração, a atualização da coluna não - básica escolhida para entrar na base e a atualização do vetor independente, o custo computacional por iteração é igual, no mínimo, ao custo de resolver dois sistemas lineares .

Foram baseados neste fato que alguns estudiosos resolveram seguir a linha de pesquisa voltada para a resolução de sistemas lineares de grande porte com matriz de restrições esparsa , com o

propósito de diminuir o trabalho do Simplex Revisado a cada iteração .

Este caminho já vinha sendo aberto através de trabalhos importantes para a resolução de sistemas esparsos. Um deles é o trabalho de Forsythe - Moler (1967) , o qual utiliza a fatoração L-U no processo de eliminação de Gauss .

Ao aplicarmos o método de eliminação de Gauss para sistemas com matrizes esparsas , é de certa forma natural pensarmos em reordenar as linhas e as colunas da matriz de tal forma a obtermos uma matriz , equivalente à primeira, tão próxima quanto possível a uma forma triangular (superior ou inferior ) com o propósito de acelerarmos o processo de resolução do sistema . Hellerman e Rarick ( 1971 , 1972 ) utilizaram este procedimento antes de aplicarem a fatoração na matriz, obtendo bons resultados, porém, causando o aparecimento de novos elementos não nulos na matriz " U " da fatoração . Saunders ( 1976 ) especializou esta técnica predizendo a região da matriz " U " na qual são criados tais elementos não nulos .

Recentemente temos o trabalho de tese de doutoramento de Friedlander (1986) no qual a autora descreve os trabalhos desenvolvidos nesta linha . Em seu trabalho, Friedlander introduz uma nova técnica de atualização das fatorações L-U da matriz escada de um (P.L.D.). Essa fatoração é feita de tal modo que os fatores L e U mantem a estrutura escada da matriz de restrições facilitando, dessa forma, um melhor aproveitamento dessa estrutura na resolução dos sistemas lineares em cada iteração do Simplex . Neste trabalho existe um compromisso entre a estabilidade numérica do algoritmo e preservação da estrutura escada da matriz.

Devemos ainda destacar, entre os métodos diretos, o recente trabalho de Tese de Mestrado de Oliveira ( 1989 ) o qual mostra o eficiente desempenho dos Métodos de Pontos Interiores na resolução de problemas lineares cuja matriz de restrições é esparsa.

## Métodos Indiretos :

Muitos outros métodos surgiram propondo novas formas de resolução de um problema linear dinâmico, segundo levantamento feito por Propoi (1976). Dentre eles estão aqueles aqui denominados como métodos indiretos, dos quais destacamos os métodos de decomposição em cascata , do Gradiente Projetado e , em particular, dos Multiplicadores de Hestenes e Powell. Como já assinalamos, estes métodos procuram explorar a decomposição temporal do problema visando a obtenção de subproblemas de porte menor e cuja resolução pode ser obtida através de algoritmos existentes na literatura da programação matemática .

O método da decomposição em cascata baseia-se na aplicação sucessiva do princípio de decomposição de Dantzig-Wolfe a cada intervalo de tempo, o qual age como " problema mestre " para os intervalos seguintes e como subproblema para o intervalo anterior a ele.

A técnica da decomposição de Dantzig-Wolfe [14] é bastante conhecida, sendo eficiente para tratar problemas cuja matriz de restrições apresenta uma ou mais equações acoplando as variáveis do problema .

Notamos que na matriz de restrições de um (P.L.D.) existem várias equações de estado acoplando dois consecutivos vetores de estado por vez, o que sugere a aplicação do princípio de Dantzig-Wolfe tantas vezes quantas forem as equações de estado. Dessa forma, cria-se uma estrutura multinível para o algoritmo, onde cada subproblema é um problema linear estático situado num nível e que é resolvido através do algoritmo do Simplex Revisado. A cada nível uma solução ótima é obtida para o subproblema em questão, o qual envia novas informações para os níveis adjacentes.

Este método apresenta uma estrutura complicada de vários níveis interdependentes. É um método primal e necessita de uma



solução básica factível para ser inicializado, sendo obrigatório, para isso, a resolução de uma "FASE I". Nesta fase podemos aplicar a mesma metodologia da decomposição em cascata incluindo variáveis artificiais ao problema.

Seguindo esta linha encontramos os trabalhos de Glassey (1973) e Ho e Manne (1974). Aplicando esta mesma metodologia a problemas lineares dinâmicos com variáveis de estado livres está o trabalho de Krivonozhko (1977) [35]. Uma aplicação deste método a sistemas com atrasos distribuídos nas variáveis de estado e de controle é encontrado no trabalho de Tamura (1977) [60].

No entanto, as experiências computacionais relatadas por Glassey [26] e Ho e Manne [31] não foram animadoras quando comparados o seu gasto de operações com o gasto do Simplex Revisado. Mas, em seu trabalho de tese de doutoramento, Ho conseguiu, através da aplicação de técnicas especiais, reduzir o gasto computacional do método da decomposição em cascata para aquém do tempo gasto pelo Simplex Revisado. Esses autores concluíram que o método da Decomposição traz vantagens somente quando não dispomos de computador suficientemente grande para tratar o problema (P.L.D.), apresentando uma economia de memória 2,5 vezes menor do que a necessária para o método do Simplex Revisado.

Distintamente, a técnica proposta por Tamura [60] é estendida para problemas lineares dinâmicos com atrasos distribuídos, os quais apresentam os subproblemas e suas interconecções de natureza mais complicada que os demais. A grande vantagem da técnica utilizada por Tamura é que ele soluciona facilmente o sistema de equações sem reduzi-lo a um sistema equivalente com "equação de estado" de primeira ordem e, conseqüentemente, um número muito maior de variáveis, obtendo, dessa forma, redução da carga computacional. Nesta técnica encontramos o estágio final como problema-mestre, os estágios intermediários como subproblemas-mestre e o estágio inicial como subproblema.

Já o algoritmo proposto por Glassey designa o estágio inicial como problema-mestre, os estágios intermediários como subproblemas-mestre e o estágio final como subproblema. As

trocas de informações entre os vários níveis é executada somente entre níveis adjacentes ; dessa forma , a técnica de Glassey não pode ser aplicada diretamente a modelos com atrasos distribuídos

Recentemente encontramos os trabalhos de Ho , Lee e Sundarval [30], (1988), onde descrevem implementação do algoritmo de Decomposição de Dantzig-Wolfe para sistemas lineares com matriz bloco-angular usando processamento paralelo nos subproblemas , com o qual obtiveram bons resultados computacionais .

Também recentemente encontramos o trabalho de Jackson e Lynch [32] (1987) no qual os autores propõem a decomposição de Dantzig-Wolfe " revisada " para problemas lineares com matriz bloco-angular . Neste trabalho os autores propõem uma maneira de seleccionar colunas para acelerar a resolução dos problemas-mestre. Apesar dos bons resultados computacionais obtidos , os autores acreditam que a técnica da decomposição de Dantzig-Wolfe deve ser reservada para problemas cujo tamanho ultrapassa os limites do "Software " básico do método Simplex .

O método do Gradiente Projetado aplicado a problemas lineares dinâmicos baseia-se no método clássico conhecido por esse mesmo nome. O método é do tipo primal e a trajetória em busca da solução ótima é feita através de direcções factíveis que são determinadas pela projeção de gradientes no subespaço das restrições ativas . Aqui o carácter de decomposição do método verifica-se no cálculo da projeção do gradiente que é feito através de uma decomposição dual (a função Lagrangeano do problema de projeção é separável por intervalos de tempo ) obtendo-se soluções analíticas para cada componente da projeção . O tamanho do passo na otimização é calculado com muita facilidade para problemas lineares .

Em seu trabalho de tese de mestrado , Armentano (1979) faz uma adaptação analítica deste método à programação linear dinâmica. Todavia, resultados computacionais deste estudo ainda não são conhecidos.

Sabemos que o método do Gradiente Projetado aplicado a

problemas lineares com variáveis " canalizadas " ( variáveis com limite superior e inferior ) pode ser visto como o método Simplex quando projeta o gradiente no espaço das variáveis que estão nos seus limites ( neste caso , supõe-se que o problema não possui soluções degeneradas ), ou seja, no espaço das variáveis não-básicas, e podemos garantir que , sob determinadas condições ( vide Lasdon [38] ), os dois processos são equivalentes a cada iteração . Portanto, podemos prever sobre o comportamento do método quando utilizado na resolução de problemas (P.L.D.) que, semelhantemente ao Simplex, pode apresentar problemas de memória devido à necessidade de armazenamento da matriz de restrições.

O método dos multiplicadores de Hestenes e Powell, assim comumente conhecido, será aqui batizado como o método do Lagrangeano Aumentado. Este método consiste em adicionar à função penalidade de um problema com restrições um termo linear que é o produto do vetor multiplicador de Lagrange pela matriz das restrições do problema.

Esta nova metodologia de abordar problemas com restrições surgiu com os trabalhos de Hestenes (1969 ) e Powell (1969), nos quais os autores procuravam, de maneiras distintas, contornar o problema de estabilidade numérica que é encontrado no método de penalização ( vide Luenberger [39] ) quando encontramos necessidade de acelerar o processo iterativo aumentando demasiadamente o parâmetro de penalidade .

As considerações levantadas por Powell sobre a convergência do novo método despertaram a atenção de muitos estudiosos. Conseqüentemente, muitos trabalhos importantes explorando a potencialidade do método surgiram . Dentre eles devemos destacar os trabalhos feitos por Rockafellar (1973,1974) e Bertsekas (1975,1976) .

Rockafellar apresenta um extenso estudo relacionando a teoria de dualidade com o método do Lagrangeano Aumentado e prova que o método , no caso convexo , converge para qualquer fator de

penalização sem a exigência de que um mínimo exato na variável primal deve ser calculado para cada atualização do multiplicador [ 51 ].

Bertsekas faz novos estudos e propõe novas técnicas para aplicação a problemas convexos . Dentre esses estudos, apresenta um resultado sobre a taxa de convergência do algoritmo que relaciona multiplicadores e fator de penalização : ou seja, o valor atribuído a este fator influencia na rapidez com que o vetor multiplicador pode chegar ao seu valor ótimo [ 6 ].

O método do Lagrangeano Aumentado é visto como um método dual, dado que somente aos valores ótimos das variáveis duais é que temos associada uma solução factível primal, que também deve ser a ótima. Ao atualizarmos o valor do multiplicador de Lagrange, estamos dando " um passo " no sentido de otimizar a função dual relacionada ao problema em questão, que sabemos ser uma função côncava.

Uma análise sobre a matriz hessiana da função dual ( vide Luenberger [39] ) mostra que a razão dos autovalores dessa matriz é próxima da unidade quando atribuímos valores suficientemente grandes ao fator de penalidade da função do Lagrangeano Aumentado, o que confere uma taxa de convergência extremamente alta para o método sob o ponto de vista dual .

Esse método também foi alvo de interesses para a resolução de problemas não-convexos . Bertsekas [9], por sua vez, propôs um método mais global, apoiado no método dos multiplicadores e no método clássico da penalização , provando a sua convergência .

Nos problemas não convexos em geral , é comum o aparecimento de "gaps " de dualidade . Em estudos realizados, o método do Lagrangeano Aumentado mostrou ser uma técnica poderosa para resolver problemas com " gaps " , o que seria impossível de ser realizado com a aplicação do método do Lagrangeano ordinário.

Uma interpretação geométrica do problema do " gap " pode ser feita através da " função de perturbação " com ilustração do efeito convexificante do termo quadrático da função do Lagrangeano Aumentado sobre a função de perturbação original. Encontramos

esses estudos nos trabalhos de Stephanopoulos e Westerbeg (1975) e Watanabe e outros (1978), este último inserindo uma nova interpretação geométrica, do ponto de vista dual do método, através de hiperbolóides-suporte à função de perturbação .

Já em 1974, encontramos o trabalho de Rockafellar [49] estendendo seus estudos para problemas não-convexos; neste trabalho o autor, ao contrário do problema clássico, permite, a priori, que as variáveis duais correspondentes às equações de desigualdade do problema primal assumam quaisquer valores.

Muitos outros trabalhos apareceram com o propósito de apresentar melhorias à técnica do Lagrangeano Aumentado. Entre eles citamos os trabalhos de Miele e outros (1971,1972) [41,42] onde propõem novas formas de atualização para multiplicador e fator de penalidade , analisando propriedades de convergência do algoritmo para diferentes implementações para o problema do ponto de vista primal. Com este enfoque também encontramos o trabalho de Glad (1979) [25].

Várias foram as aplicações deste método na resolução de sistemas de controle. Tripathi e Narendra (1972) propõem uma mudança no método dos multiplicadores e descrevem uma aplicação a problemas de controle ótimo. Glad (1979) [24] mostra que o menor valor para a constante de penalização, necessário para chegarmos à solução ótima de um problema de controle contínuo no tempo, cujas funções são três vezes continuamente diferenciáveis, é obtido através da equação de Riccati e dá a taxa de convergência para o algoritmo do ponto de vista dual .

Uma aplicação do método do Lagrangeano Aumentado na solução de problemas de fluxo de carga ótimo em redes de energia elétrica pode ser encontrado nos trabalhos de tese de doutoramento de Santos (1985) [52], onde bons resultados foram obtidos .

O método do Lagrangeano Aumentado não só teve a sua convergência e as suas propriedades demonstradas para problemas de programação não-linear . Em 1975 , Propoi e Yadykin [45] publicam seus estudos sobre o método aplicado na resolução de problemas

lineares dinâmicos , onde apresentam propriedades importantes que garantem a convergência do método.

A priori é bastante audacioso pensarmos em aplicar o método do Lagrangeano Aumentado na resolução de um (P.L.D.), visto que o mesmo destrói a característica linear do problema, gerando uma função Lagrangeano não decomponível com inúmeras variáveis devido à dinâmica da estrutura do problema e, assim, complicando o processo de sua otimização .

Todavia, dadas as garantias teóricas dos estudos feitos por Propoi e Yadykin sobre o bom desempenho do método ( taxa de convergência rápida para o problema do ponto de vista dual ), propostas foram elaboradas ( Armentano , 1979 ) no sentido de viabilizarmos a separabilidade temporal da função Lagrangeano Aumentado através da aplicação da técnica de Watanabe (1978) . Com essa separabilidade simplificamos o problema de minimizar uma função com muitas variáveis , criando vários problemas menores e de fácil resolução a cada estágio de tempo . Entretanto, os resultados computacionais desta técnica aplicada à (P.L.D.) ainda não são conhecidos .

A técnica do Lagrangeano Aumentado aplicado à (P.L.D.) nos chamou a atenção por seus vários atrativos : primeiro , a do fato de que a decomposição temporal da função Lagrangeano Aumentado (L.A.) certamente nos proporciona uma grande economia de memória e, mais do que isso, nos fornece, a cada estágio de tempo, subproblemas de fácil resolução ; segundo, dado o caráter dual do método, não necessitamos da já conhecida " FASE I" para inicializarmos o processo iterativo do algoritmo, como necessitam os demais métodos aqui enunciados. Aliado a esses dois fatores muito importantes, encontramos ainda garantias sobre a diferenciabilidade e a continuidade da função dual obtida através dessa técnica .

A possibilidade de empregarmos uma nova técnica de decomposição para a resolução de problemas do tipo (P.L.D.) foi trazida através de uma importante característica encontrada nas soluções ótimas desses problemas, qual seja: o Princípio de Otimalidade de Bellman, válido para problemas dinâmicos em geral, indica que, na presença de não-degenerescência, as variáveis básicas da solução ótima de problemas (P.L.D.) devem distribuir-se igualmente pelos intervalos de tempo sugerindo que, conhecidos os valores das variáveis não-básicas, as variáveis básicas podem ser facilmente calculadas através de "subsistemas" que interagem de forma recursiva.

Esta idéia vem ao encontro do fato de que, na programação linear, com a obtenção do multiplicador ótimo de Lagrange, conseguimos determinar os valores ótimos das variáveis não-básicas primais através de simples inspeção nos coeficientes da função Lagrangeana. Com isso, obtemos facilmente as colunas não-básicas e, conseqüentemente, as colunas básicas da matriz de restrições do problema na solução ótima primal.

Estes dois fatores somados indicam que, obtida a solução ótima dual de um (P.L.D.), a obtenção da solução ótima primal pode se dar de maneira eficiente em termos computacionais através da resolução de um sistema linear que pode ser resolvido "por partes".

A técnica assim sugerida não é uma técnica primal e necessita de uma "1ª FASE" que trata da obtenção do multiplicador ótimo do problema. Dessa forma, a nova técnica deve ser composta de duas "FASES": 1ª) obtenção do multiplicador ótimo de Lagrange; 2ª) resolução do sistema.

Sendo a "2ª FASE" bastante simples de ser resolvida computacionalmente, requerendo apenas um rearranjo de colunas (reconhecimento das colunas básicas e não-básicas) e um indicador da localização dos "subsistemas" na matriz de restrições do problema, vamos tecer alguns comentários acerca das técnicas a serem empregadas para a resolução da "1ª FASE" dada a sua carga

computacional apresentar-se significativa para o procedimento proposto.

Dado que a função dual de um problema linear é côncava linear-por-partes, ou seja, não diferenciável, algumas técnicas são sugeridas para a busca de sua solução ótima. Dentre elas podemos citar o método dos subgradientes e o método da Aproximação Tangencial, os quais se adaptam às características da função.

Bazaraa e Goode [ 2 ] discutem especialmente métodos que utilizam direções de gradiente e subgradiente na resolução do problema dual quando a Relaxação Lagrangeana é empregada em problemas de forma geral. Mostram a relação existente entre direções de gradiente e subgradiente sob o aspecto de várias normas.

Sabemos que, em geral, o método de direções de gradiente não converge para a solução ótima se a função a ser otimizada é não diferenciável. No entanto, através de uma restrição no "tamanho do passo" na busca unidimensional, Bazaraa e Goode mostram que o método converge para o caso da função ser linear-por-partes.

Por outro lado, sabemos que o método das direções do gradiente possui convergência extremamente lenta, principalmente quando trata de funções "mal comportadas", e uma busca unidimensional em funções não definidas explicitamente requer bastante esforço computacional.

O método da Aproximação Tangencial ( ou métodos de Planos-de-Corte [33] ) em sua forma dual é equivalente ao Método de Decomposição de Dantzig-Wolfe. Apresentada por Geoffrion (1967), a versão do método consiste na combinação de Dualização e Linearização Externa como manipulações do problema. Aplicado na resolução de problemas lineares, o método obtém uma aproximação da função dual através da geração de hiperplanos-suporte, resultando na obtenção da solução ótima dual do problema. Sua convergência foi estudada por Kelley (1960), em cujo trabalho o autor tece algumas considerações quanto ao problema de precisão computacional do método quando o mesmo está próximo da solução. Isto deve-se ao



fato de que, quando próximo ao ponto crítico, o método pode gerar sucessivos hiperplanos à função dual geralmente com tendência a serem cada vez mais paralelos uns aos outros. Para sanar esta dificuldade, Kelley propõe o uso de um parâmetro de tolerância para a parada de seu algoritmo .

Para problemas lineares, o fato da função dual ser linear-por-partes traz vantagens no sentido de ser menor o risco do método gerar hiperplanos " numericamente paralelos " na vizinhança da solução, com exceção aos casos em que a função dual apresenta-se muito " suave ". Visto ser côncava linear-por-partes e finita, a função dual pode ser representada por um número finito de hiperplanos-suporte, podendo-se, dessa forma, garantir teoricamente que o método converge em um número finito de passos.

Para casos de problemas não-convexos que apresentam " gap " de dualidade, o método da Aproximação Tangencial não é aplicável , visto que a determinação do multiplicador ótimo não levará à solução primal, sendo, entretanto, aplicável em geral para problemas convexos.

A implementação deste método pode ser feita de maneira eficiente dada a existência de muitas rotinas e muitos algoritmos na literatura da programação linear e que servem como apoio para a maior tarefa computacional ( resolução do problema-mestre ) do método. No entanto, dado o número considerável de hiperplanos a serem gerados quando problemas de porte maior são resolvidos, estratégias devem ser elaboradas no sentido de minimizar os gastos operacionais na resolução dos problemas -mestres que podem apresentar matrizes demasiadamente grandes .

#### Descrição do Trabalho:

Dessa forma , nosso trabalho divide-se em duas etapas distintas, que procuram, como objetivo único, apresentar estudos de técnicas com características duais para a resolução de problemas

lineares dinâmicos .

Na primeira parte incluímos os capítulos II e III que tratam da técnica do Lagrangeano Aumentado na resolução de (P.L.D.), implementação e testes computacionais .

No capítulo II analisamos o método do ponto de vista dual e sua aplicabilidade na resolução de problemas lineares dinâmicos . A separabilidade temporal do problema obtida através da utilização da técnica de Watanabe ocasiona o aparecimento de um nível intermediário no algoritmo , fator complicante para a obtenção da sua eficiência . Para este caso , fazemos uma análise da função a ser otimizada neste nível demonstrando a sua diferenciabilidade e , conseqüentemente , garantindo uma implementação mais robusta para o algoritmo. Ainda neste capítulo fazemos interpretações geométricas do método na resolução de (P.L.D.) seguindo os trabalhos de Stephanopoulos - Westerberg .

No capítulo III descrevemos as técnicas de otimização utilizadas na implementação do método , juntamente com fluxogramas explicativos . Problemas-teste são descritos e tabelas com resultados computacionais são apresentadas .

Na segunda parte do trabalho incluímos os capítulos IV e V que tratam da técnica de decomposição do problema via dualidade, a qual obtém a solução ótima primal através de "subsistemas" dado os valores ótimos da solução dual .

Esta segunda técnica aqui abordada, quando aplicada a problemas (P.L.D.) , consiste em duas fases bastante distintas : a fase de obtenção da solução dual , a qual denominamos "1<sup>a</sup> FASE" , e a fase de obtenção da solução ótima primal via solução ótima dual, que denominamos "2<sup>a</sup> FASE" . Na 1<sup>a</sup> FASE destacamos o método da Aproximação Tangencial de Geoffrion , adaptando-o para problemas lineares dinâmicos ; descrevemos a maneira como geramos os hiperplanos à função dual e a "relaxação" dos mesmos como tratamento para sanarmos o problema de superdimensionamento da matriz de restrições gerada pelo método . A 2<sup>a</sup> FASE consiste na

resolução de um sistema linear que apresenta características importantes e que favorecem a sua decomposição em " subsistemas " com conseqüente vantagem computacional . Essa decomposição em "subsistemas " é feita com base em análise sobre o comportamento da trajetória ótima apresentado por uma amostra de problemas do tipo (P.L.D.) . Os estudos da 1ª e 2ª FASES são apresentados no capítulo IV .

No capítulo V descrevemos a técnica do Dual-Simplex para problemas com variáveis canalizadas utilizada na resolução do problema-mestre advindo da geração de hiperplanos pela técnica da Aproximação Tangencial . A implementação desta técnica é importante para o método proposto , uma vez que grande gasto operacional verifica-se nesta fase do algoritmo . A relaxação dos hiperplanos gerados e que não estão "ativos" é utilizada para obtenção de maior eficiência computacional.

O emprego da relaxação das restrições não-ativas do problema-mestre do algoritmo da Aproximação Tangencial requer extenso trabalho de atualização dos índices das linhas e colunas da matriz de restrições resultante e , conseqüentemente , dos índices das variáveis básicas e não-básicas naquela fase do trabalho . Nesta mesma fase , a matriz básica deve ser atualizada e sua matriz inversa recalculada , garantindo , dessa forma , menor risco de erros numéricos por arredondamentos no problema .

## CAPITULO II

### O MÉTODO DO LAGRANGEANO AUMENTADO

#### II.1 - INTRODUÇÃO

A utilização do método do Lagrangeano Aumentado na resolução de problemas de programação linear dinâmico surgiu como uma consequência da teoria de dualidade existente que trata da viabilidade do emprego do Lagrangeano ordinário a problemas de otimização. Sabemos, dessa teoria, ser a técnica do Lagrangeano bem sucedida quando utilizada na resolução de problemas convexos em geral; todavia, sua aplicação a problemas lineares não nos fornece, de imediato, a solução ótima primal (sendo necessária a sua determinação), pois à solução dual obtida, quando única, está associado um conjunto de soluções primais que minimizam a função Lagrangeano.

Ao adicionarmos o termo de penalização à função Lagrangeano, como veremos a seguir, conseguimos contornar o problema da determinação da solução ótima primal e isto confere ao método um fator bastante atrativo. Na seção II.3. fazemos uma interpretação geométrica desse fato, onde usaremos o conceito de função de perturbação e hiperbolóides-suporte semelhantes às interpretações encontradas nos trabalhos de Stephanopoulos - Westerbeg e Watanabe.

Contudo, a adição do termo quadrático à função Lagrangeano destrói a sua separabilidade temporal, a qual é restituída através da introdução de uma nova variável ao problema. Consequentemente, um "nível intermediário", que também deve ser otimizado, é introduzido ao algoritmo.

Uma interpretação geométrica do mecanismo de convergência do

método do Lagrangeano Aumentado é feito na seção 11.4 e na seção 11.5 fazemos uma análise da sua decomposição temporal e das consequências advindas dessa decomposição para o algoritmo - o aparecimento do "nível intermediário".

## 11.2- APRESENTAÇÃO DO (P.L.D.)

Seja o problema de programação linear dinâmico apresentado na forma

$$(PLD): \text{ Minimizar } \sum_{t=0}^{T-1} c(t+1)x(t+1) + d(t)u(t)$$

$$\text{s.a } x(t+1) = A(t)x(t) + B(t)u(t) \quad (1)$$

$$f(t) = C(t)x(t) + D(t)u(t) \quad (2)$$

$$x(0) = x^0 \quad (3.1)$$

$$x(t+1), u(t) \in S, \quad t=0, \dots, T-1 \quad (3.2)$$

onde definimos o conjunto

$$S = \{ (x(t+1), u(t)) / \underline{x}(t+1) \leq x(t+1) \leq \bar{x}(t+1), \underline{u}(t) \leq u(t) \leq \bar{u}(t), \\ t=0, \dots, T-1 \}$$

Os vetores  $c(t+1)$  e  $d(t)$  são vetores coeficientes da função objetivo do problema e  $f(t)$  é um vetor dado pertencente ao espaço  $m$ -dimensional  $R^m$ ;  $A(t)$ ,  $B(t)$ ,  $C(t)$  e  $D(t)$  são matrizes de ordens  $(n \times n)$ ,  $(n \times r)$ ,  $(m \times n)$  e  $(m \times r)$ , respectivamente.

As variáveis do problema são  $x(t)$  e  $u(t)$ , onde a primeira é a variável de estado e a segunda é a variável de controle, ambas no intervalo de tempo  $t$ , pertencentes aos espaços  $R^n$  e  $R^r$ , respectivamente. Essas variáveis possuem uma relação de

interdependência com a variável de estado no intervalo posterior,  $(t+1)$ , que é caracterizada pela equação (1).

O problema se apresenta como um problema dinâmico discreto e com horizonte "T" finito, cujas condições iniciais são dadas pela equação (3.1).

### 11.3 - O PROBLEMA DUAL DO (P.L.D.)

#### 11.3.1 - A Função Dual

Definimos a função Lagrangeano de (PLD) como

$$L(x,u,p,\lambda) = \sum_{t=0}^{T-1} [ c(t)x(t+1) + d(t)u(t) + p(t+1) [ x(t+1) - A(t)x(t) - B(t)u(t) ] + \lambda(t) [ f(t) - C(t)x(t) - D(t)u(t) ] ] \quad (4)$$

onde  $p(t+1)$  e  $\lambda(t)$ ,  $t=0, \dots, T-1$ , são os multiplicadores de dimensão  $(1 \times n)$  e  $(1 \times m)$ , respectivamente.

Tendo suas variáveis  $x(t)$  e  $u(t)$  separáveis no tempo, podemos escrever (4) na forma

$$L(x,u,p,\lambda) = \sum_{t=1}^{T-1} [ p(t) + c(t) - p(t+1)A(t) - \lambda(t)C(t) ] x(t) + \sum_{t=0}^{T-1} [ d(t) - p(t+1)B(t) - \lambda(t)D(t) ] u(t) + [ p(t) + c(t) ] x(t) - [ p(1)A(0) + \lambda(0)C(0) ] x(0) + \sum_{t=0}^{T-1} \lambda(t) f(t) \quad (4.1)$$

Logo, obtemos as seguintes formas para cada intervalo de tempo :

$$t = 0 \quad (4.1.1)$$

$$L(x(0), u(0), p(1), \lambda(0)) = [ -p(1)A(0) - \lambda(0)C(0) ]x(0) + \\ + [ d(0) - p(1)B(0) - \lambda(0)D(0) ]u(0) + \lambda(0)f(0)$$

$$t = 1, \dots, T-1 \quad (4.1.2)$$

$$L(x(t), u(t), p(t+1), \lambda(t)) = [ p(t) + c(t) - p(t+1)A(t) - \\ - \lambda(t)C(t) ]x(t) + [ d(t) - p(t+1)B(t) - \lambda(t)D(t) ]u(t) + \lambda(t)f(t)$$

$$t = T$$

$$L(x(T), p(T)) = [ p(T) + c(T) ]x(T) \quad (4.1.3)$$

Através do Lagrangeano, definimos a função dual do (P.L.D.) como

$$(FD) \quad \phi(p, \lambda) = \min_{x, u \in S} L(x, u, p, \lambda)$$

que é côncava linear-por-partes [45], cujo domínio é o espaço vetorial  $R^{(n+m)T}$

O problema dual associado ao problema (P.L.D.) é

$$(DLD) \quad \max_{p, \lambda} \phi(p, \lambda)$$

Podemos interpretar o problema (DLD) da seguinte forma: para cada valor atribuído aos vetores duais  $p$  e  $\lambda$ , o valor da

função para esses valores é obtido resolvendo o problema

$$(P1) \quad \min_{x,u \in S} L(x,u,p,\lambda)$$

que tem como conjunto-solução

$$X(p,\lambda) = \{ (x,u) \in S / (x,u) \text{ minimiza a função (4)} \}$$

Sabemos, pela teoria da programação linear, que o valor da função dual é um limitante inferior do valor da função primal e que, na solução ótima, seus valores coincidem. Dessa forma, ao maximizarmos (DLD), o seguinte critério de optimalidade deve estar satisfeito:

Seja  $(p^*, \lambda^*)$  um par de valores duais factíveis: se a este par estiver associado o par  $(x^*, u^*)$  de valores primais factíveis tal que o valor da função objetivo do (P.L.D.) coincida com o valor de (FD), então  $(x^*, u^*)$  é solução ótima para (P.L.D.) e  $(p^*, \lambda^*)$  é a solução ótima de (DLD).

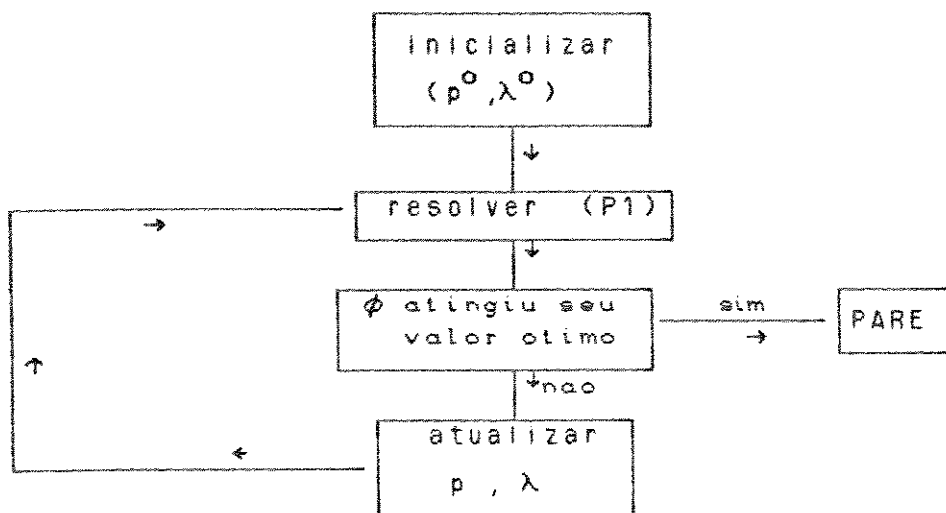
Portanto, para que consigamos alcançar a solução de (P.L.D.) através da resolução de (P1) é necessário, em primeiro lugar, pensarmos num processo de atualização dos valores dos multiplicadores que garanta a sua convergência, ou seja, que nos forneça com segurança os valores ótimos  $p^*$  e  $\lambda^*$ . No presente caso, temos uma função dual linear - por partes, logo, não diferenciável, o que implica em utilizarmos metodologias adequadas como, por exemplo, métodos que utilizam subgradientes para calcular direções factíveis, ou o método de Aproximação Tangencial, cuja minuciosa descrição é feita no terceiro capítulo.

A cada atualização dos vetores multiplicadores temos que



resolver um problema do tipo (P1), que é um problema extremamente simples de minimizar uma função linear, separável nos estágios de tempo e cujas variáveis são "canalizadas".

O processo iterativo fica simples de ser visualizado através de um fluxograma geral, como o da figura abaixo.



( figura 2.1 )

No entanto, este simples procedimento não nos garante o fornecimento da solução ótima primal do problema (P.L.D.) uma vez que (P1) arbitra os valores das variáveis associadas a coeficientes nulos em (4.1), fornecendo assim uma solução não necessariamente factível.

Uma interpretação geométrica deste fato, bastante conhecida da teoria da dualidade, utiliza o conceito de "função de perturbação" relacionada ao problema. Essa interpretação é utilizada para problemas gerais e será usada aqui apenas para o caso específico da programação linear.

### 11.3.2 - Interpretação Geométrica

Com o propósito de facilitarmos a notação, vamos considerar o problema (P.L.D.) como um problema clássico, que denotamos por

$$(PL) \quad \begin{aligned} & \min f(x) \\ & \text{s.a. } g(x) = 0 \\ & \underline{x} \leq x \leq \bar{x} \end{aligned}$$

onde  $f(x)$  é uma função linear com domínio no espaço  $R^n$  e  $g(x)$  matriz das restrições de ordem  $(m \times n)$  e  $x$ ,  $\underline{x}$  e  $\bar{x}$  vetores de  $R^n$ .

Dessa forma, escrevemos o problema (P1) como

$$(P1)' \quad \min_{\underline{x} \leq x \leq \bar{x}} l(x, \lambda) ; \quad \lambda \text{ dado}$$

onde  $l(x, \lambda)$  é a função Lagrangeano de (PL) dada por  $f(x) + \lambda^T g(x)$  e  $\lambda$  o multiplicador de Lagrange de dimensão  $(m \times 1)$ .

Para cada valor  $\lambda^0$  atribuído ao multiplicador de Lagrange corresponde um valor da função dual

$$(FD)' \quad \phi'(\lambda^0) = \min_{\underline{x} \leq x \leq \bar{x}} l(x, \lambda^0) = l(x^0, \lambda^0)$$

onde  $x^0$  é elemento do conjunto solução  $X(\lambda^0)$  do problema (P1)'.

Segundo Everett [18] cada solução de (P1)' está associada à solução de um (PP), que é denominado "problema perturbado", da seguinte forma:

Se  $x^0$  resolve (P1)' para um dado valor  $\lambda^0$ , então  $x^0$  resolve o problema

$$(PP) \quad \begin{aligned} & \min f(x) \\ & \text{s.a. } g(x) = y \quad ; \quad y = g(x^0) \\ & \underline{x} \leq x \leq \bar{x} \end{aligned}$$

Realmente, se  $x^0$  resolve (P1)' para  $\lambda^0$ , temos que  $f(x^0) + \lambda^0 g(x^0) \leq f(x) + \lambda^0 g(x)$ ,  $\forall x$  tal que  $\underline{x} \leq x \leq \bar{x}$ ; logo,  
 $f(x^0) \leq f(x) + \lambda^0 (g(x) - g(x^0))$   
 donde temos que  $f(x^0) \leq f(x)$ ,  $\forall x$  t.q.  $\underline{x} \leq x \leq \bar{x}$  e  $g(x) = g(x^0)$

Definimos a "função de perturbação" relacionada ao problema (PP) como

$$(FP) \quad V(y) = \inf_{\underline{x} \leq x \leq \bar{x}} [ f(x) \text{ s.a. } g(x) = y ]$$

que associa a cada "vetor de perturbação"  $y$  a solução ótima do problema (PP). O domínio de definição de (FP) é dado pelo conjunto  $D = \{ y / \exists x, \underline{x} \leq x \leq \bar{x} \text{ e } g(x) = y \}$ .

Notamos que o valor da função de perturbação (FP) para  $y=0$  nos dá a solução ótima de (PL).

Podemos enunciar as seguintes propriedades para a função de perturbação ( ver [23] ):

(PI)  $V(y)$  é convexa sobre um conjunto convexo se  $f(x)$  e  $g(x)$  forem convexas. No caso da (P.L.D.), (FP) é convexa linear-por-partes.

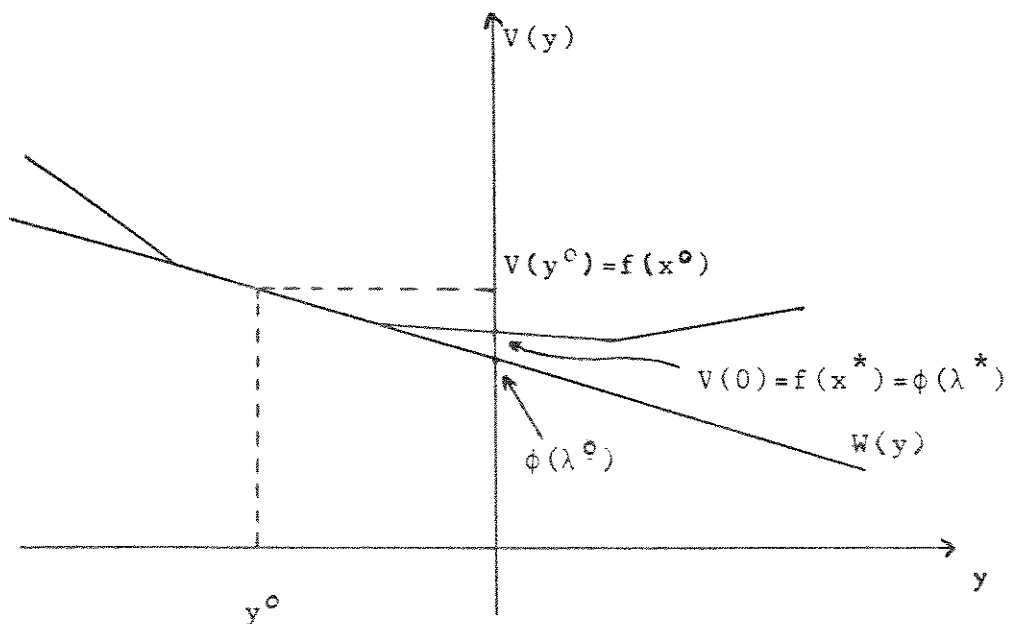
(PII)  $x^0$  minimiza  $l(x, \lambda^0)$  para  $\underline{x} \leq x \leq \bar{x}$  se e somente se  $V(y) \geq V(y^0) - \lambda^0 (y - y^0)$ ,  $\forall y \in D$  e  $y^0 = g(x^0)$

Esta última propriedade nos diz que resolver o problema (P1) é equivalente a determinarmos o hiperplano suporte à função de perturbação (FP), cujo ponto de tangência é  $y^0 = g(x^0)$ , com inclinação  $-\lambda^0$ .

A equação deste hiperplano é dada por

$$w(y) = f(x^0) - \lambda^0 [y - g(x^0)] = v(x^0, \lambda^0) - \lambda^0 y \quad (5)$$

Através da equação (5) notamos que  $w(0) = v(x^0, \lambda^0)$ , o que significa que o hiperplano  $w(y)$  corta o eixo das ordenadas exatamente no valor da função dual em  $\lambda^0$  (ver figura 2.2). Dessa forma, podemos verificar que o valor da função (FD) é sempre um limitante inferior do valor da função primal  $f(x)$ . Logo, o problema dual pode ser interpretado como o problema de encontrarmos o hiperplano com inclinação ótima,  $-\lambda^*$ , que maximiza a sua intersecção com o eixo das ordenadas.



( figura 2.2 )

Analisando a figura acima, percebemos dois casos distintos:

1º caso: cada trecho linear de "V" corresponde a um único hiperplano com inclinação  $-\lambda^\circ$  que, por sua vez, corresponde a infinitas soluções  $v(y^\circ) = f(x^\circ)$ .

2º caso: pelos pontos de quebra,  $y^\circ$ , da função "V" passam infinitos hiperplanos,  $w(y^\circ)$ , que correspondem a uma única solução primal (mínimo de  $L(x, \lambda^\circ)$  único).

O primeiro caso descreve exatamente o problema da indeterminação da solução ótima primal ao tentarmos resolver (PL) através do Lagrangeano ordinário.

A aplicação do Lagrangeano Aumentado na resolução de problemas lineares está intrinsecamente relacionada ao problema de encontrarmos a solução ótima primal quando a inclinação ótima,  $-\lambda^*$ , do hiperplano suporte (5) em  $y = 0$  se enquadra no "1º caso" acima.

#### 11.4 - O METODO LAGRANGEANO AUMENTADO - RESOLUÇÃO DO (P.L.D.)

A função Lagrangeano Aumentado relacionada ao problema (P.L.D.) é definida por

$$\begin{aligned}
 \text{(LA)} \quad m(x, u, p, \lambda, q) = & L(x, u, p, \lambda) + \\
 & + (1/2)q \sum_{t=0}^{T-1} \{ \| x(t+1) - A(t)x(t) - B(t)u(t) \|^2 + \\
 & + \| f(t) - C(t)x(t) - D(t)u(t) \|^2 \}
 \end{aligned}$$

onde  $q > 0$  é uma constante de penalização.

Dessa forma, a função (LA) pode ser encarada como a função Lagrangeano do (P.L.D.) acrescida de termos não-lineares referentes às restrições (1) e (2).

Um resultado importante associado à resolução do problema dual obtido através da função (LA) e demonstrado por Propoi e Yadykin [45] é o seguinte:

TEOREMA 1 - A função dual definida por

$$\phi_A(p, \lambda) = \min_{x, u \in S} m(x, u, p, \lambda, q) \quad (P2)$$

é côncava, continuamente diferenciável e finita para todos os valores de  $p$  e  $\lambda$ . Além disso,

$$\frac{\partial \phi_A(p, \lambda)}{\partial p(t+1)} = x(t+1) - A(t)x(t) - B(t)u(t) \quad (6)$$

$$\frac{\partial \phi_A(p, \lambda)}{\partial \lambda(t)} = f(t) - C(t)x(t) - D(t)u(t)$$

onde  $x(t+1)$  e  $u(t)$ ,  $t=0, \dots, T-1$ , são elementos de  $X(p, \lambda)$ .

A diferenciabilidade de  $\phi_A$  segue do fato da expressão em (6) ser constante para qualquer escolha de  $x(t+1)$  e  $u(t)$  em  $X(p, \lambda)$  [45].

Dessa forma, o problema dual a ser resolvido é

$$\max_{p, \lambda} \phi_A(p, \lambda) \quad (P3)$$

Uma das grandes vantagens ao trabalharmos com a função (LA) vem do fato de podermos resolver o problema (P3) facilmente tendo em vista as características da função dual dada pelo teorema acima, apresentando gradiente (6) de fácil obtenção.

Dessa forma, métodos de otimização que utilizam informação do gradiente da função e busca unidimensional podem ser aplicados na resolução de (P3). Como uma busca unidimensional na função  $\phi_A$  requer a resolução de (P2) diversas vezes, seguimos os trabalhos de Miele e outros [42], os quais propõem uma maneira simples e eficiente de atualização dos vetores duais e que pode ser aplicada na resolução de (P3).

Para o problema (PL), seção 11.3.2, a função Lagrangeano Aumentado correspondente é dada por

$$(LA)' \quad I_A(x, \lambda, q) = I(x, \lambda) + (1/2) q \|g(x)\|^2$$

e a função dual é

$$(P1)' \quad \phi_A'(\lambda) = \min_{\underline{x} \leq x \leq \bar{x}} I_A(x, \lambda, q)$$

Podemos escrever (LA)' como  $I_A(x, \lambda, q) = I(x, \lambda) + (1/2)g(x)^T g(x)$ . Para cada resolução de (P1), devemos ter as condições de primeira ordem garantidas. Ou seja,

$$\begin{aligned} & \nabla f(x) + \nabla g(x)^T \lambda + q \nabla g(x)^T g(x) = 0 \\ \text{ou} & \nabla f(x) + \nabla g(x)^T [\lambda + q g(x)] = 0 \end{aligned} \quad (9)$$

A equação (9) sugere que ao atualizarmos o vetor multiplicador de Lagrange na forma

$$\lambda_{k+1} = \lambda_k + q g(x) \quad (10)$$

estamos "perseguido", a cada atualização, as condições necessárias de otimalidade para o problema (P1)'.

Para  $\phi_A$  diferenciável, sabemos que  $g(x)$  é o seu gradiente para  $x \in X(\lambda)$  [4,39]. A atualização segundo (10) pode ser interpretada como um "passo" do método do gradiente ordinário onde "q" passa a ser o tamanho do passo.

A atualização conforme (10) nos garante que

$$I_A(x, \lambda_{k+1}, q) - I_A(x, \lambda_k, q) = q g(x)^T g(x) > 0$$

o que nos leva a concluir que a função dual cresce monotonicamente independente de uma atualização para o fator de penalidade. Logo, basta fixarmos um valor para "q" e a convergência de (P3) está garantida.

Propoi e Yadykin [45] mostram que



$$\phi_A(p_{k+1}, \lambda_{k+1}) - \phi_A(p_k, \lambda_k) \geq 1/2 \alpha \sum_{t=0}^{T-1} \| x_k(t+1) - A(t)x_k(t) - B(t)u_k(t) \|^2 + 1/2 \alpha \sum_{t=0}^{T-1} \| f(t) - C(t)x_k(t) - D(t)u_k(t) \|^2$$

ou seja, a lei de crescimento da função dual  $\phi_A$  é quadrática com o erro nas restrições.

Toda a teoria discutida relacionada à função (LA) está baseada no seguinte teorema, tendo em vista que o mesmo nos garante que a solução ótima encontrada para (P3) é a solução ótima de (DLD) [45].

TEOREMA 2 - Seja  $Q_m^*$  o conjunto de todas as soluções ótimas de (P3); sejam  $Q^*$  o conjunto solução de (DLD) e  $Z^*$  o conjunto solução de (P.L.D.). Então

$$Q_m^* = Q^* \quad \text{e} \quad X(p^*, \lambda^*) \supset Z^* = X_M(p, \lambda), \quad \forall (p, \lambda) \in Q_M^*$$

onde definimos  $X_M(p, \lambda) = \{ (x, u) \in S / (x, u) \text{ é solução de (P2)} \}$

#### 11.4.1 - Interpretação Geométrica

Para facilitarmos a notação, interpretamos novamente o problema (PLD) como um (PL) conforme seção 11.3.

O problema linear, agora, sofreu em sua função objetivo o acréscimo do termo não-linear. Logo, escrevemos o problema modificado como

$$\begin{array}{ll}
 \text{(PML)} & \min f(x) + (1/2) \rho g(x)^T g(x) \\
 & \text{s.a.} \quad g(x) = 0 \\
 & \underline{x} \leq x \leq \bar{x}
 \end{array}$$

O problema perturbado relacionado a (PML) é

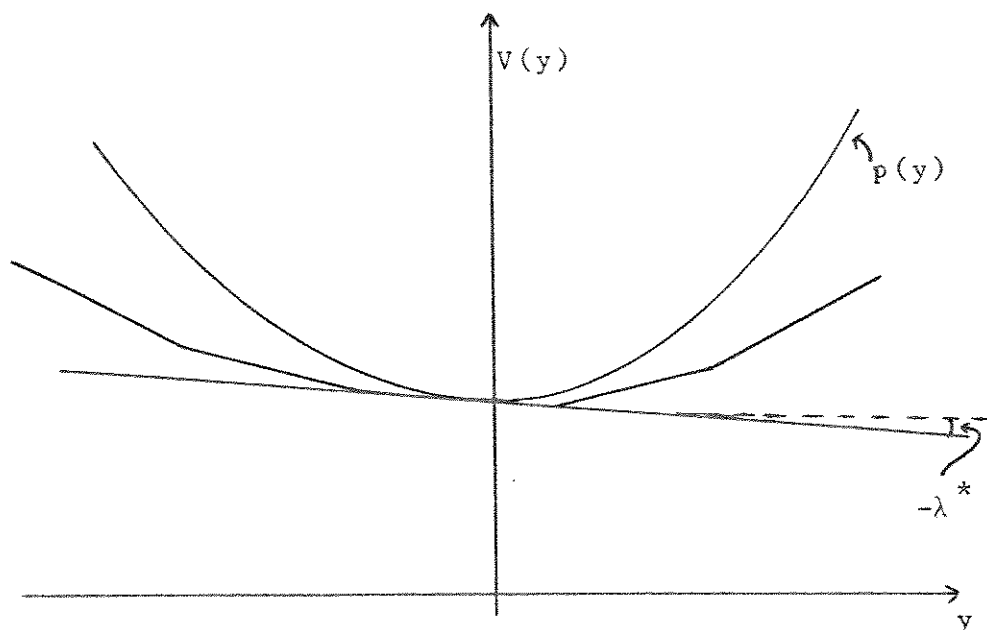
$$\begin{array}{ll}
 \text{(PPM)} & \min f(x) + (1/2) \rho g(x)^T g(x) \\
 & \text{s.a.} \quad g(x) = y \quad ; \quad y \in R^m \\
 & \underline{x} \leq x \leq \bar{x}
 \end{array}$$

com função de perturbação definida por

$$\text{(FPM)} \quad p(y) = \inf_{\underline{x} \leq x \leq \bar{x}} [ f(x) + (1/2) \rho g(x)^T g(x) \text{ s.a. } g(x)=y ]$$

A função (FPM) é não-linear e satisfaz as propriedades (Pi) e (Pii) da função de perturbação (FP) [23].

O teorema 2 da seção anterior nos garante que o hiperplano suporte à função (FP) no ponto  $y = 0$  também é suporte, neste mesmo ponto, à função (FPM) e este fato somado ao caráter não linear desta última possibilita a determinação da solução ótima primal através da obtenção da inclinação ótima,  $-\lambda^*$ , do hiperplano.



( figura 2.3 )

Uma outra interpretação geométrica para o efeito do termo quadrático da função (LA) sobre a função de perturbação (FP) pode ser vista através do conceito de hiperparabolóides-suporte [62].

Seja a função Lagrangeano Aumentado (LA)

$$l_A(x, \lambda, q) = f(x) + \lambda g(x) + (1/2) q g(x)^T g(x).$$

Interpretada no espaço da função de perturbação (FP), podemos escrever  $y = g(x)$  e  $z = f(x)$ . Assim, cada ponto  $\lambda^0$  o lugar geométrico onde  $l_A(x, \lambda^0, q)$  é uma constante (cons).

$$\text{cons} = z + \lambda^0 y + (1/2) q y^2 \quad (11)$$

que é a equação de um parabolóide com concavidade voltada para baixo e cujo máximo é atingido no ponto  $y = -\lambda^0/q$ .

Dessa forma, o procedimento do "Lagrangeano Aumentado"

pode ser interpretado da seguinte maneira: para um certo  $\lambda^0$  fixo, o problema de resolver

$$(P1)' \quad \min_{\underline{x} \leq x \leq \bar{x}} l_A(x, \lambda^0, q) = \phi_A'(\lambda^0)$$

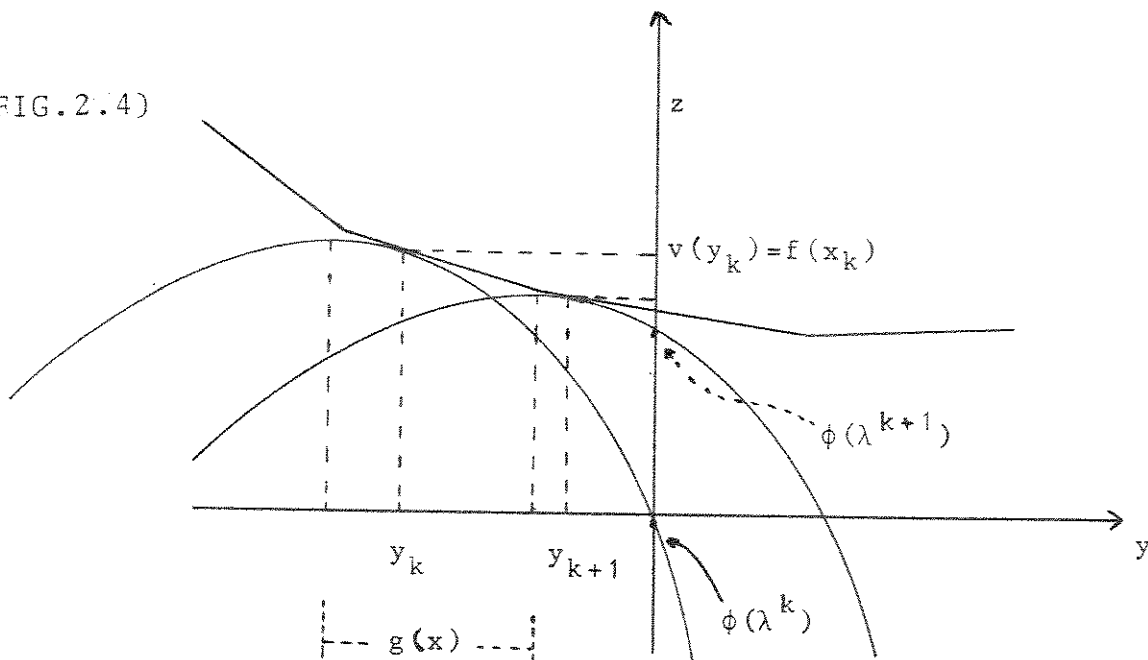
equivale a encontrarmos o parabolóide de equação (11) que tangencia a função de perturbação (FP).

Podemos notar que o ponto de intersecção do hiperparabolóide com o eixo  $y = 0$  também nos fornece o valor da função dual no ponto  $\lambda^0$  e o processo de atualização do multiplicador, conforme (10) nos garante a obtenção de valores crescentes para o problema dual, como mostra a figura 2.4.

Ao fazermos  $\lambda_{k+1} = \lambda_k + q g(x)$ , obtemos

$$\frac{-\lambda_{k+1}}{q} = \frac{-\lambda_k}{q} - g(x)$$

(FIG. 2.4)



Portanto, a adição do termo quadrático na função Lagrangeano de um (PL) soluciona o problema de encontrarmos a solução ótima primal ao tentarmos resolver um problema linear através de uma sequência de "problemas Lagrangeanos"

Dessa forma, podemos enunciar o procedimento iterativo que soluciona problemas lineares dinâmicos via aplicação do método do Lagrangeano Aumentado.

#### 11.4.2 - O Algoritmo do Lagrangeano Aumentado

PASSO1 - Sejam  $\lambda^0(t)$ ,  $p^0(t+1)$ ,  $t=0, \dots, T-1$ , valores iniciais para os multiplicadores e  $q^*$  um valor fixo para a constante de penalidade.

PASSO2 - Minimize  $m(x, u, p_k, \lambda_k, q^*)$  obtendo a solução  $u_k(t)$ ,  $x_k(t+1)$ ,  $t=0, \dots, T-1$ .

PASSO3 - Calcule o gradiente de  $\phi_A(p_k, \lambda_k)$  através de (5).

Se 
$$\frac{\partial \phi_A}{\partial p(t+1)} = \frac{\partial \phi_A}{\partial \lambda(t)} = 0$$
, pare.

PASSO4 - Atualize os valores dos multiplicadores de forma a maximizar a função dual  $\phi_A$  (conforme (6)):

$$p_{k+1}(t+1) = p_k(t+1) + q \frac{\partial \phi_A}{\partial p_k(t+1)}$$

$$\lambda_{k+1}(t) = \lambda_k(t) + q \frac{\partial \phi_A}{\partial \lambda_k(t)}. \quad \text{Volte para PASSO2.}$$

O algoritmo apresentado dessa forma possui uma estrutura de "dois níveis" semelhante à estrutura do fluxograma do algoritmo do

Lagrangeano apresentada na seção 11.3: porém, com um problema mais complicado a ser resolvido que é o de minimizarmos a função (LA).

Ao simplificarmos a resolução de (P3), complicamos sensivelmente a resolução de (P2), pois ao adicionarmos o termo quadrático ao Lagrangeano passamos de um problema de otimizar uma função linear, limitada e separável no tempo, para um problema que requer otimização de uma função não-linear com muitas variáveis, dado que a separabilidade temporal de (LA) é quebrada pelo produto de variáveis em tempos distintos que aparecem no termo quadrático adicionado.

Com o intuito de restaurar a separabilidade temporal perdida, Armentano [1] sugere a implementação, ao algoritmo apresentado acima, da forma proposta por Watanabe [62] e testada com êxito para problemas não lineares de programação dinâmica.

#### 11.5 - DECOMPOSIÇÃO DO LAGRANGEANO AUMENTADO

Ao tentarmos a separabilidade temporal da função (LA) temos, necessariamente, que escrever numa "forma separada" a expressão quadrática

$$\| x(t+1) - A(t)x(t) - B(t)u(t) \|^2 \quad (12)$$

que é a causadora dos "termos cruzados" que aparecem nessa função.

Da igualdade abaixo, válida para todo  $v(t)$ , (12.1)

$$\begin{aligned} \| A(t)x(t)+B(t)u(t)-v(t) \|^2 + \| x(t+1)-v(t) \|^2 = \\ (1/2) \| x(t+1)-A(t)x(t)-B(t)u(t) \|^2 + \\ + 2 \| v(t)-(1/2)[ A(t)x(t)+B(t)u(t)+x(t+1) ] \|^2 \end{aligned}$$

tomamos o mínimo em  $v(t)$  e obtemos (12.2)

$$\begin{aligned} & (1/2) \| x(t+1) - A(t)x(t) - B(t)u(t) \|^2 = \\ & = \min_{v(t)} \{ \| A(t)x(t) + B(t)u(t) - v(t) \|^2 + \| x(t+1) - v(t) \|^2 \} \end{aligned}$$

Dessa forma, a expressão (12.2) fornece uma forma separada para (12), porém, incluindo um novo problema de minimização ao algoritmo.

Substituindo (12.2) na função (LA), obtemos

$$\begin{aligned} m(x, u, p, \lambda, q) = & \min_{v(t)} \sum_{t=0}^{T-1} \{ L(x(t), u(t), p(t+1), \lambda(t)) + \\ & + [ p(T) + c(T) ] x(T) + q \| A(t)x(t) + B(t)u(t) - v(t) \|^2 \\ & + q \| x(t+1) - v(t) \|^2 + (1/2) q \| f(t) - C(t)x(t) - D(t)u(t) \|^2 \} \end{aligned}$$

Com isso, podemos escrever (LA) na forma

$$m(x, u, p, \lambda, q) = \min_{v(t)} \sum_{t=0}^T m(t), \quad \text{onde}$$

$$\text{para } t = 0, \tag{13.1}$$

$$\begin{aligned} m(0) = & L(x(0), u(0), p(1), \lambda(0)) + q \| A(0)x(0) + B(0)u(0) - v(0) \|^2 + \\ & + (1/2) q \| f(0) - C(0)x(0) - D(0)u(0) \|^2 \end{aligned}$$

$$\text{para } t = 1, \dots, T-1 \tag{13.2}$$

$$\begin{aligned} m(t) = & L(x(t), u(t), p(t+1), \lambda(t)) + q \| A(t)x(t) + B(t)u(t) - v(t) \|^2 + \\ & + q \| x(t) - v(t-1) \|^2 + (1/2) q \| f(t) - C(t)x(t) - D(t)u(t) \|^2 \end{aligned}$$

$$\text{para } t = T, \tag{13.3}$$

$$m(T) = [ p(T) + c(T) ] x(T) + q \| x(T) - v(T-1) \|^2$$

As funções  $m(t)$ ,  $t=0, \dots, T$ , definidas acima são quadráticas convexas, com  $m(t)$ ,  $t=1, \dots, T$ , estritamente convexa, conforme Propriedade 2 mostrada abaixo.

Conseqüentemente, a função dual fica na forma

$$\begin{aligned}
 \phi_A(p, \lambda) &= \min_{x, u \in S} \min_{v(t)} \sum_{t=0}^T m(t) = \\
 &= \min_{v(t)} \left[ \min_{x, u \in S} \sum_{t=0}^T m(t) \right] \\
 &= \min_{v(t)} V(p, \lambda, v, q) \quad (P6)
 \end{aligned}$$

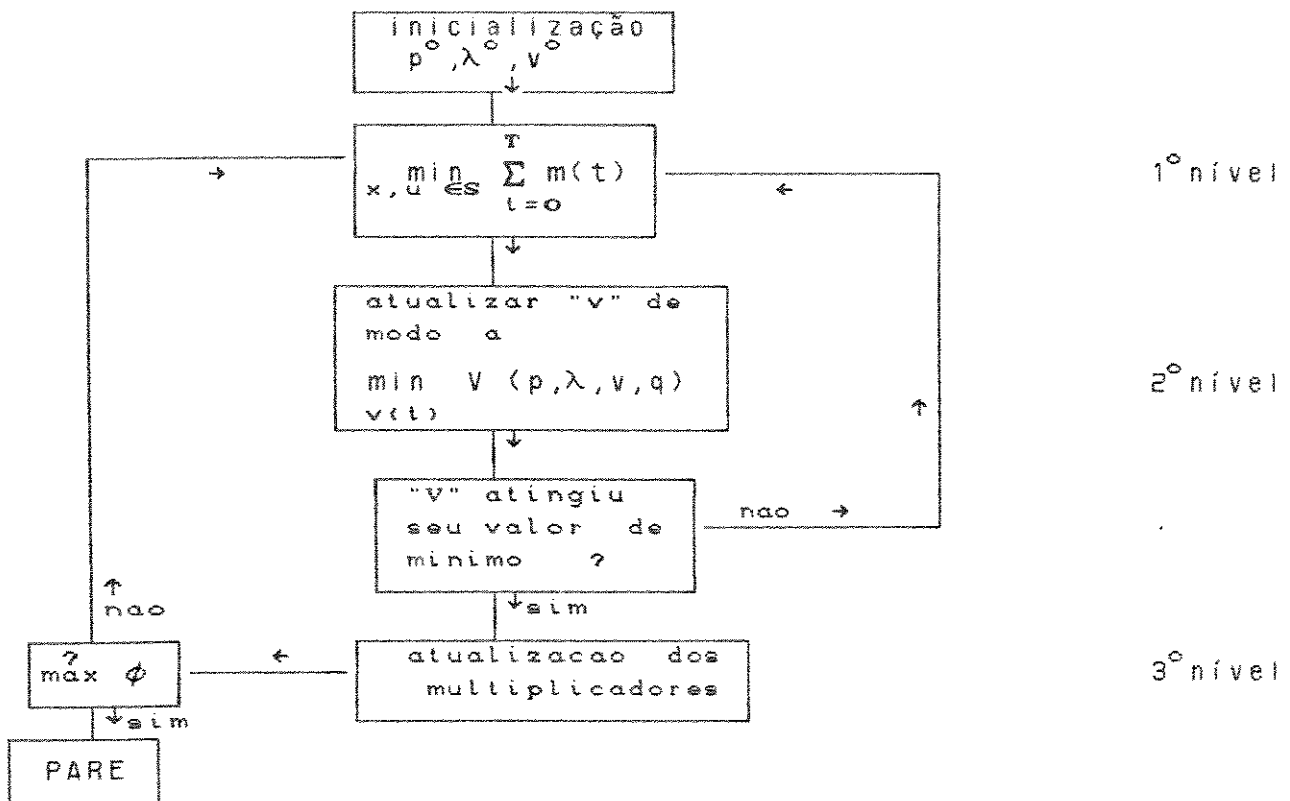
onde

$$V(p, \lambda, v, q) = \min_{x, u \in S} \sum_{t=0}^T m(t) \quad (FV)$$

Dessa forma, a estrutura do algoritmo do Lagrangeano Aumentado, anteriormente de dois níveis, agora apresenta-se em três níveis, mais complexa, conforme figura abaixo.

No "1º nível" o problema torna-se mais simples de ser resolvido; a cada estágio de tempo temos que resolver um problema de minimizar uma função quadrática convexa, como mostramos através da Propriedade 2.





( figura 2.5 )

PROPRIDADE 1 - As funções definidas em (13.1) a (13.3) são funções quadráticas convexas com  $m(t)$ ,  $t=1, \dots, T$ , estritamente convexa.

Prova:

1 - Sabemos ser a soma de funções quadráticas convexas uma função quadrática convexa, a soma de funções quadráticas por funções lineares uma função quadrática e a soma de funções convexas por uma estritamente convexa uma função estritamente convexa.

2 - Podemos escrever, para  $t=0, \dots, T-1$ ,

$$\| A(t)x(t) + B(t)u(t) - v(t) \|^2 = \| G(t)\gamma(t) - v(t) \|^2$$

onde  $G(t) = \begin{bmatrix} A(t) & B(t) \end{bmatrix}$  e  $\gamma(t) = (x(t), u(t))^T$ , que é claramente a expressão de uma função quadrática convexa.

3 - Analogamente, podemos afirmar que  $\| f(t) - C(t)x(t) - D(t)u(t) \|^2$  expressa uma função quadrática convexa em  $x(t)$  e  $u(t)$ .

4 -  $\| x(t) - v(t-1) \|^2 = x(t)^T x(t) - 2 v(t-1)^T x(t) + v(t-1)^T v(t-1)$  para  $t=1, \dots, T$ , que é uma função estritamente convexa.

Logo, para  $p(t+1)$ ,  $\lambda(t)$ ,  $v(t)$  e  $f(t)$  dados, podemos concluir que:

- a)  $m(0)$  é uma função quadrática convexa ( por 1, 2 e 3 );
- b)  $m(t)$  é uma função quadrática estritamente convexa ( por 1, 2, 3 e 4 ) para  $t = 1, \dots, T-1$  ;
- c)  $m(T)$  é uma função quadrática estritamente convexa (por 1 e 4).

Dadas as boas características e as facilidades em resolvermos os problemas dos "1º e 3º níveis", nosso problema agora se resume em estudarmos as características que se apresentam no problema a ser resolvido no "2º nível" do novo algoritmo.

Seguindo o trabalho de Watanabe [62], Armentano [1] demonstra os seguintes resultados para (P.L.D.) :

TEOREMA 3 - A função (FV) é convexa em relação a "v".

TEOREMA 4 - Se (FV) é uma função diferenciável no ponto  $v = \bar{v}$  então

$$\frac{\partial V(p, \lambda, v, q)}{\partial v(t)} = 4q [\bar{v}(t) - (1/2)(A(t)\bar{x}(t) + B(t)\bar{u}(t) + \bar{x}(t+1))]$$

onde  $\bar{u}(t)$ ,  $\bar{x}(t+1)$ ,  $t=0, \dots, T-1$ , são as soluções dos subproblemas para  $v(t) = \bar{v}(t)$ . (14)

É inegável a importância desses resultados. Porém, outros dois fatores são fundamentais para obtermos um bom desempenho para o algoritmo: a diferenciabilidade e o "bom comportamento" (\*) da função (FV). Juntos, certamente haverá condições suficientes para uma boa eficiência do algoritmo. Mas, basta um mal comportamento da função (FV) para que o mesmo apresente lentidão em sua convergência associado a um alto custo computacional em consequência da interdependência da resolução dos "1º e 2º níveis", conforme mostra a figura 2.5.

Obviamente, a condição de diferenciabilidade de (FV), juntamente com os resultados dos Teoremas 4 e 5, nos dá garantias de convergência para um algoritmo que utiliza direções de decréscimo (como, por exemplo, a direção de menos gradiente) na resolução do "2º nível": ou seja, para (FV) diferenciável, a resolução do "2º nível" torna-se mais eficaz com a possibilidade de utilizarmos algoritmos já conhecidos que minimizam funções sem restrições.

---

(\*) Dizemos que uma função é "bem comportada" quando a relação entre os autovalores da matriz hessiana está próxima à unidade.

11.5.1 - A Diferenciabilidade de (FV)

Da igualdade (12.1) podemos escrever a função (FV) na forma

$$V(p, \lambda, v, q) = \min_{x, u \in S} F(x, u, v) \quad (15)$$

onde definimos (16)

$$\begin{aligned} F(x, u, v) = & \sum_{t=0}^{T-1} [ l(x, u, p, \lambda) + [ p(T) + c(T) ] x(T) + \\ & + (1/2) q \| x(t+1) - A(t)x(t) - B(t)u(t) \|^2 + \\ & + (1/2) q \| f(t) - C(t)x(t) - D(t)u(t) \|^2 + \\ & + 2q \| v(t) - (1/2) [ A(t)x(t) + B(t)u(t) + x(t+1) ] \|^2 ] \\ & (q > 0) \end{aligned}$$

Seja S conjunto conforme definimos em 11.2; temos que:

- C1) S é um conjunto compacto;
- C2) F é uma função contínua, pois é soma de funções contínuas<sup>(\*)</sup>;

---

(\*) (a) Por 11.5, podemos tomar  $F(x, u, v) = \sum_{t=0}^T m(t)$

(b) Qualquer função definida na forma  $f(x) = \| Ax - b \|^2$  é uma função quadrática de classe  $C_1$ .

$$C3) \quad \frac{\partial F(x, u, v)}{\partial v(t)} = 4q [v(t) - (1/2)(A(t)x(t) + B(t)u(t) + x(t+1))] \\ \text{para } t = 0, \dots, T-1, \text{ é contínua para cada } t.$$

Seja o conjunto

$$Y(v) = \{ (x(t+1), u(t)) / (x(t+1), u(t)), t=0, \dots, T-1, \text{ minimiza } F(x, u, v) \text{ sobre } S \}.$$

Aplicando o Teorema de Danskin ( Apêndice 1 ), temos que a derivada direcional da função em (10) existe em qualquer direção "s" para qualquer vetor  $v(t)$ ,  $t=0, \dots, T-1$ , e é dada por

$$D V(p, \lambda, v, q) = \min_{x, u \in Y(v)} \sum_i s_i \frac{\partial F(x, u, v)}{\partial v_i(t)}$$

Ao tomarmos a direção " $s_i$ " como sendo uma das direções dos eixos coordenados, obtemos

$$D V(p, \lambda, v, q) = \min_{x, u \in Y(v)} \frac{\partial F(x, u, v)}{\partial v} \quad (17)$$

A condição necessária para que tenhamos a diferenciabilidade da função (FV) tratada no "2 nível" é

C4) A função " F " definida em (16) é uma função estritamente convexa.

Prova: Em 11.5 vimos que a função (FV) é

$$V(p, \lambda, v, q) = \min_{x, u \in S} \sum_{t=0}^T m(t).$$

Logo,  $F(x, u, v) = \sum_{t=0}^T m(t)$ ; ou seja,  $F$  é soma de funções convexas com  $m(t)$ ,  $t=1, \dots, T$ , estritamente convexa (conforme verificado em 11.5), o que nos garante a sua convexidade estrita.

Dessa forma, para cada  $v(t)$ ,  $t=0, \dots, T-1$ , o conjunto  $Y(v)$  é unitário; portanto, a condição C4) nos garante que a derivada direcional em (17) é constante para todo  $v(t)$ . Logo, concluímos que (FV) é diferenciável com

$$\frac{\partial V(p, \lambda, v, q)}{\partial v_i(t)} = \frac{\partial F(x, u, v)}{\partial v_i(t)} ; \quad \forall v(t) \quad (18)$$

Com a garantia da diferenciabilidade de (FV) no "2º nível", um algoritmo eficiente deve ser utilizado na resolução deste nível para obtenção de melhoria na convergência e custo computacional do Método do Lagrangeano Aumentado.

## CAPITULO III

### IMPLEMENTAÇÃO DO LAGRANGEANO AUMENTADO TESTES E CONCLUSÕES

#### III.1 - INTRODUÇÃO

Como vimos no capítulo anterior, o método do Lagrangeano Aumentado na resolução de problemas lineares dinâmicos apresenta-se com uma estrutura de três níveis, estando todos eles interligados. Vimos também que devemos obter eficiência na resolução do "2º nível" dada a sua forte influência na eficiência do algoritmo como um todo. A cada atualização da função (FV) (capítulo anterior) devemos "visitar" o "1º nível" e resolver "T+1" subproblemas. Resolver cada subproblema do "1º nível" significa minimizarmos uma função quadrática convexa cujas variáveis possuem limites superior e inferior (ou seja as variáveis são "canalizadas"); portanto, temos condições favoráveis para o bom desempenho do método quando da resolução deste "nível".

Da teoria básica da otimização de funções não-lineares, vários são os métodos que podemos utilizar na resolução dos "1º e 2º níveis"; dentre eles destacamos os métodos de Newton e Gradiente Conjugado. Sabemos que, para pontos de inicialização próximos da solução ótima, o método de Newton apresenta convergência mais rápida e custo computacional menor que o método do Gradiente Conjugado; porém, para funções de muitas variáveis e pontos de inicialização distantes da solução ótima, o algoritmo do Gradiente Conjugado (G-G) apresenta-se mais "robusto" com relação ao método de Newton [10], além de requisitar

menos memória pelo fato de ser necessário guardar apenas alguns vetores .

Pelos motivos citados acima e pelo fato de não dispormos da matriz Hessiana da função (FV) a ser minimizada no "2º nível", escolhemos o método C - G para a implementação dos 1º e 2º níveis do algoritmo aqui proposto . Portanto , damos especial atenção a este método descrevendo parte de sua teoria na segunda seção deste capítulo ; na seção III.3 descrevemos o algoritmo do Gradiente Conjugado utilizado na resolução de funções com variáveis "canalizadas" (1º nível ) e na seção III.4 a versão do mesmo método utilizado no tratamento do "2º nível" . Finalizamos o capítulo apresentando alguns testes realizados e suas conclusões .

### III.2 - O MÉTODO DO GRADIENTE CONJUGADO

O algoritmo das direções conjugadas foi introduzido originalmente por Hestenes e Stiefel ( 1952 ) com ênfase na aplicação para a resolução de sistemas lineares. Desde então, modificações e extensões deste método tem sido apresentadas por vários autores ; citamos dentre eles os trabalhos de Davidon , Fletcher e Powell.

O método do Gradiente Conjugado equivale ao método de Newton a cada  $n$  passos, sendo que, a cada passo, obtém uma aproximação maior da inversa da matriz Hessiana da função a ser otimizada [29]. Esse método também baseia-se em aproximar a função por uma forma quadrática através da qual o cálculo das direções conjugadas é imediato e pode ser estendido para funções não-quadráticas em geral

Seja  $f: E^n \rightarrow R$  função de classe  $C^2$  definida no espaço  $n$ -dimensional  $E^n$ .

Dessa forma , a cada ponto  $x \in E^n$ , podemos representar a



função  $f$  numa vizinhança de  $x$  na forma

$$F(x) = (1/2) x^T A x - h^T x + C \quad (1)$$

onde  $A$  é a matriz hessiana de  $f$ , de dimensão  $n \times n$ ,  $h$  é o vetor gradiente, de dimensão  $n$  e  $C$  é uma constante.

A condição necessária de primeira ordem da otimização diz que o ponto crítico de  $F$  deve ser tal que  $F'(x)=0$ . Mas,  $F'(x) = Ax-h = 0$ , ou seja, o ponto crítico da função em (1) deve ser solução do sistema

$$Ax = h \quad (2)$$

Se a matriz  $A$  é não-singular, então o ponto crítico é único e é dado por  $x^+ = A^{-1}h$ .

Sabemos que  $x_0$  é um ponto de mínimo para  $F$  se e somente se a matriz  $A$  é definida positiva e a direção de máxima descida de  $F$  num ponto  $x$  qualquer é

$$r(x) = -F'(x) = h - Ax \quad (3)$$

O método do Gradiente Conjugado pode ser visto como um método intermediário entre o método que usa direções do tipo (3) e o método de Newton [29].

Vejamos, primeiramente, as propriedades acerca das direções conjugadas.

Definição : Dada uma matriz simétrica  $A$  ( genérica ), dizemos que dois vetores  $d_1$  e  $d_2$ , em  $\mathbb{E}^n$ , são  $A$ -ortogonais, ou conjugados com respeito a  $A$ , se  $d_1^T A d_2 = 0$ .

NOTA : Duas observações imediatas resultam da definição acima :

1<sup>a</sup>) Se  $A = 0$ , então quaisquer vetores de  $\mathbb{E}^n$  são  $A$ -ortogonais;

2<sup>a</sup>) Se  $A = I$ , então  $d_1$  e  $d_2$  são ortogonais.

Dessa forma, um conjunto de vetores  $d_0, d_1, \dots, d_k$  são

A-ortogonais se  $d_i^T A d_j = 0$  para  $i \neq j$ .

A proposição abaixo é muito importante, pois nos dá garantias de que um conjunto de  $k$  vetores A-ortogonais gera o espaço de dimensão  $k$ .

PROPOSIÇÃO : Se  $A$  é positiva definida e se o conjunto de vetores não nulos  $d_0, d_1, \dots, d_k$  é A-ortogonal, então esses vetores são linearmente independentes.

PROVA : ( é imediata )

Assim, se tomamos  $n$  vetores  $d_0, d_1, \dots, d_{n-1}$  A-ortogonais (supondo  $A$  definida positiva), então qualquer vetor do espaço  $\mathbb{E}^n$  pode ser escrito como uma combinação linear desses vetores. Em particular, a solução do sistema (2) pode ser escrita como

$$x^+ = \alpha_0 d_0 + \alpha_1 d_1 + \dots + \alpha_{n-1} d_{n-1} \quad (4)$$

onde algum dos  $\alpha_i$ 's é não nulo.

Da igualdade acima podemos escrever

$$A x^+ = \alpha_0 A d_0 + \alpha_1 A d_1 + \dots + \alpha_{n-1} A d_{n-1}$$

ou ainda,  $d_i^T A x^+ = \alpha_i d_i^T A d_i$  ; ou seja,

$$\alpha_i = \frac{d_i^T A x^+}{d_i^T A d_i} = \frac{d_i^T h}{d_i^T A d_i} \quad (5)$$

Conseqüentemente, a equação (4) pode ser reescrita como

$$x^+ = \sum_{i=0}^{n-1} \frac{d_i^T h}{d_i^T A d_i} d_i \quad (6)$$

Através de (6) podemos pensar em obter a solução  $x^+$  iterativamente, onde, a cada passo, um termo  $\alpha_i d_i$  é adicionado; com isso, ao final do n-ésimo passo devemos esperar ter alcançado nosso objetivo. É o que resume o teorema seguinte.

TEOREMA 1. Seja  $\{d_i\}_{i=0}^{n-1}$  um conjunto de vetores não nulos A-ortogonais. Para qualquer  $x_0 \in \mathbb{E}^n$ , a sequência  $\{x_k\}$  gerada da forma

$$x_{k+1} = x_k + \alpha_k d_k \quad (7)$$

com

$$\alpha_k = \frac{r_k^T d_k}{d_k^T A d_k}, \text{ onde } r_k = h - A x_k, \quad k \geq 0$$

converge para a solução  $x^+$  do sistema (2) após n passos, ou seja,  $x_n = x^+$ .

Prova: (vide em Luenberger [39])

Uma propriedade muito importante do procedimento iterativo acima é que, a cada passo, o novo elemento da sequência,  $x_{k+1}$ , é exatamente o mínimo da função (1) com relação à direção  $d_k$  e, mais ainda,  $x_{k+1}$  é o mínimo de  $f$  sobre o espaço de dimensão  $k$ ,  $k < n$ .

Para provarmos este resultado, definimos primeiramente por  $B^k$  o subespaço de  $\mathbb{E}^n$  gerado pelo conjunto  $\{d_i\}_{i=0}^{k-1}$ .

Agora podemos enunciar o teorema

TEOREMA 2 Seja  $\{d_i\}_{i=0}^{k-1}$  uma seqüência de vetores não nulos A-ortogonais. Então, para qualquer  $x_0 \in \mathbb{E}^n$ , a seqüência gerada por (7) tem a propriedade de que  $x_k$  minimiza a função quadrática (1) sobre a reta

$$x_k = x_{k-1} + \alpha d_{k-1} \quad (\alpha \in \mathbb{R})$$

assim como sobre a variedade linear  $x_0 + B^k$ .

Prova : A prova deste teorema se resume em mostrarmos que o gradiente de  $f$  no ponto  $x_k$  deve ser ortogonal à variedade  $x_0 + B^k$  e é feita por indução .

Por indução , como  $B^0$  é vazio , a hipótese é verdadeira para  $k = 0$  . Supomos então que  $r_k$  é ortogonal a  $B^k$  e demonstramos que  $r_{k+1}$  é ortogonal a  $B^{k+1}$  .

Temos 
$$r_{k+1} = r_k + \alpha_k A d_k \quad (8)$$

Logo , 
$$d_k^T r_{k+1} = d_k^T r_k + \alpha_k d_k^T A d_k = 0 \quad (\text{por (7)})$$

e , 
$$d_i^T r_{k+1} = d_i^T r_k + \alpha_k d_i^T A d_k = 0 \quad \text{para } i < k .$$

Assim ,  $r_{k+1}$  é ortogonal a  $B^{k+1}$  , pois é ortogonal ao conjunto de vetores que o gera . ( c.q.d. ).

Para completarmos o processo iterativo do método , resta-nos saber de que maneira devemos gerar as direções conjugadas a cada passo.

Podemos verificar [39] que o seguinte critério de geração nos garante que , a cada passo , a nova direção obtida é A-ortogonal , ou conjugada , com as direções anteriores .

Para  $k = 0$  (passo inicial) fazemos  $d_0 = r_0$  ;

Para  $k = 1, \dots, n-1$  ,  $d_k = r_k + \beta_{k-1} d_{k-1}$  ,

onde , 
$$\beta_{k-1} = \frac{r_k^T A d_{k-1}}{d_{k-1}^T A d_{k-1}}$$

Notamos que se no passo  $k$   $r_k$  é nulo, então devemos estar no ótimo do problema .

Logo , podemos enunciar o algoritmo do Gradiente Conjugado da seguinte forma :

PASSO 1 : seja  $x_0 \in \mathbb{E}^n$  o ponto inicial e seja

$$d_0 = r_0 = h - A x_0 \quad (9.1)$$

PASSO 2 : se  $r_k = 0$  , PARE ;  $x_k$  é ponto de ótimo . (9.2)

Caso contrário, vá para o PASSO 3 .

PASSO 3 : atualize  $x_{k+1} = x_k + \alpha_k d_k$  ,

onde 
$$\alpha_k = \frac{r_k^T d_k}{d_k^T A d_k} \quad (9.3)$$

e  $d_{k+1} = r_{k+1} + \beta_k d_k$  ,

onde 
$$\beta_k = \frac{-r_{k+1}^T A d_k}{d_k^T A d_k} \quad (9.4)$$

Faça  $k = k + 1$  e vá ao PASSO 2 .

Como podemos observar , este algoritmo é de fácil implementação e converge num número finito de passos .

Porém , o método do Gradiente Conjugado está sujeito a erros de arredondamentos causados pelas aproximações feitas pelo computador e , dependendo da máquina , esses erros podem ter grande

influência na convergência do algoritmo : além do mais, devemos lembrar que para funções não-quadráticas dificilmente o algoritmo convergirá em  $n$  passos, dados os erros já causados pela aproximação local da função por uma forma quadrática .

Entretanto, para funções quadráticas podemos esperar que o algoritmo convirja em um número  $m$  de passos , com  $m \leq n$  .

Dessa forma , devemos considerar que o algoritmo acima pode não convergir em um número de passos igual a  $n$ , sendo, neste caso, necessária uma reinicialização do processo iterativo .

Portanto, ao algoritmo descrito acima devemos acrescentar um "passo" intermediário aos PASSO 2 e PASSO 3 :

PASSO 2.1 : (reinicialização do processo)

Se  $k > n$  , faça  $x_0 = x_n$  e vá para o PASSO 1 .

Podemos encontrar em Luenberger [39] fórmulas alternativas para (9.3) e (9.4) , mais convenientes computacionalmente que as originais .

Muitos foram os trabalhos realizados com o propósito de simplificar a implementação deste algoritmo quando aplicado na otimização de funções não quadráticas . Como podemos perceber , a cada passo temos que fazer uma aproximação local da função por uma forma quadrática, o que requer avaliar não somente o gradiente, mas também a matriz hessiana a cada ponto. Dessa forma, surgiram outras fórmulas para as atualizações dadas por (9.3) e (9.4) que dispensam a necessidade de tais avaliações. Dentre elas estão as já bastante conhecidas fórmulas de Sorenson-Wolfe, Flecher-Reeves, Polyak-Rebire, Shanno-Phua, todas equivalentes quando tratam funções quadráticas, mas apresentando diferentes comportamentos para outras funções.

O método de Flecher - Reeves , por exemplo , substitui a avaliação (9.3) por uma procura unidimensional e  $\beta_k$  é dado, na  $k$ -ésima iteração , por

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \quad (10)$$

### III.3 - IMPLEMENTAÇÃO DO "1º NÍVEL"

Para a resolução do problema situado neste "nível" a escolha do método do Gradiente Conjugado deveu-se também pela facilidade de sua implementação para problemas com variáveis "canalizadas" e pelas garantias de seu sucesso para funções quadráticas.

Como vimos no capítulo anterior, a função a ser minimizada no "1º nível" do método do Lagrangeano Aumentado é separável por estágio de tempo. Através das equações (13.1) e (13.3), capítulo II, vimos que cada função  $m(t)$  é no mínimo quadrática semi-definida positiva, para  $t = 0, \dots, T$ . Logo, como mostra a figura 2.5, capítulo II, a resolução do "1º nível" implica resolvermos "T+1" subproblemas da forma

$$\min_{x,u \in S_t} m(t) \quad (11)$$

$S_t \subset S$ ,  $S_t = \{ u(t), x(t+1) / ( u(t), x(t+1) ) \in X( p(t+1), \lambda(t) ) \}$   
e "m" função quadrática convexa.

O algoritmo utilizado para resolução deste nível é uma extensão da técnica desenvolvida por Polyak (1969) para minimizar funções restritas a conjuntos [29].

#### III.3.1 - Algoritmo Para Gradiente Conjugado ( C - G - C )

Seja o problema

$$(P_P) : \begin{aligned} \text{Min } F(x) &= (1/2) x^T A x - h^T x \\ \text{s.a.} \quad & l \leq x \leq u \end{aligned}$$

com  $A_{n \times n}$ ,  $h^T$ ,  $x$ ,  $u$  e  $l$  vetores de  $E^n$ .

Dessa forma enunciaremos :

PASSO 1 : Escolha um ponto factível para  $(P_P)$  e calcule

$$r_1 = h - A x_1 \quad (1.a)$$

PASSO 2 : Seja  $I$  o conjunto de índices  $i$  tais que

$$x_1^i = l_i \quad \text{e} \quad r_1^i \leq 0 \quad (2.a)$$

Seja  $J$  o conjunto dos índices  $i$  tais que

$$x_1^i = u_i \quad \text{e} \quad r_1^i \geq 0 \quad (2.b)$$

se  $r_1^i = 0$  para qualquer  $i \in \{I, J\}$ , então  $x_1$  é ponto de mínimo de  $F$  ; PARE . Caso contrário , vá ao PASSO 3 .

PASSO 3 : Faça  $p_1 = \bar{r}_1$ , onde  $\bar{r}_1 = \begin{cases} 0, & i \in I \cup J \\ r_1^i, & i \notin I \cup J \end{cases}$

PASSO 4 : ( SUBROTINA C - G )

$k = 1$  ; calcule

$$\begin{aligned} s_k &= A p_k \quad ; \quad c_k = p_k^T r_k \\ d_k &= p_k^T s_k \quad \text{e} \quad a_k = c_k / d_k \end{aligned} \quad (4.a)$$

$$x_{k+1} = x_k + a_k p_k \quad (4.b)$$

$$r_{k+1} = r_k - a_k s_k$$

Se  $x_{k+1}$  é infactível , vá para o PASSO 5 ; caso contrário ,



redefinir  $I$  e  $J$  e calcule  $\bar{r}_{k+1}$  e  $p_{k+1}$  onde

$$\bar{r}_{k+1} = \begin{cases} 0, & i \in I \cup J \\ r_i^i, & i \notin I \cup J \end{cases} \quad (4.c)$$

$$p_{k+1} = \bar{r}_{k+1} + b_k p_k, \quad \text{onde} \quad b_k = \frac{-s_k^T \bar{r}_{k+1}}{d_k} \quad (4.d)$$

Se  $\bar{r}_{k+1}^i = 0$  para  $i \notin (I, J)$ , PARE. Senão, calcule (4.d), faça  $k = k + 1$  e vá para (4.a).

PASSO 5 : Sejam  $i_1, \dots, i_p$  os índices  $i$  tais que  $x_{k+1}^i < l_i$  e  $j_1, \dots, j_q$  os índices  $j$  tais que  $x_{k+1}^j > u_j$ .  
Sejam :

$$\alpha_k = \min \{ (l_i - x_k^i) / p_k^i, i \in I_p \}$$

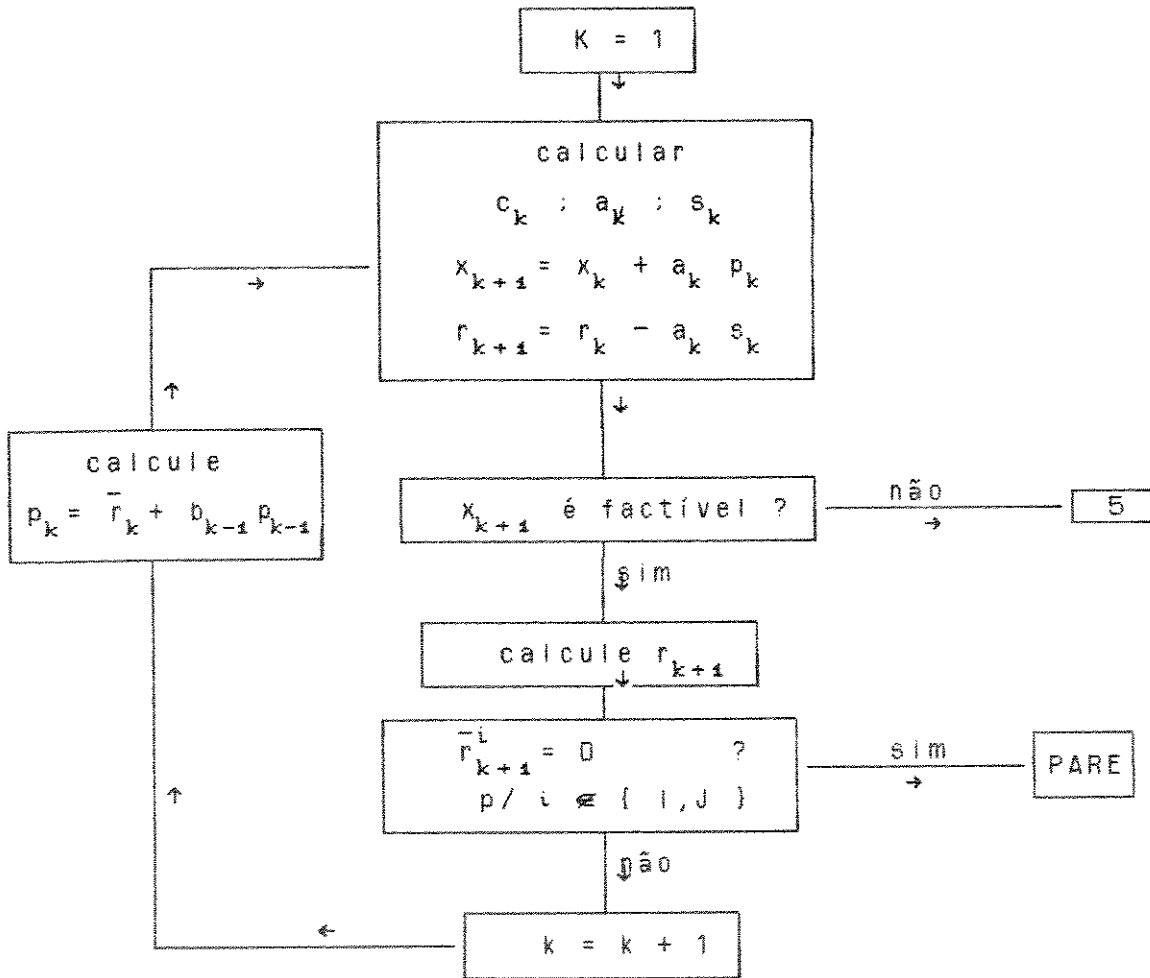
$$\text{e} \quad \beta_k = \min \{ (u_j - x_k^j) / p_k^j, j \in J_q \}$$

$$\text{Seja} \quad \bar{a}_k = \min \{ \alpha_k, \beta_k \} \quad (5.a)$$

Recoloque  $x_{k+1} = x_k + \bar{a}_k p_k$  e vá para PASSO 1.

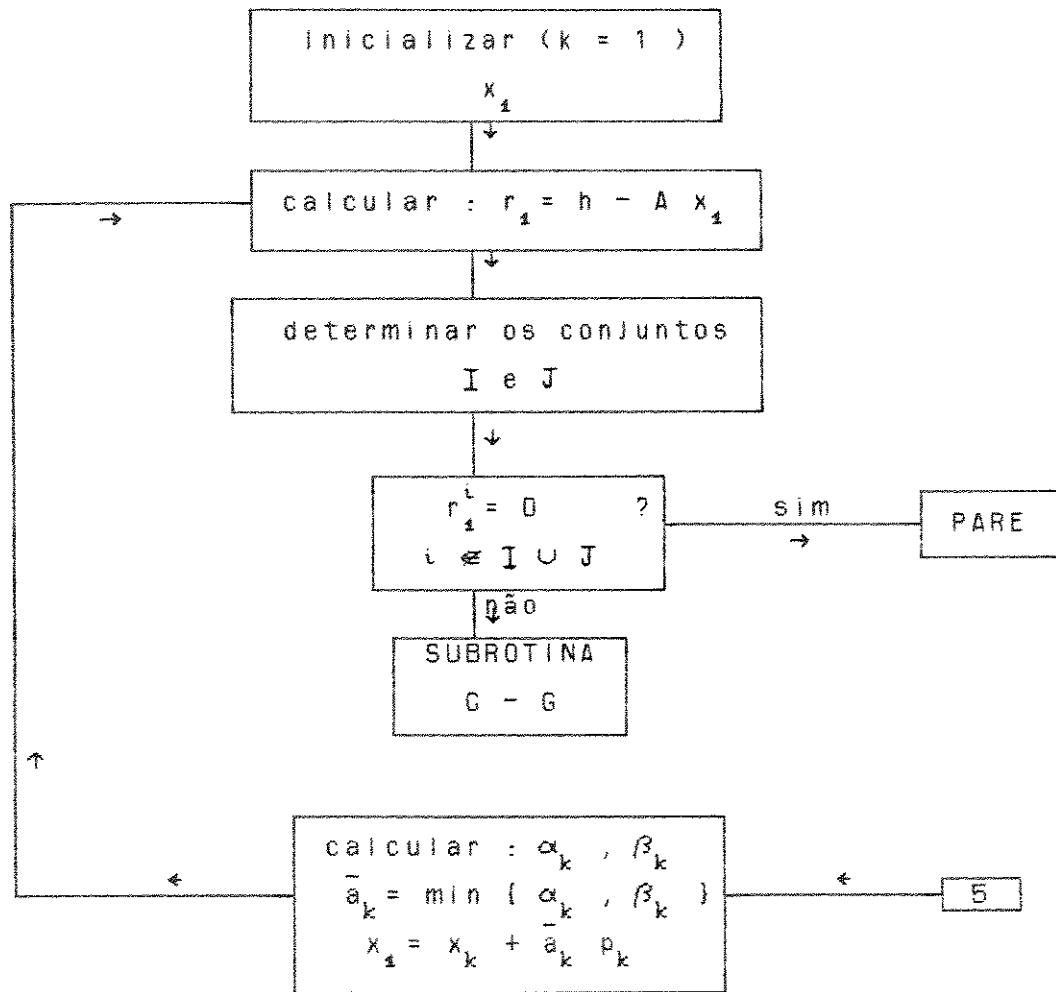
Podemos agora apresentar o fluxograma do algoritmo descrito acima.

SUBROTINA C - G



(FIG. 3.1)

FLUXOGRAMA PARA C - G - C

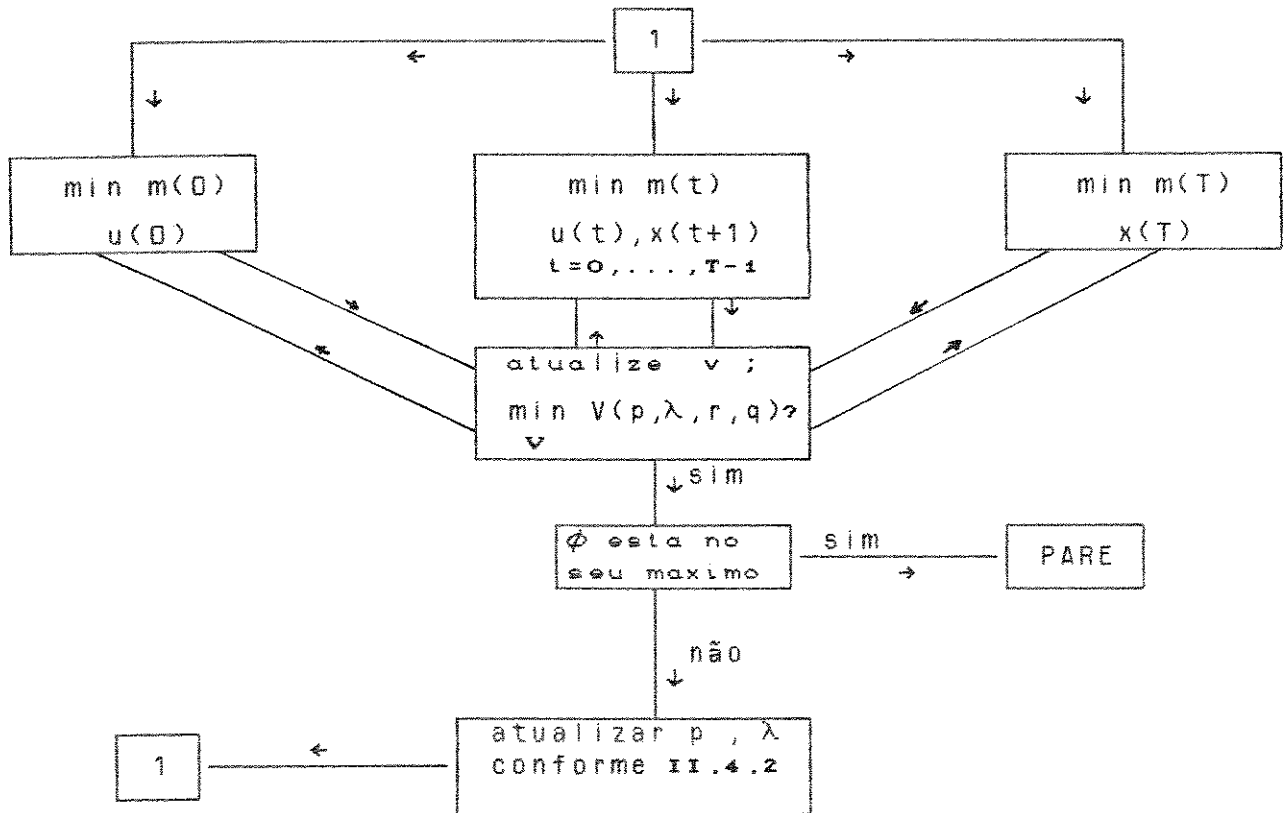


(FIG 3.2)

III.4 - IMPLEMENTAÇÃO DO " 2º NÍVEL "

Ao resolvermos o problema do " 2º nível ", notamos que cada

atualização do vetor  $v(t)$  ( $t=0, \dots, T-1$ ) da função (FV) corresponde a resolvermos também o problema do "1º nível", conforme o problema (P6) do capítulo II ; logo, estes dois níveis estão fortemente interligados e resolvermos o "2º nível" significa resolvermos simultaneamente o "1º nível". Assim, o fluxograma da figura 2.5 fica da seguinte forma



( figura 3.3 )

Dessa maneira, devemos empregar esforços no sentido de obtermos uma rápida resolução para o "2º nível", dada sua influência sobre a eficiência da abordagem como um todo.

Logo, a forma de atualizarmos os valores de  $v(t)$  ( $t=0, \dots, T-1$ )

é bastante significativa para esse algoritmo. Demonstrada a diferenciabilidade da função (FV) ( seção 11.5 ), podemos garantir eficácia na resolução deste "nível" ao utilizarmos métodos existentes na literatura que possuem convergência garantida.

#### 111.4.1.- Atualização com Busca Unidimensional - Utilização de Gradientes Conjugados

Explorando o fato demonstrado no capítulo anterior sobre a diferenciabilidade e convexidade da função (FV), podemos aplicar um método com convergência assegurada na resolução do problema do "2º nível". Assim foi que optamos pela utilização do método do Gradiente Conjugado, para funções com variáveis irrestritas, para a atualização do vetor " v " .

Nossa escolha pelo algoritmo do Gradiente Conjugado deve-se, principalmente, ao fato do mesmo apresentar convergência mesmo quando é inicializado em pontos não muito próximos da solução ótima e também por ser um algoritmo de fácil implementação .

#### 111.4.2 - Algoritmo C - G

Este algoritmo é uma versão do método para problemas com função não quadráticas e foi introduzido por Polyak e Ribire (1969) ( ver Hestenes [29] ) .

Ao algoritmo apresentado (9.1) a (9.4) são feitas modificações de maneira que não necessitamos fazer uso de (9.3) e (9.4) para os cálculos de "  $\alpha_k$  " e "  $\beta_k$  " .

Seja f a função a ser minimizada ; selecionamos  $\alpha = a_k$  que minimiza

$$\Phi_k(\alpha) = f(x_k + \alpha p_k)$$

onde,  $p_k$  é a direção conjugada na  $k$ -ésima iteração.

Logo,

$$0 = \Phi_k'(a_k) = f'(x_k + a_k p_k)^T p_k = f'(x_{k+1})^T p_k = -r_k^T r_{k+1}$$

donde vem que  $r_{k+1} = -f'(x_{k+1})$  é ortogonal a  $p_k$  e

$$p_{k+1}^T r_{k+1} = (r_{k+1} + b_k p_k)^T r_{k+1} = \|r_{k+1}\|^2.$$

Portanto, 
$$C_k = p_k^T r_k = \|r_k\|^2, \quad \forall k. \quad (13.1)$$

Seja  $s_k$  definido na forma

$$s_k = \frac{f'(x_k + a_k p_k) - f'(x_k)^T}{a_k} = \frac{r_k - r_{k+1}}{a_k} \quad (13.2)$$

então,

$$d_k = p_k^T s_k = \frac{p_k^T (r_k - r_{k+1})}{a_k} = \frac{p_k^T r_k}{a_k} = \frac{C_k}{a_k},$$

ou seja,  $d_k = \frac{C_k}{a_k}$ , como no algoritmo anterior e

$$b_k = \frac{-s_k^T r_{k+1}}{d_k} = \frac{(r_{k+1} - r_k)^T r_{k+1}}{C_k} = \frac{\|r_{k+1}\|^2 - r_k^T r_{k+1}}{\|r_k\|^2} \quad (13.3)$$

Dessa forma , escrevemos

$$\begin{aligned} \text{PASSO 1 : Iniciar com } x_1 ; \quad \text{sejam } r_1 &= -f'(x_1) & (14.1) \\ & \text{e } p_1 = r_1 \end{aligned}$$

PASSO 2 : se  $r_k = 0$  , PARE !  $x_k$  é ponto de ótimo .

$$\begin{aligned} \text{PASSO 3 : escolher } \alpha = a_k \text{ que minimiza } \Phi_k(\alpha) &= f(x_k + \alpha p_k) \\ \text{e fazer } x_{k+1} &= x_k + a_k p_k & (14.2) \\ r_{k+1} &= -f'(x_{k+1}) \end{aligned}$$

$k = k + 1$  ; se  $k > n$  , vá para o PASSO 1 .

PASSO 4 :

$$\begin{aligned} p_{k+1} &= r_{k+1} + b_k p_k \\ b_k &= \frac{\| r_{k+1} \|^2 - r_k^T r_{k+1}}{\| r_k \|^2} & (14.3) \end{aligned}$$

vá para o PASSO 2 .

Podemos escrever  $b_k = \frac{\| r_{k+1} \|^2}{\| r_k \|^2}$  , visto que

$$r_k^T r_{k+1} = r_k^T (r_k - a_k A p_k) = c_k - a_k d_k = 0 ,$$

pois por (13.1)  $c_k = \| r_k \|^2$  . No entanto , devido aos erros de arredondamentos a que o método ( C - G ) está sujeito , optamos

pela permanência da fórmula (14.3) para  $b_k$ . Além do mais, a condição de ortogonalidade acima depende exclusivamente de uma busca unidimensional eficiente para " $a_k$ ".

Ao aplicarmos o algoritmo (14.1) a (14.3) na resolução do "2º nível", devemos escolher um método de busca adequado para o problema e que seja eficiente, pois novas avaliações da função serão necessárias. Para isso decidimos trabalhar com o Método da FALSA POSIÇÃO [39] por ser um método que se "aproxima" do método de Newton, não necessitando de informações sobre derivada segunda da função.

### III.5 - TESTES COMPUTACIONAIS

Para os testes computacionais geramos alguns problemas, que serão antes descritos, de tal forma que pudéssemos avaliar o comportamento do algoritmo proposto, analisando a importância do tratamento dado ao "2º nível" e a influência do termo de penalização " $q$ " na sua convergência.

#### III.5.1 - Problemas-Teste

Todos os programas foram elaborados obedecendo a estrutura particular do problema (P.L.D.) apresentado no capítulo II. Dessa forma, descrevemos aqui apenas as matrizes e os vetores conforme o problema especificado, lembrando estar o ordenamento das variáveis de acordo com as colunas da matriz do sistema (vide pag.2).

(PT-1) N RESTRIÇÕES-DEMANDA / ESTÁGIO = M = 1



NÚMERO DE ESTÁGIOS (T) = 4

" " VARIÁVEIS DE ESTADO / ESTÁGIO = N = 2

" " " " CONTROLE / ESTÁGIO = R = 3

$$A = \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} ; B = \begin{bmatrix} -1 & 2 & 0 \\ 0 & 1 & -1 \end{bmatrix} \quad C = [ 2 \quad 1 ] \\ D = [ 1 \quad 2 \quad -1 ]$$

$$0 \leq u_1(0) \leq 20$$

$$0 \leq u_2(0) \leq 30$$

$$0 \leq u_3(0) \leq 50$$

$$0 \leq x_1(1), x_2(1) \leq 50$$

$$0 \leq u_1(1), u_2(1), u_3(1) \leq 50$$

$$0 \leq x_1(2), x_2(2) \leq 50$$

$$0 \leq u_1(2) \leq 40$$

$$0 \leq u_2(2) \leq 60$$

$$0 \leq u_3(2) \leq 30$$

$$0 \leq x_1(3), x_2(3) \leq 100$$

$$0 \leq u_1(3), u_2(3), u_3(3) \leq 100$$

$$0 \leq x_1(4), x_2(4) \leq 100$$

VECTOR -COEFICIENTE DA FUNÇÃO OBJETIVO:

(-3, 1, 3, 0, 5, 4, 7, -2, -3, 2, -1, -2, 3, 8, 6, 1, 8, -2, -1, 1)

VECTOR DAS RESTRIÇÕES :

(56.; 12.; 7.; 85.; 45.; 10.; 205.; 45.; 125.; 205.; 5.; 90.)

(PT-2) T = 8 ; N = 2 ; R = 3 ; M = 0

$$A = \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} ; B = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 2 & -1 \end{bmatrix}$$

$$0 \leq x_1 \leq 20$$

$$0 \leq x_2 \leq 30$$

$$0 \leq x_3 \leq 50$$

$$0 \leq x_4, \dots, x_{10} \leq 50$$

$$0 \leq x_{11} \leq 40$$

$$0 \leq x_{12} \leq 60$$

$$0 \leq x_{13} \leq 30$$

$$0 \leq x_{14}, x_{15} \leq 50$$

$$0 \leq x_{16}, \dots, x_{20} \leq 40$$

$$0 \leq x_{21}, \dots, x_{25} \leq 60$$

$$0 \leq x_{26}, \dots, x_{35} \leq 50$$

$$0 \leq x_{36}, \dots, x_{40} \leq 70$$

VETOR -COEFICIENTE DA FUNÇÃO OBJETIVO:

(-3, 1, 3, 0, 5, 4, 7, -2, -3, 2, -1, -2, 3, 8, 6, 1, 8, -2, -1, 3, -4, 1, -2, 2, 7, 5, -7, -1, -3, 0, 2, 6, -3, -1, -2, 2, 1, 7, 1, 4)

VETOR DAS RESTRIÇÕES :

(8., 27., 43., 70., 17., 75., 42., 106., 2., 46., 25., 40., 5., 45., 5., 110.)

(PT-3) T = 4 ; N = 3 ; R = 4 ; M = 0

$$A = I_9 \quad B = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & -1 & 0 & 1 \\ 2 & 0 & 1 & 1 \end{bmatrix}$$

$$0 \leq x_i \leq 40, \quad i=1, \dots, 7$$

$$0 \leq x_i \leq 70, \quad i=14, \dots, 20$$

$$0 \leq x_i \leq 50, \quad i=8, \dots, 13$$

$$0 \leq x_i \leq 60, \quad i=21, \dots, 28$$

VETOR -COEFICIENTE DA FUNÇÃO OBJETIVO:

(-3, 1, -6, 0, 5, -1, 7, -2, 0, 2, -1, -2, 3, 8, 6, 1, 8, -2, -1, 1, 0, 5, -3, 4, -1, 2, 3, 1)

VETOR DAS RESTRIÇÕES :

(30., 5., 80., 15., 20., 120., 65., 12., 65., 12., 220., 50., 18., 140.)

(PT-4) T = 10 ; N = 3 ; R = 5 ; M = 0

$$A = I_9 \quad B = \begin{bmatrix} 1 & -1 & 2 & -1 & 1 \\ 1 & 0 & -1 & 2 & 1 \\ 0 & 2 & 0 & 1 & 1 \end{bmatrix}$$

$$0 \leq x_i \leq 50, \quad i=1, \dots, 12$$

$$0 \leq x_i \leq 80, \quad i=41, \dots, 56$$

$$0 \leq x_i \leq 70, \quad i=13, \dots, 24 \quad 0 \leq x_i \leq 50, \quad i=57, \dots, 70$$

$$0 \leq x_i \leq 100, \quad i=25, \dots, 40 \quad 0 \leq x_i \leq 60, \quad i=71, \dots, 80$$

VETOR -COEFICIENTE DA FUNÇÃO OBJETIVO:

(4, -1, 0, 3, 7, -5, 1, -3, 8, 21, -10, 0, 5, -1, 3, 9, 6, -2, 2, 0, 25, 14, -1, 3, -6, -3, 1, 3, 0, 5, 4, 10, -2, -7, 2, -3, -12, 3, 8, 5, 1, -4, 8, -2, -1, 3, -4, 1, -2, 2, 7, 15, -7, -1, 3, 0, 2, 8, -3, -1, -2, 2, 3, 10, 1, -4, 0, 2, -3, 1, 6, 8, -1, 3, 0, 0, 4, -3, 5, 15)

VETOR DAS RESTRIÇÕES :

(95., 10., 35., 105., 55., 80., 28., 97., 38., 5., 203., 200., 35., 165., 250., 11., 64., 36., 35., 70., 190., 83., 95., 143., 45., 23., 5., 10., 80., 140.)

(PT-5)      T = 20 ;   N = 5 ;   R = 5

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

$$0 \leq x_i \leq 20, \quad i=1, \dots, 10 \quad 0 \leq x_i \leq 20, \quad i=81, \dots, 110$$

$$0 \leq x_i \leq 30, \quad i=11, \dots, 30 \quad 0 \leq x_i \leq 50, \quad i=111, \dots, 130$$

$$0 \leq x_i \leq 15, \quad i=31, \dots, 50 \quad 0 \leq x_i \leq 70, \quad i=131, \dots, 160$$

$$0 \leq x_i \leq 10, \quad i=51, \dots, 80 \quad 0 \leq x_i \leq 60, \quad i=161, \dots, 180$$

$$0 \leq x_i \leq 30, \quad i=181, \dots, 200$$

VETOR -COEFICIENTE DA FUNÇÃO OBJETIVO:

(12, 3, -6, 9, 1, 1, 1, 1, 3, -2, 1, -2, -10, 0, 0, 3, 2, 5, 10, -4, 3, 4, 5, 1, 1, 2, 7, 10, -2, 5, -3, 9, 0, 12, 2, 0, 3, 1, 1, 1, 6, -2, 2, 0, -3, -7, 0, 5, 2, 15, 1, 8, 3, 13, 0, -1, -1, 4, 7, 1, 4, -2, -7, 14, 0, 3, 3, -2, -7, 2, 5, -2, 14, 9, 1, 8, 12, 1, -3, 11, -4, 0, 10,

3,1,2,-4,5,1,7,1,-9,4,1,3,0,0,6,-3,1,12,22,31,-10,2,4,3,10,22,2,0,  
16,0,-5,7,-2,-7,0,4,0,-1,3,14,9,2,-3,5,11,21,-2,1,4,4,2,1,0,-5,2,  
14,6,-7,-3,0,0,9,22,5,0,-12,-3,16,1,-13,12,0,7,2,-1,-1,2,6,-2,-7,  
0,21,-14,25,1,1,6,3,0,6,2,9,3,-4,0,9,1,-1,-1,5,2,8,4,8,-5,-1,3,1,  
10,-1,-5,4,14,0,0,12,-5)

VETOR DAS RESTRIÇÕES : (-9.,-25.,-13.,5.,0.,10.,12.,-12.,-4.,9.,  
3.,-2.,-10.,-2.,1.,-4.,-19.,-6.,12.,5.,3.,9.,-6.,1.,-1.,16.,5.,  
-5.,-8.,11.,-1.,10.,2.,-4.,4.,3.,2.,8.,12.,-5.,1.,-7.,-10.,-17.,  
25.,-15.,8.,-22.,-2.,17.,9.,-3.,-5.,-10.,5.,7.,-16.,-8.,14.,-23.,  
10.,-11.,-16.,33.,-8.,28.,26.,30.,-5.,14.,7.,5.,22.,-11.,-42.,-7.  
18.,64.,90.,28.,33.,3.,-30.,2.,5.,39.,39.,-17.,3.,7.,20.,-12.,22.  
22.,2.,44.,46.,33.,23.,15.)

### III.5.2 - Resultados Computacionais

Foi elaborado programa para o algoritmo do Lagrangeano Aumentado na resolução do problema (P.L.D.) tratando os "1º e 2º níveis" conforme descrevemos nas seções III.3 e III.4 e o "3º nível" conforme procedimento descrito em III.4.2.

Tabelas apresentadas abaixo mostram os dados obtidos sobre o comportamento do algoritmo na versão "DICON" para os problemas-testes descritos na seção anterior.

A linguagem utilizada é o FORTRAN e os testes realizados em um computador DEC system-10.

O programa "DICON" utiliza as seguintes subrotinas:

HESS - subrotina que calcula a matriz hessiana da função Lagrangeano Aumentado de problemas do tipo (P.L.D.) nos tempos  $t = 0, \dots, T-1$  ;

HEO - subrotina que atualiza a matriz hessiana do Lagrangeano Aumentado para o primeiro estágio a cada atualização do multiplicador de Lagrange ;

HENE - subrotina que calcula a matriz hessiana da função Lagrangeano Aumentado no último estágio ;

VEB - subrotina que calcula a parte linear da forma quadrática da função Lagrangeano Aumentado por estágio de tempo ;

GRACON - subrotina que otimiza a função Lagrangeano Aumentado a cada estágio de tempo; é uma implementação do algoritmo (C-G-C), seção III.3.1, para funções quadráticas com variáveis "canalizadas" ;

CONJUG - é a implementação do algoritmo (G-C), seção III.4, para resolução do "2º nível" ;

BUSCA - é a implementação do algoritmo da "Falsa Posição" para efetuar a busca unidimensional no "2º nível" ;

HAJACO - coordena as chamadas das demais subrotinas do programa.

Nas tabelas a seguir utilizamos a seguinte nomenclatura :

Q : valor atribuído ao fator de penalização .

ITER : número de iterações necessárias no " 3º nível " ( ou atualizações do multiplicador de Lagrange ) para a

convergência do algoritmo .

AV.V : número total de iterações gastos no "2<sup>o</sup> nível " até a convergência do algoritmo .

CPU : tempo computacional gasto para a convergência do algoritmo em segundos

ND : número de variáveis de estado do problema;

RD : número de variáveis de controle do problema;

ND+MD : número de variáveis duais do problema;

(+) : atingiu o número máximo de iterações fixado para a busca unidimensional do "2<sup>o</sup> nível" ( da ordem de 3ND );

(\*) : atingiu o número máximo de iterações fixado para a convergência do "1<sup>o</sup> nível" ( da ordem de 2(ND + RD) );

(\*\*) : atingiu o número máximo de iterações fixado para a convergência do "3<sup>o</sup> nível" ( da ordem de 4(ND+MD) );

(\*\*\*) : atingiu o número máximo de iterações fixado para a convergência do "2<sup>o</sup> nível" ( da ordem de 2ND ).

TABELA 1 : TESTES COM ( PT-1 )

D I C O N						
Q	.2	.3	.5	1.	2.	
AV.V.	33	37	43	43	50	
ITER	08	05	06	02	02	
CPU	4.54	10.08	9.06	14.70	15.47	

Inicialização: Multiplicador de Lagrange igual ao vetor nulo.

OBSERVAÇÃO :

Claramente, percebemos a importância do fator de penalização no comportamento dos vários níveis do algoritmo : para valores maiores deste fator aceleramos a convergência do "3º nível" dificultando, porém, a convergência do "2º nível" , fato que ocasiona um gasto computacional que cresce à medida que cresce o valor de "q" .

TABELA 2 - TESTES COM ( PT-2 )

Inicialização: Multiplicador de Lagrange igual ao vetor nulo.

D I C O N						
Q	.05	.5	1.	1.5	2.5	3.0
AV.V.	44	40	35	37	50	47
ITER	88	06	04	03	03	02
CPU	7.9	5.98	6.29	8.14	9.68	19.27

OBSERVAÇÃO :

Neste caso constatamos a influência do fator de penalização também para o "1º nível", como ocorre com o programa DICON ; o crescimento do fator "q" acelera a convergência do "3º nível", mas aumenta o número de "CPU".

TABELA 3 - TESTES COM ( PT-3 )

Inicialização: Multiplicador de Lagrange igual ao vetor nulo.

		D I C O N					
Q		.1	.5	1.5	2.5	3.	3.5
AV.V.			22	28	23		
ITER	(**)		24	09	06	(+)	(+)
CPU			5.49	7.26	9.40		

OBSERVAÇÃO :

A influência da penalização "q" na convergência do algoritmo continua como fator preponderante, permanecendo a observação sobre o tempo de "CPU".

TABELA 4 - TESTES COM ( PT-4 )

Inicialização: Multiplicador de Lagrange igual ao vetor nulo.



D I C O N						
Q	.007	.012	.05	.1	.5	1.
AV.V.	49	45	51	54	40	57
ITER	132	83	42	31	05	04
CPU	21.8	16.8	17.74	23.13	21.33	30.98

OBSERVAÇÃO :

Notamos novamente a influência do fator "q" na convergência do algoritmo, sendo que para  $q = 0.012$  atingiu o seu valor ótimo em termos de "CPU".

TABELA 5 - TESTES COM ( PT-5 )

Inicialização: Multiplicador de Lagrange igual ao vetor nulo.

D I C O N						
Q	.012	.1	.5	1.	2.	3.
AV.V.	320	313	269	260	251	
ITER	36	25	17	13	09	(***)
CPU	6:38.4	6:40.1	6:07.3	5:53.8	5:58.7	

## OBSERVAÇÕES:

Observamos, neste caso, que valores crescentes do fator de penalização afetam a convergência do "2º nível" do algoritmo, comprometendo o bom desempenho de "DICON". Todavia, para valores menores da penalidade o algoritmo converge demonstrando estabilidade em seu tempo de GPU.

## III.6 - CONCLUSÕES

Em todos os testes realizados pudemos verificar que valores "grandes" para o fator "q" aceleram a convergência do erro nas restrições ( ou, no nosso caso, a convergência do "3º nível" ), mas, em contrapartida, provocam "mal comportamento" das funções trabalhadas nos 1º e 2º níveis, causando maiores custos operacionais ao algoritmo.

Dos resultados apresentados, concluimos

- 1º : convergência rápida no "3º nível" não significa eficiência do algoritmo em termos operacionais ( GPU );
- 2º : existe o valor  $q^*$  para o fator de penalidade que traz o melhor tempo computacional para o algoritmo;
- 3º : comprovando a teoria do capítulo II, seção II.4, o "3º nível" converge, mesmo para valores pequenos de "q" ;

Como já tínhamos previsto anteriormente, a convergência do "2º nível" é extremamente importante para o sucesso da convergência do algoritmo; por isso, o tratamento utilizado para a sua resolução é a "chave" para a boa eficiência do Lagrangeano Aumentado. Ao concluirmos sobre as condições de diferenciabilidade da função (FV) (vide seção 11.5.1) isto nos possibilitou adotarmos para a resolução deste nível um método que nos garantiu a aceleração de sua convergência, apesar da "busca unidimensional" exigir novas avaliações dessa função (ou seja, novos retornos ao "1º nível"). Porém, à medida que obtemos valores para as variáveis primais próximos de seus valores ótimos, a convergência do "1º nível" é bastante rápida, possibilitando um bom desempenho para DICON.

Além do mais,

- 4ª : os testes realizados mostram que, assim como já demonstrado para problemas não-lineares (vide APENDICE 2), crescimento para o fator de penalidade  $Q$  implica em melhoria da taxa de convergência do método no nível dual;
- 5ª : o programa DICON demonstrou que o método do Lagrangeano Aumentado pode ser utilizado na resolução de problemas do tipo (P.L.D.) com eficiência computacional, apresentando bons resultados para valores  $q^*$  ;

Nosso objetivo neste trabalho foi estudarmos a viabilidade de tratarmos problemas do tipo (P.L.D.) através de métodos duais que apresentam características de decomposição.

Não podemos deixar de salientar que os dados com relação ao tempo de CPU verificados nos vários testes com DICON nos indicaram para a viabilidade da utilização do método do Lagrangeano Aumentado na resolução de problemas lineares dinâmicos, principalmente aqueles conseguidos através de  $q^*$ .

## SUGESTÕES :

Vimos, através dos resultados computacionais apresentados, que o fator de penalidade exerce grande influência sobre o desempenho computacional do algoritmo e que o melhor valor a ele atribuído não é, necessariamente, aquele que traz menor número de iterações para o "3<sup>o</sup> nível".

O algoritmo do Lagrangeano Aumentado converge para quaisquer valores fixados para o fator "Q" ; porém, numericamente, valores muito grandes de Q podem causar "mal comportamento" para as funções dos "1<sup>o</sup> e 2<sup>o</sup> níveis", bem como valores muito pequenos de Q causam excessivo número de iterações para o "3<sup>o</sup> nível".

Esses fatos nos levam a propor políticas para atualização do valor da penalização durante o procedimento iterativo do método com o objetivo de podermos alcançar melhoria dos tempos computacionais conseguidos. Para isso, sugerimos que :

O valor da penalização "Q" seja atualizado a cada iteração do algoritmo de modo a tornar possível a utilização de valores grandes em pontos próximos da solução. Dessa forma torna-se possível inicializar o procedimento com valores pequenos de "Q", diminuindo os riscos de surgimento de problemas numéricos e garantindo a convergência do algoritmo.

A implementação de métodos de otimização, tais como Gradiente ordinário ou Gradientes Conjugados, para a maximização da função dual não significa, necessariamente, melhoria nos tempos computacionais obtidos ( mesmo para aqueles cujo valor de "Q" é

pequeno ) tendo em vista, para isso, a necessidade de utilização de busca unidimensional e, conseqüentemente, de novas resoluções dos "1º e 2º níveis", o que pode causar maior carga computacional ao algoritmo.

## DECOMPOSICAO VIA LAGRANGEANO SIMPLES

## IV.1 - INTRODUCAO

Como já destacamos anteriormente, nosso trabalho tem como objetivo estudar a resolução de problemas do tipo (P.L.D.) através de métodos "indiretos".

O propósito da utilização desta técnica vem da possibilidade de obtermos, de maneira simples e rápida, a solução ótima primal via sistemas de porte bem menor do que o apresentado pelo problema original

Como já ressaltamos no Capítulo II, ao obtermos a solução dual ótima de um problema linear, deparamo-nos com o problema da indeterminação da solução ótima primal associada. A rigor, a informação sobre a solução ótima dual só permite especificar os valores das variáveis não-básicas primais, restando ainda o nem sempre imediato trabalho de obtermos os valores das variáveis básicas da solução ótima primal. Entretanto, no caso de problemas lineares dinâmicos, como veremos, a informação sobre a solução dual ótima permite a obtenção decomposta no tempo da solução primal correspondente, o que torna esta abordagem potencialmente atraente.

O bom desempenho da resolução do problema dual do (P.L.D.), a qual denominamos "1ª FASE", é importante para a técnica aqui proposta, tendo em vista esta depender da obtenção da solução ótima dual.

Esta fase resulta da resolução do problema dual do (P.L.D.) via "Aproximação Tangencial". Como já é sabido, este método

envolve um coordenador, que é um problema linear cujas restrições são formadas por hiperplanos-suportes à função dual, e subproblemas que, neste caso, apresentam soluções analíticas a cada estágio de tempo, fato que traz grandes vantagens computacionais ao método.

Para problemas com muitas variáveis é esperado que o método gere um número grande de restrições, sendo necessário desprezarmos aquelas não-ativas durante seu processo iterativo. Em geral, detectar quais das restrições ( ou hiperplanos ) geradas que não sendo ativas podem ser eliminadas durante o processo de resolução é um trabalho oneroso, dependendo do problema a ser resolvido.

Por ser um método que utiliza hiperplanos-suporte à função tratada, a "Aproximação Tangencial" enquadra-se às nossas necessidades devido ao fato da função dual do (P.L.D.) ser uma função côncava linear-por-partes.

A técnica da Aproximação Tangencial ( Planos-de-Corte ) surgiu com Kelley [33] como uma técnica de decomposição, cuja estratégia de resolução é a Relaxação. Geoffrion [22] apresentou estudos da técnica em 1967 , cuja versão é utilizada em nosso trabalho.

Na seção IV.2 apresentamos o método da Aproximação Tangencial e sua adaptação a problemas do tipo (P.L.D.), a estratégia utilizada na implementação da Relaxação e inicialização do processo. Na seção IV.3 tratamos sobre o procedimento para obtenção da solução ótima primal e o problema da caracterização do sistema a ser resolvido, apresentando uma análise sobre o comportamento geral das soluções de problemas lineares dinâmicos.

#### IV.2 - METODO DA APROXIMACAO TANGENCIAL

Faremos, em primeiro lugar, uma breve exposição do método para



problemas gerais e posteriormente sua adaptação ao problema (P.L.D.) tratado neste trabalho.

Seja o problema primal descrito na forma

$$(P) \quad \begin{aligned} & \min f(x) \\ & \text{s.a. } g(x) = 0 \\ & x \in S \end{aligned}$$

onde  $f$  e  $g$  são funções convexas definidas num espaço  $n$ -dimensional e  $S$  um conjunto convexo.

O problema dual de (P) é dado por

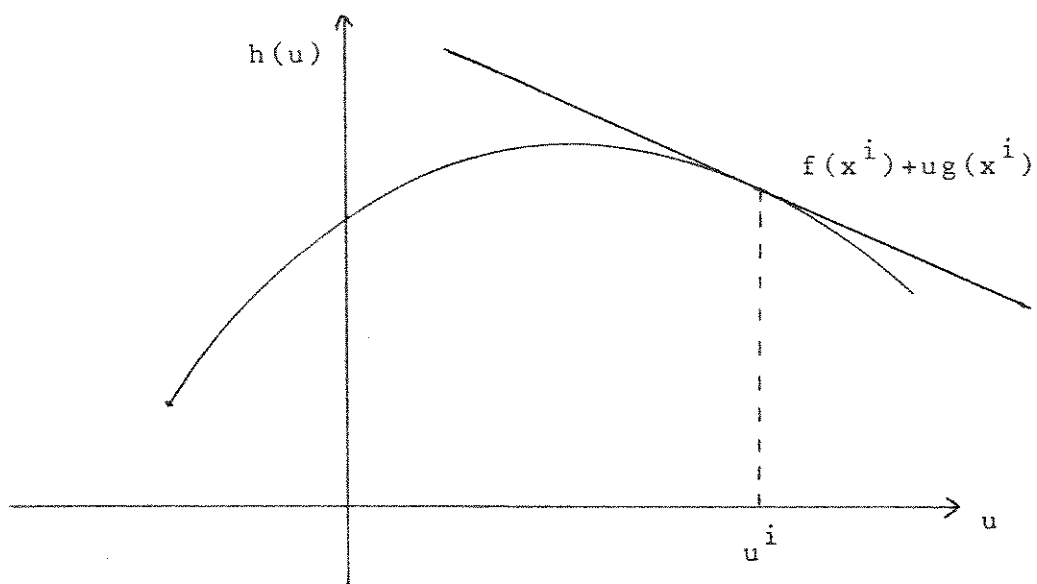
$$(D) \quad \max_u h(u)$$

onde  $h(u) = \inf_{x \in S} [ f(x) + u g(x) ]$  é a função dual associada ao mesmo problema (P).

Para "h" diferenciável em  $u$ , temos que [4,38,39]  $g(x)$  é o seu gradiente para  $x \in X(u)$ . Dessa forma, ao tomarmos  $u = u^l$ , a equação

$$h(u^l) + (u - u^l)g(x^l) = f(x^l) + u g(x^l) \quad (1)$$

onde  $x^l \in X(u^l) = \{ x \in S / x \text{ minimiza } f(x) + u^l g(x) \}$ , descreve o hiperplano-suporte à função  $h$  no ponto  $u^l$ . Se  $h$  é não-diferenciável em  $u^l$  temos que  $g(x^l)$  é subgradiente de  $h$  neste ponto [4,38] e a análise feita para (1) é válida.



( figura 4.1 )

Seja  $h^r(u)$  a representação da função dual através de sua linearização externa

$$h^r(u) = \inf_j \{ f(x^j) + u g(x^j) \}$$

onde  $x^j$  é solução do problema Lagrangeano  $\min_{x \in S} \{ f(x) + u^j g(x) \}$  com  $u^i$  pertencente ao domínio de  $h$ , para todo  $j$ .

Pelo fato de ser o infimo o maior limitante inferior, o problema (D) é equivalente ao problema

$$\begin{array}{ll}
 \text{(PM)} & \max_{u, z^0} z^0 \\
 & \text{s.a. } z^0 \leq f(x^j) + u g(x^j), \quad x^j \in X(u^j), \quad \forall j
 \end{array}$$

Dado que para problemas estritamente convexos o número de restrições de (PM) pode se apresentar elevado, a estratégia de resolução adequada para (PM) é a Relaxação: logo, o problema a ser resolvido é

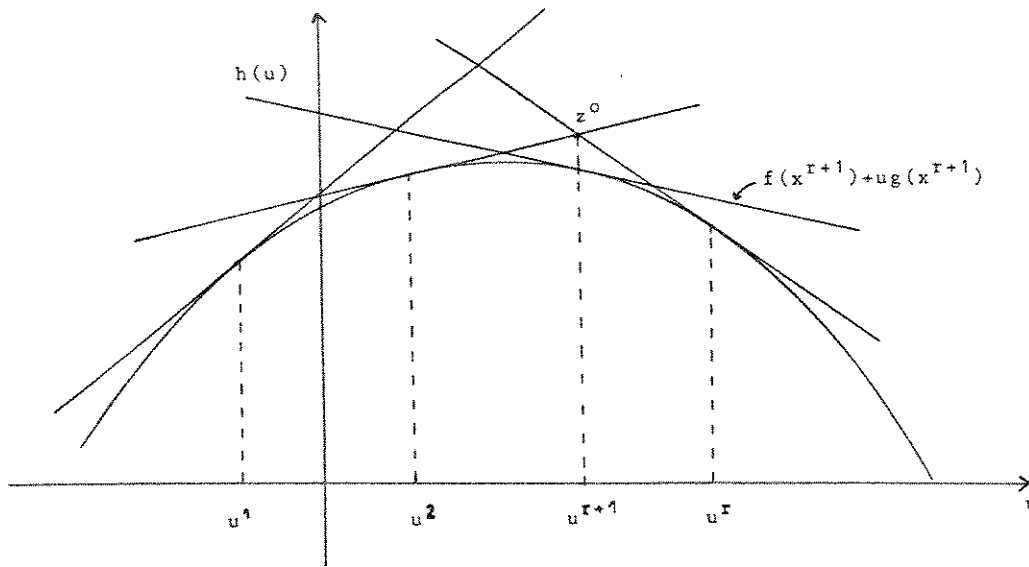
$$\begin{array}{ll}
 \text{(PM)}_r & \max_{u, z^0} z^0 \\
 & \text{s.a. } z^0 \leq f(x^j) + u g(x^j), \quad x^j \in X(u^j) \\
 & \quad \quad \quad j = 1, \dots, r
 \end{array}$$

que pode ser entendido como o problema (PM) com muitas restrições "relaxadas".

Em resumo, o procedimento iterativo do método pode ser descrito através dos seguintes "passos":

- PASSO1    Seja  $\{x^j\}_{j=1}^r$  tal que  $x^j \in X(u^j)$ ;
- PASSO2    Resolver  $(PM)_r$ ; seja  $u^{r+1}$  a solução ótima e  $z^0$  o valor ótimo de  $h^{r+1}$ ;
- PASSO3    Minimizar  $\{f(x) + u^{r+1}g(x)\}$  sobre  $S$ ;  
           Seja  $x^{r+1}$  uma solução.
- PASSO4    Se  $h(u^{r+1}) = z^0$ , pare -  $x^{r+1}$  é solução ótima;  
           caso contrário, introduza a restrição  
 $z^0 \leq f(x^{r+1}) + u g(x^{r+1})$  em  $(PM)_r$  e vá para PASSO2.

Uma visão geométrica deste procedimento está na figura abaixo



( figura 4.2 )

Como podemos verificar , o método da Aproximação Tangencial pode ser visto como o dual da técnica de decomposição de Dantzig - Wolfe ( Programação Linear Generalizada - P.L.G. ) [14]. Para este último método, o problema (P) é resolvido iterativamente pela sequência de problemas do tipo

$$\min \sum_{j=1}^r \lambda_j f(x^j) \quad (2)$$

$$\text{s.a.} \sum_{j=1}^r \lambda_j g(x^j) = 0 \quad (3)$$

$$\sum_{j=1}^r \lambda_j = 1 \quad , \quad \lambda_j \geq 0 \quad (4)$$

onde  $x^j$  são pontos extremos de  $S$ , supondo  $S$  limitado.

Sendo  $u^{r+1}$  vetor ótimo dual associado à restrição (3), uma nova coluna para o problema (2) - (4) é gerada a cada iteração através da resolução do subproblema

$$\min_{x \in S} f(x) + u^{r+1}g(x)$$

visto que  $x^{r+1} \in X(u^{r+1})$  é somada ao conjunto dos pontos extremos de  $S$ . Notamos que o subproblema acima é o problema Lagrangeano do PASSO3 do algoritmo da Aproximação Tangencial.

Dessa forma, se chamarmos " $z^o$ " a variável dual associada à restrição (4), então o problema dual de (2) - (4) é o problema  $(PM)_r$ .

Dantzig [14] provou convergência para o procedimento da P.L.G. desde que sejam mantidas todas as colunas geradas pelo método. A interpretação dual deste resultado é que nenhum hiperplano-suporte gerado pela "Aproximação Tangencial" deve ser descartado. Do ponto de vista computacional, esta hipótese é impraticável pois a matriz de restrições de  $(PM)_r$  pode apresentar-se muito grande. Veremos na seção IV.2.4. que é possível "desprezarmos" as restrições que não estão ativas no ponto-solução apontado por  $(PM)_r$  sem afetarmos a convergência do método.

Desde que  $u^{r+1}$  é factível para o problema dual (D),  $h^{r+1}$  é um limitante inferior para o valor ótimo  $h(u^*)$  e desde que  $z^o$  é um limitante superior para o mesmo valor, a condição  $h(u^{r+1}) = z^o$  nos garante que  $h(u^{r+1})$  é o maior limitante inferior da função dual e, portanto,  $u^{r+1}$  é a solução de (D).

Podemos ainda observar que  $z^o$  decresce monotonicamente, tendo em vista que a estratégia empregada pelo método na resolução do problema-mestre  $(PM)_r$  é a Relaxação (vide seção

IV.2.4), embora isso não ocorra com os valores de  $h(u^r)$ .

#### IV.2.1 - Adaptação para (P.L.D.)

Dado que estamos trabalhando com problemas cuja função objetivo e restrições são lineares e cujas variáveis são "canalizadas", as condições de convexidade para o problema estão garantidas. Logo, a adaptação do procedimento exposto na seção anterior ao problema (P.L.D.) é direta.

Para tanto, devemos considerar a função Lagrangeano e a função dual (FD) associadas ao (P.L.D.) conforme (4.1) da seção 11.3., ou seja,

$$(FD) \quad \phi(p, \lambda) = \min_{x, u \in S} L(x, u, p, \lambda)$$

cujas variáveis  $p$  e  $\lambda$  são irrestritas e  $L(x, u, p, \lambda)$  é aditivamente separável por estágio de tempo,

Para explicitarmos o problema-mestre equivalente, consideramos a seguinte notação:

$$f(x, u) = \sum_{t=0}^{T-1} c(t+1)x(t+1) + d(t)u(t)$$

$$p \ g(x, u) = \sum_{t=0}^{T-1} p(t+1) [x(t+1) - A(t)x(t) - B(t)u(t)]$$

$$\lambda \ h(x, u) = \sum_{t=0}^{T-1} \lambda(t) [f(t) - C(t)x(t) - D(t)u(t)]$$

Por (4), seção 11.3, podemos escrever

$$L(x, u, p, \lambda) = f(x, u) + p g(x, u) + \lambda h(x, u), \quad \text{ou seja,}$$

$$\phi(p, \lambda) = \min_{x, u \in S} \{ f(x, u) + p g(x, u) + \lambda h(x, u) \}$$

Assim, a função dual linearizada do (P.L.D.) pode ser representada na forma

$$\phi^r(\lambda, p, q) = \inf_j \{ f(x^j, u^j) + p g(x^j, u^j) + \lambda h(x^j, u^j) \}$$

sendo  $(x^j, u^j) \in X(p^j, \lambda^j)$ , ou seja,  $(x^j, u^j)$  é solução do problema

$$\min_{x, u \in S} L(x, u, p^j, \lambda^j) .$$

Portanto, podemos escrever o problema (DLD), seção 11.3, na sua forma adaptada a (P.L.D.) :

$$\begin{aligned} & \max_{p, \lambda, z^0} z^0 \\ \text{(PM)}_{LD} \quad & \text{s.a. } z^0 \leq f(x^j, u^j) + p g(x^j, u^j) + \lambda h(x^j, u^j) \\ & (x^j, u^j) \in X(p^j, \lambda^j) \end{aligned}$$

e o problema relaxado  $(PM_{LD})^r$  como  $(PM)_{LD}$  para  $j=1, \dots, r$

Tal como na seção anterior, o procedimento iterativo do método da Aproximação Tangencial para problemas do tipo (P.L.D.) é dado

por:

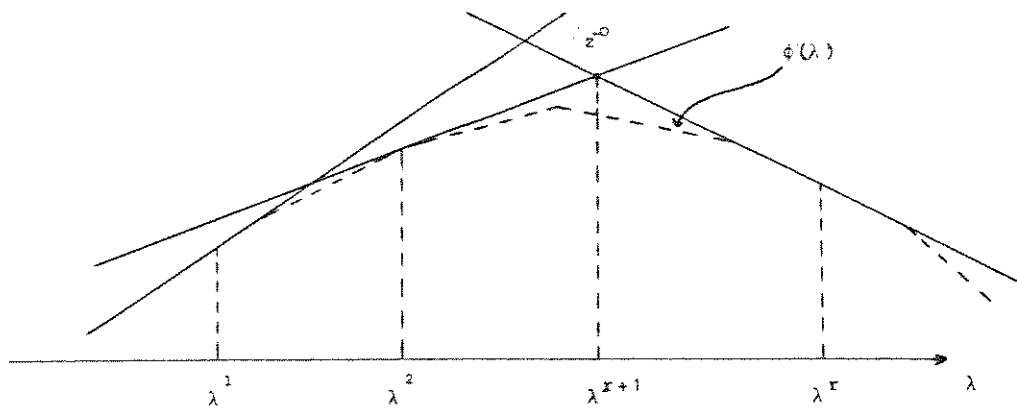
- PASS01' Seja  $\{(x^j, u^j)\}_{j=1}^r$  tal que  $(x^j, u^j) \in X(p^j, \lambda^j)$
- PASS02' Resolver  $(PM_{LD})^r$ ; seja  $(p^{r+1}, \lambda^{r+1})$  a solução e  $z^o$  o valor ótimo de  $\phi^{r+1}$
- PASS03'  $\min_{x, u \in S} L(x, u, p^{r+1}, \lambda^{r+1})$ . Seja  $(x^{r+1}, u^{r+1}) \in X(p^{r+1}, \lambda^{r+1})$
- PASS04' Se  $\phi^{r+1} = z^o$ , pare!  $(p^{r+1}, \lambda^{r+1})$  é solução ótima. Caso contrário, introduzir a restrição  $z^o \leq f(x^{r+1}, u^{r+1}) + p g(x^{r+1}, u^{r+1}) + \lambda h(x^{r+1}, u^{r+1})$  em  $(PM_{LD})^r$ , faça  $r = r + 1$  e vá para PASS02'.

O algoritmo descrito acima nos indica que a cada iteração devemos resolver, no PASS04', um problema linear que é o problema da iteração anterior acrescido de uma nova restrição. Consequentemente, caso sejam geradas muitas restrições para atingir convergência, o problema  $(PM_{LD})^r$  pode apresentar-se de grande porte, sendo necessária a utilização de técnicas que viabilizem computacionalmente o método. Trataremos desse problema nas seções precedentes.

Ao contrário do PASS04', o PASS03' oferece uma operacionalização rápida e simples, dado ser um problema de otimizar uma função linear, separável no tempo e cujas variáveis são "canalizadas". Sua resolução é semelhante à descrita na seção IV.3.2.

Dado que no PASS04' do algoritmo acima é gerada apenas uma restrição e sabendo que a função dual do problema (P.L.D.) é côncava linear-por-partes, a visão geométrica do procedimento iterativo acima é dada através da figura abaixo.





( figura 4.3 )

Pelos pontos-de-quebra da função dual passam infinitos hiperplanos tangentes: porém, a escolha arbitrária de um deles no processo de linearização do método não acarreta problemas sobre a convergência do algoritmo, dado que a monotonicidade do valor de " $z^0$ " está garantida pela estratégia da Relaxação (vide IV.2.4).

Três aspectos teóricos-computacionais devem ser aqui destacados: a estrutura da matriz de restrições do  $(PM_{LD})^r$ , a maneira com que inicializamos o processo de geração dessas restrições e a técnica da Relaxação como estratégia de resolução do problema-mestre.

#### IV.2.2 - Geração das Restrições

A cada iteração do método somente uma nova restrição deve ser acrescentada a  $(PM_{LD})^r$ , restrição que vimos ser da forma

$$z^o \leq f(x^j, u^j) + p g(x^j, u^j) + \lambda h(x^j, u^j) \quad (5)$$

Seja  $(x^o, u^o) \in X(p^o, \lambda^o)$  uma solução do problema Lagrangeano (1) para  $(p^o, \lambda^o)$  iniciais. Então,

$$z^o \leq f(x^o, u^o) + p g(x^o, u^o) + \lambda h(x^o, u^o) \quad (6)$$

é a restrição inicial gerada para  $(PM_{LD})^1$ ; ou ainda,

$$z^o = f(x^o, u^o) + p g(x^o, u^o) + \lambda h(x^o, u^o) + \lambda_o \quad (6.1)$$

onde  $\lambda_o$  é variável de folga para (6).

Seja  $(x^1, u^1) \in X(p^1, \lambda^1)$ , onde  $(p^1, \lambda^1)$  é solução de  $(PM_{LD})^1$ ; logo, a segunda restrição gerada para  $(PM_{LD})^2$  é

$$z^o \leq f(x^1, u^1) + p g(x^1, u^1) + \lambda h(x^1, u^1) \quad (7)$$

$$\text{ou} \quad z^o = f(x^1, u^1) + p g(x^1, u^1) + \lambda h(x^1, u^1) + \lambda_1 \quad (7.1)$$

onde  $\lambda_1$  é a variável de folga para (7).

Ao adotarmos o valor de  $z^o$  sendo determinado por (6.1), obtemos da equação (7.1)

$$\begin{aligned} f(x^1, u^1) - f(x^o, u^o) &= \\ &= p [g(x^o, u^o) - g(x^1, u^1)] + \lambda [h(x^o, u^o) - h(x^1, u^1)] + \lambda_o + \lambda_1 \end{aligned}$$

que é, por nós considerada, a primeira linha da matriz das restrições do problema relaxado  $(PM_{LD})^2$ .

Seguindo este raciocínio, estamos considerando que a  $J$ -ésima linha da matriz das restrições de  $(PM_{LD})^r$ ,  $r > J$ , é da forma

$$\begin{aligned} f(x^j, u^j) - f(x^0, u^0) &= \\ &= p [g(x^0, u^0) - g(x^j, u^j)] + \lambda [h(x^0, u^0) - h(x^j, u^j)] + \lambda_0 + \lambda_j \end{aligned} \quad (8)$$

onde  $(x^j, u^j) \in X(p^j, \lambda^j)$  e  $\lambda_j$  é a variável de folga da  $J$ -ésima restrição gerada

$$z^0 \leq f(x^j, u^j) + p g(x^j, u^j) + \lambda h(x^j, u^j)$$

Dessa forma, na  $(k + 1)$ -ésima iteração do processo, a matriz do problema  $(PM_{LD})^{k+1}$ , possui dimensão igual a  $k \times ((n+m)T + k + 1)$ , podendo ser escrita na forma  $[MV \ f \ I]$ , onde  $MV$  é a matriz de ordem  $k \times (n+m)T$ , " $f$ " é o vetor-coluna relacionado à variável de folga " $\lambda_0$ " e  $I$  é a matriz identidade de ordem  $k \times k$  relacionada às variáveis-de-folga  $\lambda_i$ .

#### IV.2.3 - Inicialização do Processo

Antes de adotarmos algum critério para a inicialização do processo de geração das restrições, precisamos nos ater para o fato de que a função dual a ser tratada é uma função cujas variáveis, os multiplicadores de Lagrange, são livres (conforme exposto na seção 11.2); logo, para que  $(PM_{LD})^r$  seja um problema com solução limitada, devemos inicialmente limitar o domínio da função dual de alguma forma até que o próprio processo de otimização passe a

fornecer soluções limitadas.

É conveniente adotarmos limites iniciais para as variáveis duais tais que o poliedro gerado por eles determine um espaço suficientemente pequeno. Como, a priori, as variáveis duais podem assumir quaisquer valores, propomos o seguinte procedimento heurístico para inicialização do algoritmo :

#### HEURÍSTICA PARA INICIALIZAÇÃO

PASS01 : Seja  $(p^0, \lambda^0)$  o par das variáveis duais iniciais.

Faça  $k = 0$  e defina  $(\Delta p, \Delta \lambda)$  de modo que

$$p^k - \Delta p \leq p \leq p^k + \Delta p \quad \text{e} \quad \lambda^k - \Delta \lambda \leq \lambda \leq \lambda^k + \Delta \lambda \quad (1)'$$

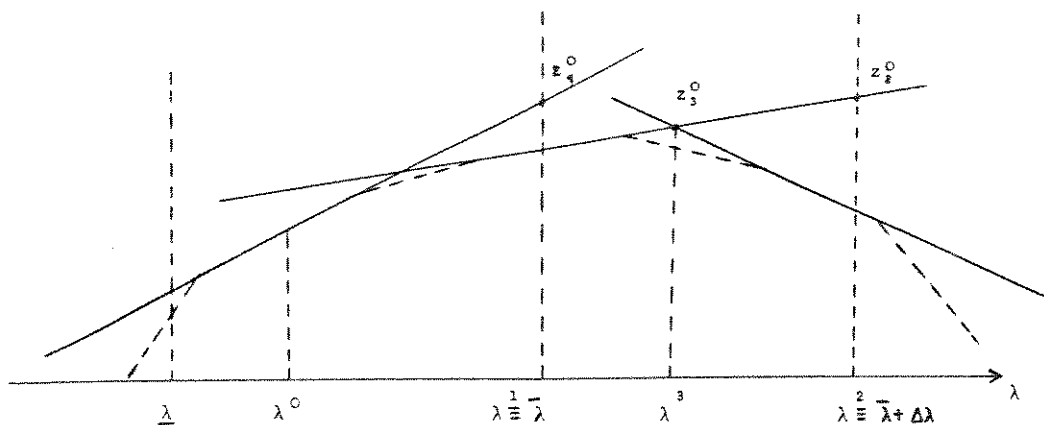
PASS02 : Aplicando o procedimento da Aproximação Tangencial, resolver  $(PM_{LD})^r$  acrescentado das restrições (1)'; seja  $(p^*, \lambda^*)$  a solução encontrada.

PASS03 : Se a solução  $(p^*, \lambda^*)$  for tal que  $p^k - \Delta p < p^* < p^k + \Delta p$   
 $\lambda^k - \Delta \lambda < \lambda^* < \lambda^k + \Delta \lambda$ , então  $(p^*, \lambda^*)$  é solução de  $(PM_{LD})^r$   
e solução dual de (P.L.D.) ; PARE.

Se a solução  $(p^*, \lambda^*)$  for ativa em algum dos limites fixados, vá para PASS04.

PASS04 : Faça  $p^0 = p^*$  e  $\lambda^0 = \lambda^*$  e vá para PASS02 aproveitando todas as restrições de linearização já geradas nas iterações anteriores.

A figura abaixo resume este processo de geração.



( figura 4.4 )

Devemos notar que se o procedimento da Aproximação Tangencial não converge para determinados limites duais fixados, significa que a solução ótima de  $(PM)_{LD}$  está fora do poliedro gerado por tais limites. Ao reinicializarmos o processo (PASSO4) a monotonicidade de " $z^o$ " deve ser alterada, devendo ser reconsiderada a partir daquele instante. Como a função dual é côncava, podemos esperar que num número finito de passos o procedimento proposto acima determina o poliedro que contém a solução ótima do problema em questão.

#### IV.2.4 - A Técnica da Relaxação

Como sabemos, o método da Aproximação Tangencial utiliza como estratégia de resolução para o problema-mestre  $(PM)_{LD}$  a

Relaxação.

Nesta seção definimos a técnica da Relaxação, suas propriedades e sua aplicabilidade a  $(PM_{LD})$ . Encontramos em Lasdon [38] os resultados que seguem.

Seja o problema

$$(P) \quad \begin{array}{l} \max f(x) \\ \text{s.a. } g_i(x) \geq 0, \quad i=1, \dots, m \\ x \in S \end{array}$$

onde "x" é vetor n-dimensional, "f" e "g<sub>i</sub>" são funções côncavas e S um conjunto convexo.

Definimos  $M = \{1, 2, \dots, m\}$  e  $R \subset M$

$$(P^R) \quad \begin{array}{l} \max f(x) \\ \text{s.a. } g_i(x) \geq 0, \quad i \in R \\ x \in S \end{array}$$

- Procedimento da Técnica da Relaxação :

- PASS00 Seja  $\bar{f} = km$ , onde  $km$  é um valor suficientemente grande; escolha  $R$  tal que  $f$  seja limitada superiormente;
- PASS01 Resolver  $(P^R)$ . Se  $(P^R)$  é infactível, então  $(P)$  é infactível; senão, obter  $x^R$  solução de  $(P^R)$ .
- PASS02 Se  $g_i(x^R) \geq 0, i \in M - R$ , então  $x^R$  é solução ótima de  $(P)$ ; caso contrário, vá para PASS03.
- PASS03 Seja  $V \subset M - R$  contendo índices de no mínimo uma restrição violada e seja

$$D = \{ i / g_i(x^R) > 0, i \in R \}.$$

PASSO4 Se  $f(x^R) = \bar{f}$ , faça  $R' = R \cup V$ ,  $R = R'$   
e vá para PASSO1.

PASSO5 Se  $f(x^R) < \bar{f}$ , faça  $R' = R \cup V - D$ ,  $R = R'$   
 $f = f(x^R)$  e vá para PASSO1

Os teoremas abaixo nos garantem que o procedimento acima converge em um número finito de "passos".

TEOREMA 1 : Se  $x^R$  é solução de  $(P^R)$  e as restrições do conjunto  $D$  são "relaxadas", então  $x^R$  é solução de  $(P^{R-D})$ . Assim, se  $x^R$  é solução única de  $(P^R)$ , então  $x^R$  é solução única para  $(P^{R-D})$ .

O teorema acima nos garante o seguinte resultado:

TEOREMA 2 :  $f(x^{R'}) \leq f(x^R)$ ; ou seja, a técnica da Relaxação nos garante decréscimo no valor da função  $f$  e, conseqüentemente, uma aproximação "passo a passo" à solução de  $(P)$ .

TEOREMA 3 : O procedimento da Relaxação termina depois que um número finito de problemas  $(P^R)$  foi resolvido, nos dando informação acerca da solução do problema  $(P)$ .

Dessa forma, os dois últimos "passos" da técnica da Relaxação nos dizem que podemos subtrair de  $(P^R)$  as restrições não ativas em  $x^R$  caso a função  $f$  tenha seu valor diminuído em  $x^R$ ; o "PASSO4" garante que o problema  $(P)$  é resolvido em um número finito de problemas  $(P^R)$ .

As condições de concavidade sobre as funções  $f$  e  $g_i$  e de

convexidade sobre o conjunto  $S$  são fundamentais para a verificação dos resultados acima [38]. No nosso problema, o conjunto  $S$  é definido conforme seção anterior; logo,  $S$  é convexo. As funções  $g_i$  são os hiperplanos tangentes à função dual; como a função dual é linear-por-partes, pode ser representada por um número finito de hiperplanos-suporte.

A cada iteração do método da Aproximação Tangencial o conjunto  $R$  é acrescido de uma restrição violada e temos, dessa forma, garantida a monotonicidade de  $f = z^o$ .

Dessa forma, obedecidos os critérios acima, temos garantido a convergência do método da Aproximação Tangencial num número finito de passos.

No nosso problema, acrescentar uma restrição violada ao conjunto  $R$  significa acrescentar uma restrição ao problema-mestre  $(PM_{LD})^r$ . Logo,  $(PM_{LD})^r$  pode apresentar superdimensionamento no caso do problema (P.L.D.) a ser resolvido ser um problema com muitas variáveis. Portanto, para viabilizarmos computacionalmente o procedimento da Aproximação Tangencial é necessário, numa dada fase do algoritmo, desprezarmos as restrições geradas e que não estão ativas na solução daquela fase.

Dessa forma, no capítulo V apresentamos a técnica da Relaxação como estratégia utilizada ( e implementada ) para o tratamento das restrições não-ativas de  $(PM_{LD})^r$  conjuntamente com um fluxograma do método.

#### IV.3 - PROCEDIMENTO PARA OBTENCAO DA SOLUCAO OTIMA PRIMAL

Devemos, primeiramente, lembrar alguns pontos da teoria da Programação Linear que nos servem de apoio para a resolução dessa fase do algoritmo - a "2ª FASE".



#### IV.3.1 - Preliminares sobre Programação Linear

Seja o problema linear na sua forma primal

$$\begin{array}{ll}
 \text{(P)} & \min \quad c x \quad (1) \\
 & \text{s.a. } A x = b \quad (2) \\
 & \underline{x} \leq x \leq \bar{x} \quad (3)
 \end{array}$$

onde  $A$  é uma matriz de dimensão  $m \times n$ ,  $c$  é um vetor linha  $1 \times n$ ,  $b$  é um vetor coluna  $m \times 1$  e  $x$ ,  $\underline{x}$ ,  $\bar{x}$  vetores-coluna  $n \times 1$ .

Seja  $B$  e  $N$  conjuntos dos índices das colunas básicas e não-básicas, respectivamente, da matriz  $A$ . Através de um arranjo de colunas, podemos escrever  $A = [A^B \quad A^N]$ ; dessa forma, a equação (2) fica

$$A^B x^B + A^N x^N = b \quad (4)$$

$$\text{ou} \quad x^B = (A^B)^{-1} b - (A^B)^{-1} A^N x^N$$

$$\text{Denotamos por } \hat{b} = (A^B)^{-1} b \quad \text{e} \quad \hat{A}^N = (A^B)^{-1} A^N$$

DEFINIÇÃO 1 : Um vetor do tipo  $x = [x^B, x^N]^T$  é uma solução básica de (P) se:

$$(a) \quad x^B = \hat{b} - \hat{A}^N x^N$$

$$(b) \quad x_j = \underline{x}_j \quad \text{ou} \quad x_j = \bar{x}_j, \quad j \in N$$

Além disso, se tivermos  $\underline{x}_j \leq x_j \leq \bar{x}_j$ ,  $j \in B$ , então  $x$  é uma solução básica factível de (P).

DEFINICAO 2 : Dizemos que  $x$  é uma solução básica degenerada de (P) se além da solução básica factível tivermos

$$x_j = (\underline{x}_j) \quad \text{ou} \quad x_j = (\bar{x}_j) \quad \text{para algum } j \in B$$

Seja  $S$  o poliedro formado pelos limites superior e inferior das coordenadas das variáveis  $x$ . Escrevemos a função Lagrangeano associada a (P)

$$L(x, \lambda) = c x + \lambda^T (b - A x) \quad (5)$$

onde  $\lambda^T \in E^n$  é o vetor linha dos multiplicadores de Lagrange.

O problema dual de (P) é

$$(D) \quad \max_{\lambda} \phi(\lambda)$$

$$\text{onde} \quad \phi(\lambda) = \min_{x \in S} L(x, \lambda) \quad (6)$$

$$\text{ou ainda,} \quad \phi(\lambda) = \min_{x \in S} \{ (c^B - \lambda^T A^B) x^B + (c^N - \lambda^T A^N) x^N + \lambda^T b \}$$

Da teoria da programação linear, se  $A^B$  é a matriz básica associada à solução ótima de (P), temos que o multiplicador ótimo é dado por [3,39]  $\lambda^* = c^B (A^B)^{-1}$ . Logo,

$$\phi(\lambda^*) = \min_{x \in S} \{ (c^N - c^B (A^B)^{-1} A^N) x^N \} \quad (7)$$

Dessa forma, temos que o problema (6) de minimizar o Lagrangeano, dado o seu multiplicador ótimo, não determina os valores das variáveis básicas primais na solução ótima; ou seja, no ponto de ótimo de (6),  $x^B$  pode possuir valores arbitrários pertencentes ao intervalo  $[\underline{x}^B, \bar{x}^B]$ .

Portanto, ao resolvermos o problema (6) obtemos simplesmente os seguintes valores para as variáveis não-básicas  $x^N$ :

$$\begin{aligned} x_j &= (\bar{x})_j, & \text{se } (c^N - c^B \hat{A}^N)_j < 0 \\ x_j &= (\underline{x})_j, & \text{se } (c^N - c^B \hat{A}^N)_j > 0 \end{aligned}, \quad j \in N \quad (8)$$

Caso  $(c^N - c^B \hat{A}^N)_j = 0$ , então  $x_j$  assume qualquer valor pertencente ao intervalo  $[(\underline{x})_j, (\bar{x})_j]$ ,  $j \in N$ , o que caracteriza solução ótima múltipla.

Da teoria da dualidade para programação linear, sabemos que ao valor ótimo do multiplicador de Lagrange está associada a solução básica ótima de (P) e que, portanto, deve satisfazer as condições da DEFINICAO 1.

Logo, se temos em mãos o valor ótimo da variável dual,  $\lambda^*$ , resolvermos o problema (6) significa obtermos, através de (8), os valores das variáveis não-básicas da solução ótima de (P).

Resta-nos, portanto, a tarefa de encontrarmos os valores das variáveis básicas da solução ótima do problema primal, as quais devem satisfazer a condição (a) da DEFINICAO 1 e a condição de factibilidade.

A seguir fazemos uma simples adaptação da teoria vista nesta seção para problemas do tipo (P.L.D.).

IV.3.2 - Obtenção das Variáveis Não-Básicas da Solução Ótima de um (P.L.D.)

Sabemos que problemas de Programação Linear Dinâmicos podem ser vistos naturalmente como problemas lineares. Logo, a teoria descrita na seção anterior é aqui estendida para os primeiros.

Vimos na seção II.3 que a função Lagrangeano associada a problemas da forma (P.L.D.) é aditivamente separável no tempo. Logo, o problema de determinarmos os valores das variáveis não-básicas ótimas primais é tratado por estágio de tempo.

Dessa forma, conhecidos os valores ótimos dos multiplicadores  $p^*(t+1)$  e  $\lambda^*(t)$ ,  $t=0, \dots, T-1$ , temos que:

para  $t = 0$ ,

$$L(x(0), u(0), p^*(1), \lambda^*(0)) = [ - p^*(1)A(0) - \lambda^*(0)C(0) ] x(0) + \\ + [ d(0) - p^*(1)B(0) - \lambda^*(0)D(0) ] u(0)$$

Neste caso, sendo  $x(0)$  valores iniciais fixados, interessa-nos saber quais variáveis  $u_1(0), \dots, u_r(0)$  estão em seus limites; ou seja, obtemos  $u_j(0)$  tais que

$$\begin{aligned} \text{se } [d(0) - p^*(1)B(0) - \lambda^*(0)D(0)]_j < 0 \text{ então } u_j(0) = \bar{u}_j(0) \\ \text{se } [d(0) - p^*(1)B(0) - \lambda^*(0)D(0)]_j > 0 \text{ então } u_j(0) = \underline{u}_j(0) \end{aligned} \quad (9)$$

para  $t = 1, \dots, T-1$ ,

$$L(x(t), u(t), p^*(t+1), \lambda^*(t)) = \text{grad}_x(t) x(t) + \text{grad}_u(t) u(t)$$

onde

$$\begin{aligned} \text{grad}_x(t) &= c(t) + p^*(t) - p^*(t+1)A(t) - \lambda^*(t)C(t) \\ \text{grad}_u(t) &= d(t) - p^*(t+1)B(t) - \lambda^*(t)D(t) \end{aligned}$$

Dessa forma, obtemos os valores das variáveis  $x(t)$  e  $u(t)$  tais que

$$\begin{aligned} \text{se } (\text{grad}_x(t))_j < 0 & \quad \text{então } x_l(t) = \bar{x}_l(t) \\ \text{se } (\text{grad}_x(t))_j > 0 & \quad \text{então } x_l(t) = \underline{x}_l(t) \\ \text{se } (\text{grad}_u(t))_j < 0 & \quad \text{então } u_l(t) = \bar{u}_l(t) \\ \text{se } (\text{grad}_u(t))_j > 0 & \quad \text{então } u_l(t) = \underline{u}_l(t) \end{aligned} \quad (10)$$

para  $t = T$ ,

$$L(x(T), u(T), p^*(T), \lambda^*(T)) = [c(T) + p^*(T)] x(T)$$

e temos

$$\begin{aligned} \text{se } (c(T) + p^*(T))_l < 0 & \quad \text{então } x_l(T) = \bar{x}_l(T) \\ \text{se } (c(T) + p^*(T))_l > 0 & \quad \text{então } x_l(T) = \underline{x}_l(T) \end{aligned}$$

Neste caso, conforme DEFINICAO 1 da seção anterior, estamos supondo que todas as variáveis não-básicas da solução ótima de (P) estão num de seus limites. Caso haja multiplicidade de soluções, uma das variáveis com coeficiente nulo deve ter seu valor atribuído em um de seus limites.

#### IV.3.3 - Obtenção da Solução ótima Primal para (P.L.D.)

Após determinarmos as variáveis não-básicas da solução primal relativa ao multiplicador ótimo do (P.L.D.), devemos considerar a matriz de restrições do problema de forma a rearranjarmos suas colunas básicas e não-básicas como apresentado em (4) de IV.3.1.

Ou seja, se  $G$  é a matriz das restrições do (P.L.D.), então podemos obter

$$[ G^B \mid G^N ] \begin{bmatrix} w^B \\ w^N \end{bmatrix} = b \quad (12)$$

onde  $w^B$  e  $w^N$  são vetores que guardam variáveis de estado e de controle básicas e não-básicas, respectivamente, para todos os instantes de tempo e  $b$  vetor de dimensão  $(n+m)T$ .

Sendo  $w^N$  conhecido, escrevemos (12) como

$$G^B w^B = \bar{b} \quad (12.1)$$

onde  $\bar{b} = b - G^N w^N$ ,  $G^B$  matriz quadrada de ordem  $(n+m)T$  e  $G^N$  matriz de ordem  $[(n+m)T \times (r-m)T]$ .

Possuindo colunas linearmente independentes, a matriz  $G^B$  é não-singular; logo, o sistema (12.1) admite solução única.

Para a resolução deste sistema (ou da "2ª FASE" do algoritmo aqui proposto) fazemos uma análise do comportamento das variáveis básicas primais ao longo dos estágios de tempo, análise esta que nos traz vantagens e facilidades computacionais. Veremos na seção seguinte que a distribuição das variáveis básicas nos estágios de tempo pode possibilitar resolvermos o sistema (12.1) de forma recursiva, através de "subsistemas interligados".

#### IV.3.4 - Características do Sistema a ser Resolvido na "2ª FASE"

Um importante resultado relacionado à solução ótima primal de Problemas Lineares Dinâmicos é verificado antes de descrevermos o sistema (12.1).

Para isso, retomamos o problema

$$\begin{aligned}
 & \min \sum_{t=0}^{T-1} c(t+1)x(t+1) + d(t)u(t) \\
 & \text{s.a. } x(t+1) = A(t)x(t) + B(t)u(t) \\
 & \quad f(t) = C(t)x(t) + D(t)u(t) \\
 & \quad x(0) \quad \text{dado} \\
 & \quad u(t), x(t+1) \in S, \quad t=0, \dots, T-1
 \end{aligned}$$

(P.L.D.)

Seja  $[u^*(t)]_{t=0}^{T-1}$  a política ótima admissível considerando  $x(0)$  como estado inicial e seja  $[x^*(t+1)]_{t=0}^{T-1}$  a trajetória ótima gerada por esta política ótima.

Definimos o problema linear dinâmico extraído do (P.L.D.) a partir do estágio  $(T-k)$  na forma

$$\begin{aligned}
 & \min \sum_{t=T-k}^{T-1} c(t+1)x(t+1) + d(t)u(t) \\
 & \text{s.a. } x(t+1) = A(t)x(t) + B(t)u(t) \\
 & \quad f(t) = C(t)x(t) + D(t)u(t) \\
 & \quad x(T-k) = x^*(T-k) \\
 & \quad u(t), x(t+1) \in S, \quad t=T-k, \dots, T-1
 \end{aligned}$$

(P<sub>T-k</sub>)

Devemos notar que o estado inicial de (P<sub>T-k</sub>) pertence à

trajetória ótima do (P.L.D.).

Dessa forma, enunciaremos o seguinte resultado:

TEOREMA: Supondo  $(P_{T-k})$ ,  $k = 1, \dots, T$ , não-degenerado, a solução do (P.L.D.) apresenta variáveis básicas distribuídas igualmente por estágio de tempo.

Prova: este resultado é uma consequência direta do Princípio de Otimalidade de Bellman, como vemos a seguir.

(C1) - Ao tomarmos o problema  $(P_{T-k})$  sabemos, pelo Princípio de Otimalidade, que sua política ótima admissível é  $[u^*(t)]_{t=T-k}^{T-1}$  resultando na trajetória ótima  $[x^*(t+1)]_{t=T-k}^{T-1}$ . Por hipótese,  $(P_{T-k})$  é não degenerado e, conseqüentemente, como um problema linear, deve apresentar em sua solução ótima  $(n+m)k$  variáveis básicas cujos valores não estão em seus limites.

(C2) - Por outro lado, ao tomarmos o problema  $(P_{T-k+1})$  sabemos também que sua política ótima admissível é a política  $[u^*(t)]_{t=T-k+1}^{T-1}$  e trajetória ótima  $[x^*(t+1)]_{t=T-k+1}^{T-1}$ . Por suposição,  $(P_{T-k+1})$  também é não-degenerado e, conseqüentemente, sua solução ótima apresenta  $(n+m)(k-1)$  variáveis cujos valores não estão em seus limites.

Logo, como  $(n+m)k = (n+m) + (n+m)(k-1)$ , podemos concluir de (C1) e (C2) que a solução do problema  $(P_{T-k})$  possui  $(n+m)$  variáveis básicas no estágio  $(T-k)$  e  $(n+m)(k-1)$  variáveis básicas distribuídas nos estágios  $(T-k+1), \dots, (T-1)$ .

Como esta análise é válida para quaisquer  $(P_{T-k})$ ,  $k=1, \dots, T$  concluímos que a propriedade acima é verdadeira.



O resultado acima nos chama atenção para características importantes encontradas na matriz do sistema (12.1), as quais nos possibilitam resolvê-lo através de "subsistemas sequenciais", como veremos a seguir.

Dado que as variáveis não-básicas da solução ótima primal são conhecidas (conforme visto na seção IV.3.2) e dado que o número de variáveis básicas é igual ao número de equações por estágio de tempo, podemos escrever os "subsistemas" conforme definimos abaixo.

Consideramos a equação (4) do (P.L.D.). Sejam  $B^B$  e  $B^N$  conjunto das colunas básicas e não-básicas, respectivamente, da matriz  $B$ ;  $A^B$  e  $A^N$  conjunto das colunas básicas e não-básicas, respectivamente, da matriz  $A$  e  $I^B$  e  $I^N$  conjunto das colunas básicas e não-básicas da matriz identidade e sejam  $u_t^B$ ,  $x_t^B$ ,  $u_t^N$  e  $x_t^N$  variáveis de controle e de estado básicas e não-básicas, respectivamente, todos no instante de tempo "t".

para  $t = 0$ ,

$$[B^B \mid B^N] \begin{bmatrix} u_0^B \\ 0 \\ u_0^N \end{bmatrix} - [I^B \mid I^N] \begin{bmatrix} x_1^B \\ x_1^N \end{bmatrix} = b_1$$

ou  $[B^B \mid -I^B] \begin{bmatrix} u_0^B \\ 0 \\ x_1^B \end{bmatrix} = \bar{b}_1$  (S<sub>1</sub>)

onde  $\bar{b}_1 = b_1 - B^N u_0^N + I^N x_1^N$ ,

e  $[B^B \mid -I^B]$  é uma matriz de ordem n.

Dessa forma, dados os valores de  $u_0^N$  e  $x_1^N$  determinamos, de maneira única, os valores de  $u_0^B$  e  $x_1^B$  procurados.

para  $t = 1$ ,

$$[A^B; A^N] \begin{bmatrix} x_1^B \\ x_2^N \\ x_1^N \end{bmatrix} + [B^B; B^N] \begin{bmatrix} u_1^B \\ u_2^N \end{bmatrix} - [I^B; I^N] \begin{bmatrix} x_2^B \\ x_2^N \end{bmatrix} = b_2$$

Como temos conhecidas  $x_1^B$  através de  $(S_1)$  e conhecidas  $x_1^N$ ,  $u_1^N$  e  $x_2^N$ , obtemos o "subsistema"

$$[B^B; -I^B] \begin{bmatrix} u_1^B \\ x_2^B \end{bmatrix} = \bar{b}_2 \quad (S_2)$$

onde

$$\bar{b}_2 = b_2 - [A^B; A^N] \begin{bmatrix} x_1^B \\ x_2^N \\ x_1^N \end{bmatrix} - B^N u_1^N + I^N x_2^N$$

e  $[B^B; -I^B]$  é uma matriz de ordem  $n$ .

Assim, sucessivamente, ao resolvermos o "subsistema"  $(S_t)$  temos a informação fornecida por  $(S_{t-1})$  sobre a variável  $x_{t-1}^B$  e então podemos escrever  $(S_t)$  na forma

$$[B^B; -I^B] \begin{bmatrix} u_{t-1}^B \\ x_t^B \end{bmatrix} = \bar{b}_t \quad (S_t)$$

onde

$$\bar{b}_t = b_t - [A^B; A^N] \begin{bmatrix} x_{t-1}^B \\ x_{t-1}^N \end{bmatrix} - B^N u_{t-1}^N + I^N x_t^N$$

$[B^B; -I^B]$  é uma matriz de ordem  $n$  e  $b_t$ ,  $1 \leq t \leq (n+m)$ , vetor correspondente às coordenadas de "b", do sistema (12), na



$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} u_1^{(1)} \\ u_3^{(1)} \end{bmatrix} = b_1 - \begin{bmatrix} 1 & 2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} u_2^{(1)} - \\ - \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \end{bmatrix} \quad (S_1)$$

Novamente, através de  $(S_1)$  obtemos os valores das variáveis básicas  $u_1^{(1)}$  e  $u_3^{(1)}$  para, a seguir, resolvermos  $(S_2)$ .

$$\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1^{(2)} \\ u_2^{(2)} \end{bmatrix} = b_2 - \begin{bmatrix} 1 & 2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \end{bmatrix} - \begin{bmatrix} 0 \\ 2 \end{bmatrix} u_3^{(2)} - \\ - \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1^{(3)} \\ x_2^{(3)} \end{bmatrix} \quad (S_2)$$

Nota: Neste exemplo notamos que a resolução de  $(S_2)$  independe da resolução de  $(S_1)$  visto que em  $T=1$  as variáveis de estado não aparecem como variáveis básicas.

Dessa forma, resolver a "2ª FASE" do algoritmo proposto neste capítulo, supondo não-degenerescência para  $(P_{T-k})$ , significa resolvermos o sistema (12.1) através de  $T$ -"subsistemas"  $(S_t)$  recursivamente.

Notamos que se algum  $(P_{T-k})$ ,  $k=1, \dots, T$ , é degenerado, então a solução do (P.L.D.) pode apresentar variáveis básicas distribuídas por períodos de tempo e em número não uniforme. Neste caso, algum  $(P_{T-k+j})(j > 0)$  deve ser não-degenerado.

EXEMPLO : - Tomamos um (P.L.D.) e supomos o problema  $(P_{T-1})$  degenerado ; então, pelo menos uma das variáveis básicas de sua solução ótima deve estar num dos limites. Sendo  $(P_{T-2})$  distinto de  $(P_{T-1})$ , pode ser ou não-degenerado ;

1<sup>o</sup>) Se  $(P_{T-2})$  é não-degenerado, então sua solução ótima apresenta  $2(n+m)$  variáveis básicas "fora" de seus limites; como  $m_1 < (n+m)$  delas estão dentre  $u(T-1)$  e  $x(T)$ , então  $m_2 \geq (n+m+1)$  devem estar dentre as variáveis de  $u(T-2)$  e  $x(T-1)$  de tal forma que  $(m_1 + m_2) = 2(n+m)$ . Assim, determinamos o período entre  $t=T-2$  e  $t=T$  no qual existe um número de variáveis básicas "fora de seus limites" igual ao número de equações da matriz do (P.L.D.) correspondendo ao mesmo período.

2<sup>o</sup>) Se  $(P_{T-2})$  é degenerado, sua solução ótima deve apresentar  $m_1 < 2(n+m)$  variáveis básicas com seus valores no interior do conjunto S.

Nesta linha de raciocínio, se  $(P_{T-3})$  é não-degenerado então sua solução ótima pode apresentar:

3<sup>o</sup>)  $(n+m)$  variáveis básicas entre  $u(T-3)$  e  $x(T-2)$  caso  $(P_{T-2})$  for não-degenerado ;

4<sup>o</sup>)  $3(n+m)$  variáveis básicas distribuídas entre  $t=T-3$  e  $t=T$ , caso  $(P_{T-2})$  for degenerado.

Seja a matriz do exemplo anterior. A análise acima nos induz a pensarmos nas colunas básicas da solução ótima do (P.L.D.) distribuídas segundo a figura abaixo.





análise encontramos facilidade computacional para resolvermos o sistema (12.1) que nos fornece os valores das variáveis básicas da solução ótima primal.

Logo, dada a ocorrência de degenerescência em  $(P_{T-k})$ , para algum  $k$ , podemos obter a decomposição do sistema (12.1) em "subsistemas" representados na seguinte forma:

Definimos  $k, \delta \in Z^+, 0 \leq \delta, k \leq T-1$  e  $R_k^{k+\delta}$  o período de tempo que compreende os estágios  $k, k+1, \dots, k+\delta$  e que contém  $(n+m)(\delta+1)$  variáveis básicas fora de seus limites.

a) Se  $k = 0$  e  $\delta = T$ , então devemos resolver o sistema (12.1) tendo como matriz  $G^B$ , matriz das colunas básicas da matriz das restrições do (P.L.D.).

b) Se  $k \neq 0$  e/ou  $\delta \neq T$ , então o sistema (12.1) deve ser resolvido através de no mínimo dois "subsistemas", que podem ou não ser iguais a  $(S_k)$ , de forma recursiva.

Seja  $(S_k^{k+\delta})$  o "subsistema" relacionado ao período  $R_k^{k+\delta}$ . Consideramos  $A_k^B, A_k^N, B_k^B, B_k^N, I_k^B$  e  $I_k^N$  matrizes das colunas básicas e não-básicas de  $A, B$  e  $I$ , respectivamente, no instante de tempo " $k$ ". Para a representação de  $(S_k^{k+\delta})$  tomamos novamente apenas a equação (1) do (P.L.D.) com o propósito de facilitar a notação e lembramos que o mesmo resultado é válido considerando também a equação (2) do mesmo problema.



$$\begin{bmatrix} A_k^B & B_k^B & -I_{k+1}^B & & & & \\ & & A_{k+1}^B & B_{k+1}^B & -I_{k+2}^B & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & A_{k+\delta}^B & B_{k+\delta}^B & -I_{k+\delta+1}^B \end{bmatrix} \begin{bmatrix} U_k^B \\ X_{k+1}^B \\ U_{k+1}^B \\ X_{k+2}^B \\ \vdots \\ U_{k+\delta}^B \\ X_{k+\delta+1}^B \end{bmatrix} = \begin{bmatrix} C_k \\ C_{k+1} \\ C_{k+2} \\ \vdots \\ C_{k+\delta} \\ C_{k+\delta+1} \end{bmatrix}$$

onde temos conhecidos os valores de  $x_k^B$ ,  $x_k^N$ ,  $u_k^N$ ,  $x_{k+1}^N$ , ...,  $u_{k+\delta}^N$  e  $x_{k+\delta+1}^N$  e

$$C_k = b_k - [A_k^B; A_k^N] \begin{bmatrix} x_k^B \\ - \\ x_k^N \end{bmatrix} - B_k^N u_k^N + I_{k+1}^N x_{k+1}^N, \quad k \neq 0$$

$$C_k = b_k - B_k^N u_k^N + I_{k+1}^N x_{k+1}^N, \quad k = 0$$

$$C_{k+i} = b_{k+i} - A_{k+i}^N x_{k+i}^N - B_{k+i}^N u_{k+i}^N + I_{k+i+1}^N x_{k+i+1}^N$$

$$i = 1, \dots, \delta+1.$$

**Caso Particular:** Se no estágio  $k + \delta$  não existirem variáveis de estado básicas, então poderão aparecer "subsistemas" independentes; isto é, a matriz do sistema (12.1) pode apresentar-se bloco diagonal.

#### IV.3.5 - Procedimento Iterativo

Definimos:

$N_t^{t+\delta}$  = número de variáveis básicas no período  $(t, t+\delta)$ ,  $\delta \geq 0$ ,  
 $\delta \in \mathbb{Z}$ .

$M_t^{t+\delta}$  = número de equações do sistema no período  $(t, t+\delta)$ .

Considerando a análise feita nas seções anteriores, o procedimento iterativo adotado para a resolução da "2ª FASE" do algoritmo proposto neste capítulo é dado por:

- PASS01 : Seja  $(p^*, \lambda^*)$  solução ótima dual do (P.L.D.);  
faça  $t = \delta = 0$ .
- PASS02 : Minimize  $L(x, u, p^*, \lambda^*)$  determinando os valores das  
 $x, u \in S$   
variáveis não-básicas primais no período  $(t, t+\delta)$ ;
- PASS03 : Se  $N_t^{t+\delta} = M_t^{t+\delta}$ , vá para PASS04;  
caso contrário, faça  $\delta = \delta + 1$  e vá para PASS02.
- PASS04 : Resolver o "subsistema"  $(S_t^{t+\delta})$  (seção anterior);  
faça  $t = t + \delta$  e vá para PASS05.
- PASS05 : Se  $t = T$ , PARE.  
Caso contrário, faça  $\delta = 0$  e vá para PASS02.

Através da análise feita nesta seção, podemos perceber que a resolução desta fase do algoritmo é bastante simples e, dadas as características apresentadas pelo sistema, requer pequenos gastos operacionais, podendo ser aplicada principalmente a problemas com horizonte grande, como verificamos no capítulo seguinte.

## CAPITULO V

### IMPLEMENTAÇÃO DA DECOMPOSIÇÃO VIA LAGRANGEANO SIMPLES - TESTES E CONCLUSÕES

#### V.1 - INTRODUÇÃO

Nesta técnica para resolução de um (P.L.D.) via dualidade, destacamos o algoritmo em duas fases: a "1ª FASE", que é a fase de obtenção da solução ótima dual via aproximação tangencial a função dual e a "2ª FASE" que utiliza a solução ótima dual para obtenção da solução ótima primal através da resolução do Lagrangeano ordinário do problema e de uma sequência de sistemas lineares, conforme vimos no capítulo anterior.

Como o próprio título sugere, neste capítulo descrevemos a maneira com que implementamos estas duas FASES. Na "1ª FASE", a cada hiperplano gerado pelo método da Linearização Externa, um problema-mestre "relaxado" deve ser resolvido, problema que difere do anterior em apenas uma restrição, a qual foi acrescida. Para resolvermos (PM)<sub>r</sub>, programamos o método Dual-Simplex para problemas cujas variáveis são "canalizadas" [44], o qual descrevemos em V.2; neste programa, a matriz inversa é atualizada através de uma análise de pós-otimização [56], onde a inversa da matriz-base obtida para o problema anterior é reutilizada. Ainda nesta seção descrevemos a técnica utilizada para "relaxarmos" as restrições que não estão ativas na solução de um determinado (PM<sub>LD</sub>)<sup>r</sup>, apresentando fluxogramas desta FASE.

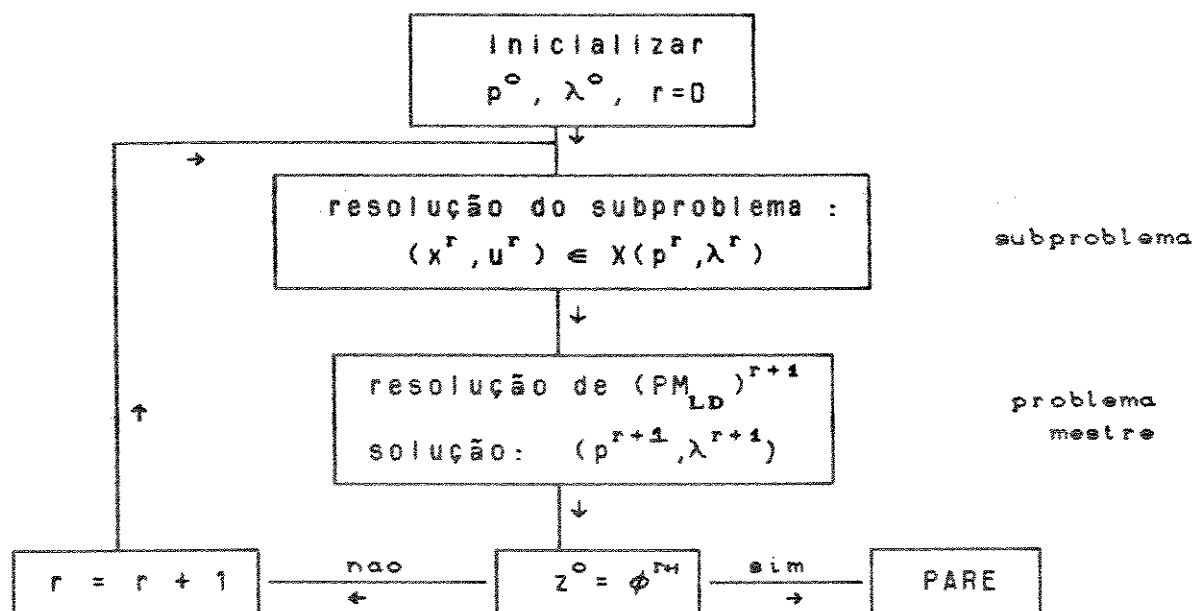
A implementação da "2ª FASE" é descrita na seção V.3, onde é apresentada estimativa da ocorrência dos "subsistemas sequenciais" na resolução desta FASE. Em V.4 apresentamos os resultados

computacionais conseguidos e as conclusões.

## V.2 - IMPLEMENTAÇÃO DA "1ª FASE"

Considerando a teoria descrita na seção IV.2, é fácil vermos que o custo computacional da "1ª FASE" do método da Linearização Externa está praticamente centrado na resolução do problema-mestre  $(PM_{LD})^r$ , dado que a obtenção da solução para o subproblema é extremamente simples e rápida.

Relembrando ser  $z^o$  o valor máximo da linearização da função dual e  $\phi^r$  o valor real da função dual no ponto correspondente da  $r$ -ésima iteração do método empregado nesta FASE, numa análise geral desta fase apresentamos seu fluxograma na forma abaixo.



( figura 5.1 )

A cada iteração do método da Aproximação Tangencial temos que resolver um problema  $(PM_{LD})^r$ , que difere de  $(PM_{LD})^{r-1}$  em apenas uma restrição. Dessa forma, conhecida a solução de  $(PM_{LD})^{r-1}$  é fácil inicializarmos o procedimento Dual-Simplex para a resolução de  $(PM_{LD})^r$ . Escolhemos o seguinte algoritmo para implementação desse "passo" :

#### V.2.1 - Algoritmo Dual-Simplex Canalizado

Como já sabemos da teoria da Programação Linear, este algoritmo trabalha com soluções básicas primais ineficazes no sentido de maximizar a função dual do problema. Antes de descrevê-lo, relembramos alguns aspectos importantes da teoria da programação linear necessários para desenvolvê-lo.

Seja o problema linear

$$\begin{aligned} (P) \quad & \min \quad c x & (1) \\ & \text{s.a.} \quad A x = b & (2) \\ & \quad \underline{x} \leq x \leq \bar{x} & (3) \end{aligned}$$

onde  $A$  é uma matriz de ordem  $(n \times m)$ ,  $c$  é um vetor-linha  $(1 \times n)$ ,  $b$  um vetor  $(m \times 1)$  e  $\underline{x}, \bar{x}, x$  vetores  $(n \times 1)$ .

Consideramos a teoria descrita em IV.3.

Sejam

$$\begin{aligned} z^N &= c^N (A^B)^{-1} A^N \\ z^B &= c^B (A^B)^{-1} A^B \end{aligned}$$

Podemos escrever, por (1) e (2),

$$\begin{aligned}
 cx &= c^B x^B + c^N x^N = c^B (A^B)^{-1} b - (z^N - c^N) x^N = \\
 &= c^B (A^B)^{-1} b - \sum_{j \in R_1} (z_j^N - c_j^N) x_j^N - \sum_{j \in R_2} (z_j^N - c_j^N) x_j^N
 \end{aligned}$$

onde  $R_1 = \{ j / x_j^N = \bar{x}_j \}$  e  $R_2 = \{ j / x_j^N = \underline{x}_j \}$

Se  $x$  é solução factível de (P) e se

$$(z_j^N - c_j^N) > 0, \quad j \in R_1 \quad (4)$$

$$(z_j^N - c_j^N) < 0, \quad j \in R_2 \quad (5)$$

então  $x$  é solução ótima,  $x^*$ , para (P).

O problema dual relacionado a (P) e equivalente a (D), seção IV.3.1, é

$$\begin{aligned}
 \text{(PD)} \quad & \text{Max} \quad \lambda^T b - \tau^T \bar{x} + \gamma^T \underline{x} \\
 & \text{s.a.} \quad \lambda^T A - \tau^T + \gamma^T \leq c \\
 & \quad \tau^T, \gamma^T \geq 0
 \end{aligned}$$

onde  $\lambda^T$ ,  $\tau^T$  e  $\gamma^T$  são os vetores multiplicadores de Lagrange associados às restrições (2),  $x \leq \bar{x}$  e  $x \geq \underline{x}$  no espaço vetorial  $E^n$ , respectivamente.

**PROPOSIÇÃO :** Seja  $x^*$  solução ótima de (P) cuja matriz-base associada é  $A^B$ . Seja  $\lambda^* = c^B (A^B)^{-1}$  e sejam

$$\tau_j^* = \begin{cases} (\lambda^*)^T A_j - c_j & , \text{ se } x_j^* = \bar{x}_j \\ 0 & \text{ caso contrário} \end{cases}$$

$$\gamma_j^* = \begin{cases} -(\lambda^*)^T A_j + c_j & , \text{ se } x_j^* = \underline{x}_j \\ 0 & \text{ caso contrário} \end{cases}$$

onde  $A_j$  é a  $j$ -ésima coluna da matriz  $A$ . Então  $(\lambda^*, \tau^*, \gamma^*)$  soluciona (PD).

O resultado acima [34] possibilita relacionarmos a condição de otimalidade dada pelo Simplex-Canalizado e as condições de factibilidade para os multiplicadores  $\lambda^T$ ,  $\tau^T$  e  $\gamma^T$  em (PD), da seguinte forma:

Seja  $x^*$  solução não-factível para (P) tal que (4) e (5) estão satisfeitas. Isto significa que a terna  $(\lambda^T, \tau^T, \gamma^T)$ , com  $\lambda^T$ ,  $\tau^T$  e  $\gamma^T$  definidos segundo a proposição acima, é solução factível para (PD) (neste caso,  $A^B$  é a matriz-base associada à solução  $x^*$ ).

Ao buscarmos a solução de (PD) estamos, equivalentemente, buscando o valor máximo da função dual associada a (P).

Definindo  $\lambda^T = c^B (A^B)^{-1}$ , podemos escrever

$$\begin{aligned} z^B &= \lambda^T A^B \\ z^N &= \lambda^T A^N = c^B \hat{A}^N, \text{ onde } \hat{A}^N = (A^B)^{-1} A^N \end{aligned} \quad (B)$$

Assim, a função dual, seção IV.3.1, pode ser descrita como

$$\phi(\lambda) = \lambda^T b + \min_{x^B \leq x^B \leq \bar{x}^B} (c^B - z^B) x^B + \min_{\underline{x}^N \leq x^N \leq \bar{x}^N} (c^N - z^N) x^N$$

onde temos  $(c^B - z^B) = 0$ .

O algoritmo Dual-Simplex trabalha no sentido de maximizar o valor da função dual " $\phi$ " preservando, a cada passo, a factibilidade dos multiplicadores até atingir a factibilidade primal para a variável  $x$ .

Portanto, partindo de uma solução dual-factível, devemos atualizar o valor do multiplicador

$$\lambda'^T = \lambda^T + \delta (A^B)^{-1} \quad (7)$$

onde  $\delta \in E^m$  deve ser tal que  $\phi(\lambda') > \phi(\lambda)$ .

Utilizando (7), obtemos substituindo em (6),

$$\begin{aligned} (z')^B &= \lambda'^T A^B + \delta = z^B + \delta \\ (z')^N &= \lambda'^T A^N + \delta (A^B)^{-1} A^N = z^N + \delta \hat{A}^N \end{aligned} \quad (8)$$

Dessa forma, o valor da função dual fica

$$\begin{aligned} \phi(\lambda') &= \lambda'^T b + \delta (A^B)^{-1} b + \min \{ (c^B - z^B - \delta) x^B + (c^N - z^N - \delta \hat{A}^N) x^N \} \\ \text{s. a.} \quad &\underline{x}^B \leq x^B \leq \bar{x}^B \\ &\underline{x}^N \leq x^N \leq \bar{x}^N \end{aligned}$$



ou ainda

$$\phi(\lambda') = \lambda^T b + \delta \hat{b} + \min_{\underline{x}^B \leq x^B \leq \bar{x}^B} \{-\delta x^B\} + \min_{\underline{x}^N \leq x^N \leq \bar{x}^N} \{(c^N - z^N - \delta \hat{A}^N)x^N\} \quad (9)$$

onde  $\hat{b} = (A^B)^{-1} b$  e temos que a solução do problema  $\min_{\underline{x}^B \leq x^B \leq \bar{x}^B} -\delta x^B$

é dada por

$$x_j^B = \begin{cases} \bar{x}_j^B, & \text{se } \delta > 0 \\ \underline{x}_j^B, & \text{se } \delta < 0 \\ x_j^B \in [\underline{x}_j^B, \bar{x}_j^B] & \text{se } \delta = 0 \end{cases} \quad (10)$$

e o problema  $\min_{\underline{x}^N \leq x^N \leq \bar{x}^N} (c^N - z^N - \delta \hat{A}^N)x^N$  tem como solução

$$x_j^N = \begin{cases} \bar{x}_j^N, & \text{se } (\hat{c}')_j^N < 0 \\ \underline{x}_j^N, & \text{se } (\hat{c}')_j^N > 0 \end{cases} \quad (11)$$

onde  $(\hat{c}')^N = c^N - z^N - \delta \hat{A}^N$ .

Denotando por  $(x^N)^*$  e  $(x^B)^*$  as soluções de (11) e (10), respectivamente, podemos escrever

$$\begin{aligned} \phi(\lambda') &= \lambda^T b + (c^N - z^N)(x^N)^* + \delta (b - A^N(x^N)^* - (x^B)^*) \\ &= \phi(\lambda) + \delta (x'^B - (x^B)^*) \end{aligned} \quad (12)$$

$$\text{onde } x'^B = \hat{b} - \hat{A}^N(x^N)^*$$

Da equação (12) concluímos que para aumentarmos o valor da função dual, devemos observar que:

(a) Se  $x_j^B > (\bar{x}^B)_j$ , então devemos fazer  $\delta > 0$  :

(b) Se  $x_j^B < (\underline{x}^B)_j$ , então devemos fazer  $\delta < 0$  .

Sabemos que, havendo infactibilidade para (P), devemos calcular "δ" de tal modo que a atualização (7) para a variável λ maximize a função dual. De (a) e (b) temos que, para isso, δ deve ser positivo caso haja alguma variável primal assumindo valor maior que seu limite superior e δ deve ser negativo caso haja alguma variável primal assumindo valor menor que seu limite inferior.

Dessa forma, para termos o valor exato de δ, supomos que φ(λ) não está no seu ponto de máximo; logo, existe uma variável básica primal  $x_l$ , "l" ocupando a posição "r" em  $A^B$  e tal que  $x_l \in [\underline{x}_l, \bar{x}_l]$ . Portanto, temos que dar "um passo" no sentido de maximizar a função dual.

#### - Procedimento iterativo para Dual-Simplex Canalizado

Tomamos  $\delta = t e_r = t (0, \dots, 0, 1, 0, \dots, 0)$  onde "1" é a r-ésima coordenada de  $e_r$  : temos por (8) as seguintes modificações:



2º CASO ) Se  $x_l < \underline{x}_l$  devemos ter  $t < 0$  (por (b));

assim,  $t = \max \{ t_3, t_4 \}$ , onde

$$t_3 = \max_{j \in N} \frac{\hat{c}_j}{(\hat{A}_r^B)_j}, \quad \text{com } \hat{c}_j > 0 \text{ e } (\hat{A}_r^B)_j < 0$$

$$t_4 = \max_{j \in N} \frac{\hat{c}_j}{(\hat{A}_r^B)_j}, \quad \text{com } \hat{c}_j < 0 \text{ e } (\hat{A}_r^B)_j > 0 \quad (15)$$

Dessa forma, ao calcularmos o valor de "t", alteramos os custos relativos das variáveis não-básicas e, em particular, obtemos um deles anulado.

Chamamos por  $x_s$  a variável cujo custo relativo  $\hat{c}'_s$  foi anulado; logo,  $x_s$  deve substituir a variável  $x_l$  na base, a qual passa a ter custo relativo  $\hat{c}'_l = -t$ .

Feita a troca de variáveis na base, devemos atualizar os valores daquelas que ali permaneceram.

Tendo em vista que os "sinais" dos custos relativos das variáveis não-básicas permanecem inalterados, então temos que atualizar somente o valor da variável  $x_l$  e, por (11),

$$x_l = \begin{cases} \bar{x}_l & , \text{ se } t > 0 \\ \underline{x}_l & , \text{ se } t < 0 \end{cases} \quad (16)$$

As variações dos valores de  $x_l$  e  $x_s$ ,  $\Delta x_l$  e  $\Delta x_s$

respectivamente, acarretam variações nos valores das variáveis básicas.

Sabemos que

$$\Delta x_l = x_l^* - x_l$$

onde  $x_l^*$  é o valor assumido conforme (16). Ao escrevermos

$$x^B = \hat{b} - \hat{A}^{N-\langle s \rangle} x^{N-\langle s \rangle} - \hat{A}^s (x_s - \Delta x_s) = x^B - \Delta x^B$$

obtemos 
$$\Delta x^B = -\hat{A}^s \Delta x_s$$

Logo, 
$$\Delta x_l = -(\hat{A}_r^s) \Delta x_s$$

Das igualdades obtidas acima podemos concluir que

$$\Delta x_s = \frac{x_l^* - x_l}{\hat{A}_r^s} \quad (17)$$

onde  $\hat{A}_r^s$  é o elemento da  $r$ -ésima linha e  $s$ -ésima coluna do "tableau" atualizado.

Assim, se  $\Delta x_s$  é a variação nos valores das variáveis básicas que permaneceram na base, então

$$x^B = x^B - \Delta x^B = x^B + \hat{A}^s \Delta x_s \quad (18)$$

Logo, por (17) e (18) temos que

$$\begin{aligned} x_s &= x_s - \frac{x_l^* - x_l}{\hat{A}_r^s} \\ x_j &= x_j + \hat{A}_j^s \frac{x_l^* - x_l}{\hat{A}_r^s}, \quad j \in B, j \neq l \end{aligned} \quad (19)$$

Dessa forma, podemos enunciar o seguinte procedimento iterativo :

- Algoritmo Dual-Simplex Canalizado -

PASS01 : Seja  $x$  solução básica dual factível:

$$x = (x^B, x^N) \text{ conforme (11) e (12)}$$

se  $\underline{x}^B \leq x^B \leq \overline{x}^B$ , PARE ; caso contrário,

se  $\exists j / x_j^B > \overline{x}_j^B$ , vá para PASS03 ;

se  $\exists j / x_j^B < \underline{x}_j^B$ , vá para PASS02.

PASS02 : Calcular  $t = \max \{ t_3, t_4 \}$  conforme (15);  
vá para PASS04.

PASS03 : Calcular  $t = \min \{ t_1, t_2 \}$  conforme (14);  
vá para PASS04.

PASS04 : Atualização do TABLEAU:

Seja  $k$  o índice da variável a entrar na base:

$$\hat{c}'_j = \hat{c}_j - t \hat{A}_i^j \quad ; \quad j \in N$$

$$\hat{c}'_k = 0$$

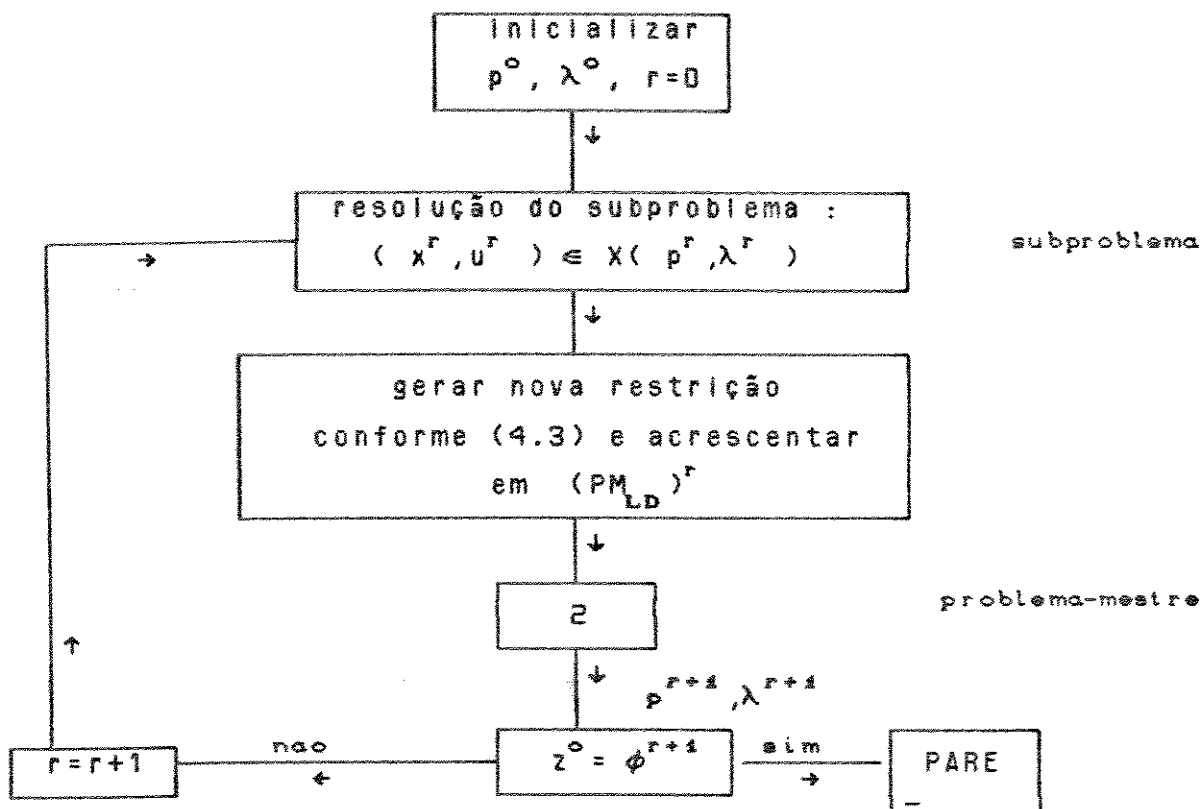
$$\hat{c}'_i = -t$$

$$x_i = x_i^* \text{ conforme (16)}$$

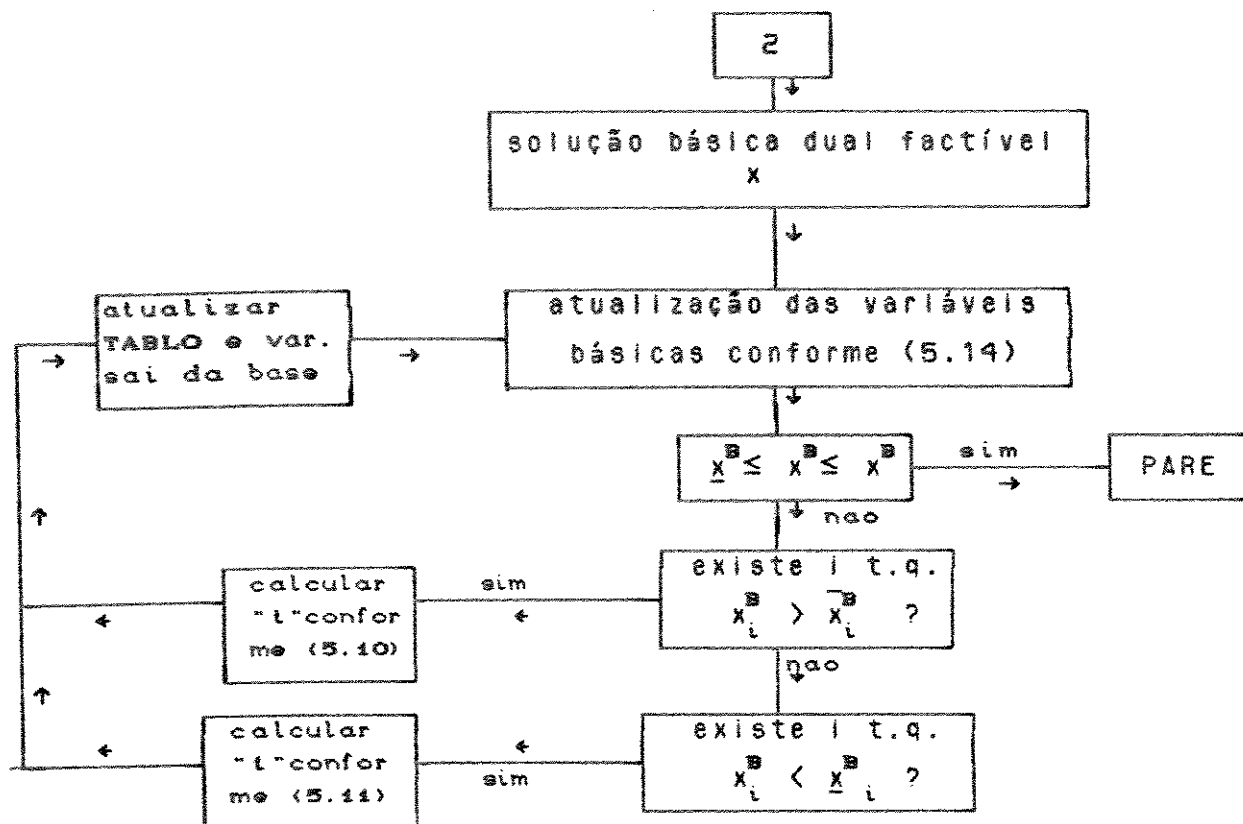
Trocar a coluna  $i$  pela coluna  $k$  na matriz  $A^B$ ;

vá para PASS01 .

Podemos, a partir do algoritmo descrito acima, reescrever o fluxograma apresentado na figura 5.1 da seguinte maneira:



( figura 5.2 )



( figura 5.2 )

Realmente, através do fluxograma acima, percebemos que o esforço computacional da "1ª FASE" do algoritmo da Aproximação Tangencial está concentrado na resolução do problema-mestre  $(PM_{LD})^r$  que, à medida que novas restrições são geradas, pode apresentar matriz de dimensão muito grande, causando, dessa forma, excessivo número de operações quando faz-se necessário "pivotar" os elementos de sua matriz.

Portanto, é imprescindível relaxarmos as restrições não-ativas do problema-mestre para reduzirmos os excessos computacionais.



Outro fato importante para a redução dos gastos computacionais na resolução do problema-mestre é a utilização da teoria de pós-otimização na atualização da inversa da matriz-base necessária a cada iteração do algoritmo Dual-Simplex Canalizado.

Como podemos perceber através da figura 5.2, a cada iteração do algoritmo associado à técnica da Aproximação Tangencial obtemos a matriz das restrições de  $(PM_{LD})^r$  acrescida de uma linha e, conseqüentemente, de uma coluna (ver seção IV.2). Assim, adaptamos nosso problema perfeitamente à seguinte situação:

- ATUALIZAÇÃO DA MATRIZ INVERSA DA BASE

Seja  $B$  a matriz básica na solução ótima de  $(PM_{LD})^r$ .

Ao adicionarmos uma restrição a  $(PM_{LD})^r$ , obtemos  $(PM_{LD})^{r+1}$ ; se a solução ótima de  $(PM_{LD})^r$  não satisfaz a nova restrição, obtemos, para  $(PM_{LD})^{r+1}$ , uma matriz base  $B'$  inicial que é dada por

$$B' = \left[ \begin{array}{c|ccc} -B & & & 0 \\ \alpha_r & & & 1 \end{array} \right] \quad (20)$$

sendo "1" coeficiente da variável de folga associada à nova restrição e  $\alpha_r$  vetor dos coeficientes da nova restrição relacionados aos índices das variáveis básicas de  $(PM_{LD})^r$  na solução ótima.

Dessa forma, a inversa  $(B')^{-1}$  é conhecida e é dada por [56]

$$(B')^{-1} = \left[ \begin{array}{c|c} B^{-1} & 0 \\ \hline -\alpha_r B^{-1} & 1 \end{array} \right] \quad (21)$$

#### V.2.2 - Relaxação em $(PM_{LD})^r$

Como vimos na seção IV.2.2,  $(PM_{LD})^r$  tende a tornar-se um problema com um número grande de restrições e, dessa forma, é fundamental adotarmos critérios que venham facilitar a sua resolução. Uma maneira natural de contornarmos o superdimensionamento do problema é eliminarmos as restrições de linearização que foram geradas e que não estão ativas na solução de determinada fase do processo de resolução.

No nosso problema, determinar quais são as restrições não-ativas numa dada solução de  $(PM_{LD})^r$  é simplesmente reconhecer quais variáveis-de-folga das restrições geradas estão na base.

Conforme a técnica descrita em IV.2.4, as restrições relaxadas podem ser reutilizadas caso elas estejam "violadas" numa determinada solução do problema. No nosso problema, toda restrição gerada pelo algoritmo de Aproximação Tangencial é uma restrição violada.

Dessa forma, podemos enunciar o algoritmo da seção IV.2.4 da seguinte maneira:

Definimos

- $k$  = parâmetro que determina a fase em que relaxamos as restrições não-ativas do problema-mestre.
- $M$  = conjunto de todas as restrições do problema;
- $R$  = conjunto das restrições geradas pelo algoritmo da Aproximação Tangencial. No nosso problema,  $R$  também contém as restrições que impomos às variáveis duais para a inicialização do processo (ver seção IV.2.3).
- $V$  = conjunto unitário, formado pela restrição gerada numa dada iteração.

Seja  $(PM_{LD})^{r+1}$  o problema  $(PM_{LD})^r$  relaxado:

PASS00 :  $R$  é tal que a função dual  $\phi$  é limitada;

PASS01 : Resolver  $(PM_{LD})^{r+1}$ ; obter  $(p^{r+1}, \lambda^{r+1})$ ;

PASS02 : Se todas as restrições de  $M - R$  estão satisfeitas,  $(p^{r+1}, \lambda^{r+1})$  é ótima para  $(PM_{LD})$ ;

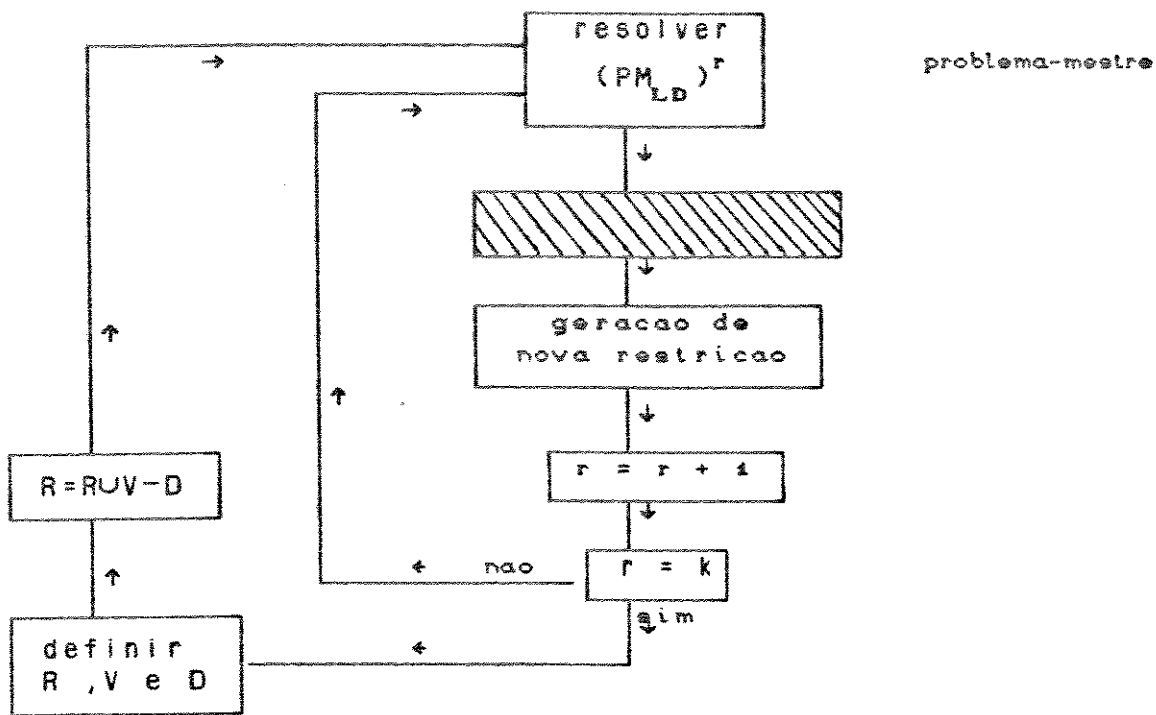
PASS03 :  $V \subset M - R$ ; faça  $R = R \cup V$ ,  $r = r + 1$   
e vá para PASS04;

PASS04 : Se  $r = k$ , seja  $D$  o conjunto de todas as restrições geradas e "folgadas";

faça  $R = R \cup V - D$  e vá para PASS01.

Se  $r < k$ , vá para PASS01.

Dessa forma, ao fluxograma da figura 5.2 devemos acrescentar o algoritmo acima ao "contarmos"  $k$ -restrições geradas (descrevemos abaixo apenas os blocos que interligam esta fase do algoritmo).



( figura 5.3 )

Os "PASSO1" e "PASSO2" estão embutidos nos blocos superior e hachurado, respectivamente, do fluxograma acima.

Devemos salientar que ao redefinirmos o conjunto  $R$  temos também que redefinir a matriz das restrições do problema  $(PM_{LD})^r$  e, conseqüentemente, reordenar os índices dos vetores que armazenam as variáveis básicas e as não-básicas do programa elaborado para o algoritmo Dual-Simplex Canalizado.

Após a relaxação das restrições não-ativas, a matriz básica obtida é uma submatriz da matriz básica anterior, cuja inversa deve ser calculada através de rotinas para tal fim.

Dessa forma, garantimos também que a cada "k" iterações do algoritmo da Aproximação Tangencial desprezamos as atualizações das inversas das matrizes-base conforme exposto na seção anterior, para invertermos uma nova matriz: isto significa que diminuimos os riscos de erros por arredondamentos que em geral aparecem quando utilizamos (21).

Um fator computacional que determina o parâmetro "k" é que, a cada conjunto D subtraído, devemos proceder na atualização da matriz das restrições, na reordenação das colunas básicas e não-básicas e na inversão da matriz-base resultante após a relaxação das restrições. Por este fato, certamente não nos é vantajoso aplicar a relaxação com muita frequência.

Por outro lado, não podemos trabalhar com matrizes muito grandes na resolução de  $(PM_{LD})^r$ , tendo em vista os pivoteamentos desnecessários que isto acarretaria ao algoritmo Dual-Simplex.

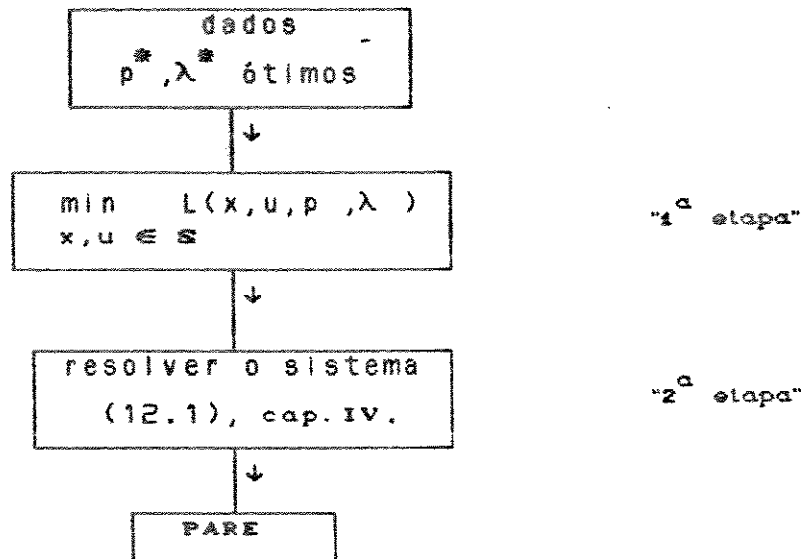
Obtida a solução dual ótima, devemos calcular, segundo tratamos em IV.3.2 e IV.3.3., os valores das variáveis básicas e não-básicas da solução ótima primal do (P.L.D.).

### V.3 - RESOLUÇÃO DA "2ª FASE"

Como vimos no capítulo anterior, a resolução da "2ª FASE" do algoritmo da Aproximação Tangencial para resolução de (P.L.D.) apresenta, a priori, duas etapas bastante simples de serem reconhecidas: a primeira refere-se ao cálculo dos valores das variáveis não-básicas (conforme IV.3.2) e, conseqüentemente, a determinação dos conjuntos N e B, respectivamente conjunto das colunas não-básicas e das colunas básicas da matriz de restrições do (P.L.D.) na solução ótima primal; a segunda etapa refere-se ao cálculo dos valores das variáveis básicas primais (conforme exposto em IV.3.3) após um rearranjo de colunas para

obtenção do sistema (12.1) , cap.IV, a ser resolvido.

Dessa forma, à primeira vista, o esquema geral para resolução é dado pela figura abaixo.



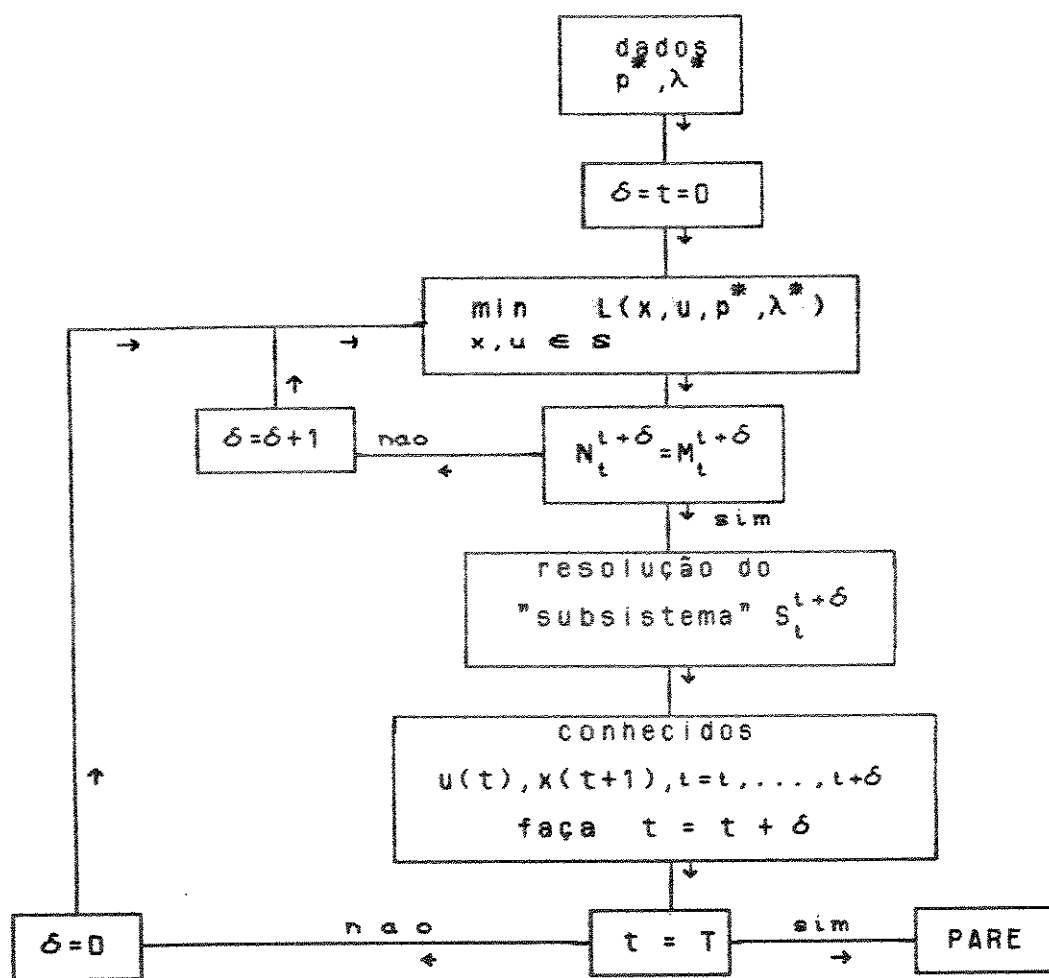
( figura 5.4 )

No entanto, as etapas acima mencionadas estão interligadas a cada estágio de tempo  $t$  ( $t=0, \dots, T-1$ ) devido à estrutura de "subsistemas" que o sistema (12.1) da "2ª etapa" em geral apresenta. Ou seja, a cada estágio de tempo  $t$ , obtemos através da "1ª etapa" o número de variáveis básicas  $u(t), x(t+1)$  que o sistema (12.1), seção IV.3, apresenta; se este número for igual ao número de equações do sistema naquele instante de tempo, então definimos um "subsistema" de (12.1) conforme  $(S_t)$ , seção IV.3.4. Caso contrário, acrescentamos o tempo em uma unidade e comparamos o

número de variáveis básicas nos instantes  $t$  e  $t+1$  com o número de equações neste mesmo período: caso forem iguais, obtemos um "subsistema" da forma  $(S_k^{t+\delta})$  de IV.3.4. Senão, acrescentamos novamente o tempo em uma unidade e assim procedemos sucessivamente, conforme procedimento descrito na seção IV.3.4.

O esquema geral do procedimento computacional desta FASE do algoritmo é dado na figura abaixo. Para isso, definimos

- $N_t^{t+\delta}$  - número de variáveis básicas no período  $(t, t+\delta)$ ,  
 $\delta \geq 0, \delta \in \mathbb{Z}$
- $M_t^{t+\delta}$  - número de equações do sistema no período  $(t, t+\delta)$ .



( figura 5.5 )

Dessa forma, a resolução da "2ª etapa" é a resolução de tantos "subsistemas" quantos os apresentados pelo sistema (12.1). cap.IV. Conforme estimativa a seguir, podemos esperar que o sistema (12.1) apresente vários "subsistemas" ( $S_k^{k+\delta}$ ) e, conseqüentemente, um custo computacional reduzido na sua resolução. Dado que o custo computacional da "1ª etapa" deve ser pequeno, podemos esperar que a técnica utilizada na "2ª FASE" do algoritmo aqui proposto seja eficiente.

#### V.3.1 - Estimativa sobre a Ocorrência de Não-Degenerescência em ( $P_{T-k}$ )

Conforme analisamos na seção IV.3, capítulo IV, a "2ª FASE" do algoritmo aqui proposto pode ser resolvida através de "subsistemas sequenciais", conseqüência extraída diretamente da não-degenerescência de ( $P_{T-k}$ ) (para algum k) e do Princípio de Otimalidade de Bellman.

Contudo, fica a indagação: É frequente a ocorrência de problemas ( $P_{T-k}$ ) não-degenerados? Podemos esperar que problemas com horizonte de planejamento grande sejam solucionados através desta metodologia?

Foi com o intuito de respondermos a estas dúvidas que realizamos vários testes com problemas do tipo (P.L.D.), observando a distribuição por estágios e/ou períodos de tempo das variáveis básicas na solução ótima de cada problema.

Nas tabelas abaixo são utilizadas as seguintes notações:

- T : horizonte de planejamento;
- NSIS : número de "subsistemas";
- RODAD : número de problemas testados;



- N : número de variáveis de estado por estágio de tempo;
- R : número de variáveis de controle por estágio de tempo.

Foram testados problemas bastante distintos, diferenças que foram impostas tanto na descrição de suas equações quanto no número de variáveis e horizonte de planejamento. Os resultados verificados estão descritos nas tabelas abaixo, as quais estão separadas pelo número de variáveis por estágio de tempo dos problemas rodados.

TABELA 1 : N = R = 2

T	NSIS>1	NSIS>2	NSIS>3	NSIS>4	NSIS>5	RODAD	% NSIS>1
12	10	8	7	4	2	10	100
8	8	6	4	2	1	10	80
4	5	2	0	—	—	10	50

TABELA 2 : N = R = 3

T	NSIS>1	NSIS>2	NSIS>3	NSIS>4	NSIS>5	RODAD	% NSIS>1
30	10	9	8	8	0	10	100
15	8	5	5	4	0	10	80
10	8	4	2	0	0	10	80
8	9	6	2	1	1	15	60
4	3	2	0	—	—	11	27

TABELA 3 : N = R = 4

T	NSIS>1	NSIS>2	NSIS>3	NSIS>4	NSIS>5	RODAD	% NSIS>1
48	10	10	9	7	7	10	100
36	9	8	8	7	5	10	100
26	6	6	5	3	0	10	60
12	3	2	2	1	1	11	27
8	3	2	2	2	0	11	27

As tabelas acima sugerem que, à medida em que o número de estágios aumenta, cresce o número de "subsistemas" (NSIS) apresentado pelo problema. Isto parece significar que o procedimento iterativo proposto neste trabalho para resolução da "2ª FASE" do algoritmo de Decomposição via Dualidade é mais apropriado a problemas do tipo (P.L.D.) com horizonte de planejamento grande.

#### V.4 - TESTES COMPUTACIONAIS

Para os testes computacionais elaboramos "pacotes" contendo os programas referentes às "etapas" e "FASES" descritas neste capítulo. A linguagem utilizada é o FORTRAN e os testes feitos em computador DEC system-10.

O pacote elaborado na implementação da "1ª FASE" do algoritmo tem como arquivo TANGDI.FOR e contém os seguintes

programas :

- LAGRA - subrotina que minimiza o Lagrangeano de um (P.L.D.);
- PASO1 - subrotina que soluciona o problema-mestre quando o primeiro hiperplano é gerado; aqui o problema se resume em minimizar uma função linear cujas variáveis são canalizadas.
- TABLO - nesta subrotina identificamos a matriz de restrições do problema-mestre, o "vetor-custo", o vetor de termos independentes e a matriz inversa da base;
- DUAL - subrotina contendo o algoritmo Dual-Simplex apresentado na seção V.2.1 ; a atualização da matriz inversa da base é feita de acordo com com (5.16).
- RELAX - subrotina que determina as restrições do problema-mestre a serem relaxadas, rearranja os índices das matrizes e dos vetores utilizados no programa.

O pacote elaborado na implementação da "2ª FASE" do algoritmo tem como arquivo FASEII.FOR e consiste das seguintes rotinas:

- COORD - subrotina que coordena as chamadas das subrotinas para esta fase do algoritmo; recupera a solução ótima primal a cada "subsistema" resolvido.
- BASVA - esta subrotina determina os valores das variáveis não-básicas primais associadas ao valor ótimo do multiplicador de Lagrange da "1ª FASE"; nesta subrotina determinamos quando um "subsistema" aparece na "2ª FASE";
- ARRANJ - detectado um "subsistema", esta subrotina é chamada de forma a rearranjar colunas básicas e colunas não-básicas , naquele período de tempo, para a sua resolução. A resolução dos "subsistemas" é feita através das rotinas DECOMP e SOLVE [20] que utilizam a Eliminação de Gauss para tal finalidade.

Dos problemas-testes utilizados para o método do Lagrangeano Aumentado, apenas ( PT-1 ) foi modificado, tendo suas restrições ( 2 ) ( ver seção 11.2 ) eliminadas. Os demais, ( PT-2 ), ( PT-3 ), ( PT-4 ) e ( PT-5 ) foram conservados para que possamos comparar os dois métodos no final deste trabalho.

( PT-1 )'    T = 4 ,   N = 2 ,   R = 3   e   M = 0

Valor Ótimo da Função : 377.636

Solução Ótima Primal : ( 20., 17.909, 0., 3.818, 10.909, 0., 41.454, 50., 45., 0., 0., 60., 0., 30., 25., 0., 5., 0., 0., 0. )

Solução Ótima Dual : ( .819234, -.6264, 2.63162, 1.728032, 2.021877, .813975, 1.194752, 5.610495 )

#### V.4.1 - Resultados Computacionais

Nos testes computacionais realizados utilizamos o "pacote" TANGDI - FASEII, descrito na seção anterior.

Para cada problema testado, houve diferenças com relação à etapa de resolução da "1ª FASE" do algoritmo em que empregamos a relaxação das restrições não-ativas do problema-mestre  $(PM_{LD})^r$ ; isto deveu-se às peculiaridades apresentadas pelos mesmos.

Nas tabelas a seguir utilizamos a seguinte nomenclatura:

- RESTR - número de hiperplanos gerados pelo algoritmo na "1ª FASE";
- TANGDI - algoritmo testado sem a Relaxação das restrições geradas;
- RELAX - algoritmo testado com Relaxação após geradas um número fixado de restrições;

- NSIS - número de "subsistemas" apresentado pelo problema na "2ª FASE";
- ITER - número total de iterações realizadas pelo algoritmo do Dual-Simplex Canalizado;
- CPU - tempo computacional medido em segundos;
- ICOL - número de variáveis duais do problema.

Em todos os testes realizados tomamos como vetor multiplicador de Lagrange inicial,  $\lambda^0$ , o vetor nulo. Para os problemas (PT-1), (PT-2) e (PT-3) o algoritmo convergiu apresentando os seguintes resultados:

TABELA 1 : Testes com ( PT-1 )

Limites Duais Fixados :  $\lambda_i^0 - 10 \leq \lambda_i^0 \leq \lambda_i^c + 10, i=1, \dots, ICOL$

Relaxação na "1ª FASE" : feita a cada 3 ICOL restrições geradas.

	RESTR	ITER	1ª FASE C P U	NSIS	C P U
TANGDI	33	81	1.91	3	2.17
RELAX	33	65	1.27	3	1.50

TABELA 2 : Testes com ( PT-2 )

Limites Duais Fixados :  $\lambda_i^0 - 10 \leq \lambda_i^0 \leq \lambda_i^c + 10, i=1, \dots, ICOL$

Relaxação na "1ª FASE" : feita a cada 2 ICOL restrições geradas.

	RESTR	ITER	1 <sup>a</sup> FASE C P U	NSIS	C P U
TANGDI	93	471	37.30	3	37.64
RELAX	113	430	12.72	3	13.06

TABELA 3 : Testes com ( PT-3 )

Limites Duais Fixados :  $\lambda_i^o - 10 \leq \lambda_i^c \leq \lambda_i^c + 10, i=1, \dots, ICOL$

Relaxação na "1<sup>a</sup> FASE" : feita a cada 2 ICOL restrições geradas.

	RESTR	ITER	1 <sup>a</sup> FASE C P U	NSIS	C P U
TANGDI	61	253	10.53	4	10.68
RELAX	63	220	4.19	4	4.42

Observação : Para os testes realizados com (PT-1), (PT-2) e (PT-3), os limites duais fixados inicialmente formaram o poliedro contendo a solução ótima dual para cada problema.

Nas tabelas acima notamos que o "CPU" para o algoritmo quando aplicamos relaxação nas restrições geradas é menor que o "CPU" apresentado pelo algoritmo quando não é utilizada esta técnica, sendo a diferença entre eles bastante acentuada para (PT-2) e (PT-3). Isto deve-se, principalmente, ao fato do número de operações, na resolução dos problemas-mestres, ser bem menor

quando utilizamos "RELAX", visto que a dimensão da matriz de restrições destes não ultrapassa o limite estabelecido pelo parâmetro que determina a relaxação, compensando sobremaneira os gastos operacionais adicionais com os cálculos da inversa da matriz básica que são realizados cada vez que o procedimento da Relaxação é aplicado.

Em todos os testes notamos o aparecimento de vários "subsistemas" na "2ª FASE" do algoritmo, o que implica em menor gasto operacional na resolução desta FASE. Para estes casos, podemos dizer que o algoritmo implementado para a decomposição do (P.L.D.) via Aproximação Tangencial mostra-se bastante viável computacionalmente.

TABELA 4 : Testes com (PT-4)

Limites Duais Fixados:  $\lambda_i^0 - 5 \leq \lambda_i^0 \leq 5 + \lambda_i^0$ ,  $i = 1, \dots, 100L$

Relaxação na "1ª FASE" : feita na 55-ésima restrição gerada.

Reinversão da Matriz-Base : feita na 30-ésima restrição gerada.

	RESTR	ITER	1ª FASE C P U	NSIS	C P U
TANGDI			(**)		
RELAX	257	—	4:34.55	07	4:36.16

(\*\*) Atingiu o limite máximo de CPU = 20:00.00 utilizado para sua execução sem convergir.

Nos testes com (PT-4) obtivemos um custo computacional

bastante elevado para RELAX, associado à resolução da "1ª FASE", sendo que para TANGDI o algoritmo não convergiu para o limite de "CPU" estabelecido. Este fato deveu-se principalmente pela necessidade maior de reinversão da matriz-base do algoritmo Dual Simplex durante o procedimento da Aproximação Tangencial.

A atualização da matriz-base conforme (21), seção V.2.1., foi necessária para acelerar a convergência do algoritmo em termos de "CPU": todavia, sua utilização mostrou ser bastante restrita neste problema, sendo que para parâmetros-de-relaxação maiores do que o apresentado na tabela acima o algoritmo da Aproximação Tangencial não convergiu .

Novamente a "2ª FASE" do algoritmo proposto obteve rápida resolução em termos de "CPU".

Para os testes com (PT-5) o algoritmo da Aproximação Tangencial não convergiu para a solução do problema dual.

A exigência de muitas reinversões da matriz-base aumentou consideravelmente o número de operações do algoritmo da Aproximação Tangencial, tornando desaconselhável sua utilização para problemas de porte médio e grande conforme implementação adotada neste trabalho.

Para a reinversão da matriz-base foi necessário utilizarmos subrotina de alta precisão numérica ( F04AMF/NAG - Unicamp ) que utiliza Decomposição de Householder . Subrotinas que possuem a decomposição L-U implementada não tiveram êxito na reinversão das matrizes-base apresentadas pelo problema, supostamente devido à sua estrutura numérica ( vide apêndice 3 e 4 ).



## Considerações:

A atualização da inversa da matriz-base segundo o algoritmo do Simplex Revisado ( através da forma produto de vetores elementares ) exige, por si só, uma periódica reinversão da mesma com o propósito de se evitar problemas numéricos. No algoritmo apresentado, além da atualização iterativa "revisada" da inversa da matriz-base, utilizamos também a atualização "pós-otimizada" a cada hiperplano gerado, o que aumenta os riscos de surgimento de erros por arredondamentos da máquina.

Allado ao fato acima mencionado, o número-de-condição verificado nas matrizes-base de (PT-4) e (PT-5) ( conforme ilustrado no apêndice 4 ) pode ter contribuído para o agravamento de problemas com a precisão numérica do algoritmo.

A utilização de variáveis-de-folga para a implementação do algoritmo Dual-Simplex obriga, necessariamente, que pelo menos uma variável-de-folga esteja na base por ocasião da geração de uma nova restrição para o problema-mestre da "1ª FASE" ( vide exemplo ilustrativo no Apêndice 3 ).

Podemos notar, como mostramos no Apêndice 4, que a existência de colunas relacionadas às variáveis-de-folga, na matriz-base, acarreta um aumento discrepante no número-de-condição da matriz.

## V.5. - CONCLUSOES

A técnica proposta nos capítulos IV e V para decomposição de

problemas do tipo (P.L.D.) possui aspectos bastante atrativos e dificuldades a serem superadas.

Dentre seus pontos atrativos estão a facilidade em ser inicializado, a facilidade em ter relaxadas as restrições geradas e não-ativas em sua "1ª FASE" e as boas características encontradas para uma eficiente resolução da "2ª FASE". Além disso, conforme demonstramos no capítulo IV, temos garantida a sua convergência.

A "2ª FASE" do algoritmo proposto apresenta-se bastante atrativa com sua estrutura de "subsistemas sequenciais", facilitando a resolução do sistema de equações lineares que fornece os valores das variáveis básicas da solução ótima primal. Dada a estimativa apresentada na seção V.3.1 e dados os gastos operacionais verificados, podemos dizer que a metodologia proposta para a resolução desta FASE é bastante eficiente e pode ser empregada para diversos casos.

Dos testes realizados com os problemas menores, (PT-1), (PT-2) e (PT-3), concluímos que a relaxação das restrições não-ativas deve ser utilizada para obtenção de maior eficiência computacional para o método. A atualização da matriz inversa, conforme seção V.2.1, deve ser utilizada para maior economia de operações; porém, um parâmetro que determina o limite máximo para essas atualizações deve ser fixado, dependendo do problema tratado, para que uma reinversão precisa da matriz-base seja realizada periodicamente e, conseqüentemente, sejam minimizados os riscos de surgimento de erros numéricos no algoritmo, causados por arredondamentos do computador.

Nos testes para (PT-4) e (PT-5) o algoritmo exigiu um grande

número de reinversões da matriz-base, causando sua sobrecarga operacional .

Dessa forma, para problemas de porte maior é necessário a implementação de técnicas mais "robustas" ( ou mais estáveis numericamente ) para a resolução do problema-mestre.

## Sugestões:

A técnica de decomposição proposta no capítulo IV e implementada conforme descrição deste capítulo deve ser modificada quanto ao tratamento dado à resolução do problema-mestre ( $PM_{LD}$ ) com o propósito de se evitar o surgimento de problemas numéricos e para obtenção de maior eficiência computacional (menor CPU).

A inicialização da técnica proposta neste trabalho deve ser mantida até que um número de hiperplanos igual a dimensão do problema tratado seja gerado. A partir daí sugerimos resolver o problema-mestre a partir do seu dual com o propósito de evitarmos o manejo da atualização da inversa conforme seção V.2.1.

O método Simplex Revisado pode ser empregado na resolução do problema dual dos problemas-mestre (PM) e deve ser implementado prevendo mais estabilidade numérica ao algoritmo. Para isso sugerimos que o cálculo da matriz-base, a cada iteração, seja feito através da decomposição L-U apresentada na forma produto [39].

A relaxação das restrições não-ativas pode ser empregada convenientemente com a finalidade de evitarmos problemas de memória computacional (dado que para problemas grandes o algoritmo deve gerar muitas restrições na "1 FASE") e maior custo operacional.

A adoção de um algoritmo que dispensa a utilização de variáveis-de-folga na resolução do problema-mestre evita o surgimento de matrizes-base com número-de-condição grande, conforme ilustrado no apêndice 4, e conseqüentemente o agravamento de problemas de estabilidade numérica. Com este propósito sugerimos a adoção do Método de Karmakar [32'] aplicado ao dual do problema-mestre.

## CONCLUSÃO

A origem de nossos estudos está relacionada com o interesse em resolvermos o problema da indeterminação da solução ótima primal de de um (P.L.D.) tendo conhecida sua solução ótima dual. Para tanto, analisamos o comportamento e características fundamentais de métodos do tipo dual, buscando sua viabilidade computacional e explorando, essencialmente, possibilidades de resolver o problema através da decomposição temporal do mesmo.

Em sua primeira parte, mais especificamente nos estudos sobre o método do Lagrangeano Aumentado, tivemos a oportunidade de demonstrar que esse método possibilita resolver problemas do tipo (P.L.D.) através da decomposição temporal da função Lagrangeano Aumentado associada, incluindo propriedades importantes para as funções apresentadas tais como diferenciabilidade, em todos os seus níveis, convexidade para os "1<sup>o</sup> e 2<sup>o</sup> níveis" e concavidade para a função dual relacionada ao problema aumentado (ou "3<sup>o</sup> nível"). Essas propriedades são fundamentais para a garantia de convergência do algoritmo. Em particular, a diferenciabilidade demonstrada para a função do "2<sup>o</sup> nível" é fundamental para o bom desempenho computacional do Lagrangeano Aumentado, pois essa propriedade possibilita a implementação de métodos "robustos" para resolução deste nível, garantindo convergência e eficiência para o algoritmo.

Os resultados obtidos indicam que o método sofre grande influência do valor adotado para o fator de penalidade sobre seus gastos computacionais, existindo sempre um valor que lhe confere melhor desempenho.

Em seu nível dual (3<sup>o</sup> nível), o método apresenta uma função côncava, diferenciável, apresentando melhor convergência para valores crescentes do fator de penalidade  $Q$ , o que sugere

(semelhantemente aos problemas não-lineares) que a função neste nível possui hessiana numericamente estável para esses mesmos valores de  $Q$ .

Dessa forma, o método do Lagrangeano Aumentado demonstrou, além dos bons resultados computacionais, possuir muitas propriedades importantes que o tornam um método com potencial de melhorias em termos de gastos operacionais, que pode ser explorado.

Na segunda parte do trabalho exploramos o método do Lagrangeano simples para obtenção de decomposição de problemas lineares dinâmicos. A nova abordagem tem como fundamento a descoberta da possibilidade de encontrarmos, dada a solução ótima dual do problema, a solução ótima primal através da decomposição de seu sistema de equações em "subsistemas sequenciais" distribuídos uniformemente por estágios de tempo. Essa decomposição foi provada na presença de não-degenerescência utilizando um dos conceitos básicos da Programação Dinâmica, o Princípio de Otimalidade de Bellman.

Ocorrendo degenerescência, comprovamos que a solução ótima primal pode ser obtida através de "subsistemas sequenciais" distribuídos não uniformemente por períodos de tempo.

Dessa forma, a técnica de decomposição via Lagrangeano simples foi apresentada em duas fases: a "1<sup>a</sup> FASE", da obtenção da solução ótima dual, e a "2<sup>a</sup> FASE", da obtenção da solução primal.

Testes realizados com uma gama variada de problemas do tipo (P.L.D.) apontaram sobre a utilidade da metodologia proposta para a "2<sup>a</sup> FASE" para problemas com horizonte grande, a qual apresentou viabilidade e eficiência computacional para os testes realizados.

Para a resolução da "1<sup>a</sup> FASE" foi implementada a técnica da Aproximação Tangencial de Geoffrion para a qual foram introduzidos limites artificiais iniciais para as variáveis duais do problema e utilizada a relaxação das restrições geradas e não-ativas com o propósito de obtenção de viabilidade e maior eficiência computa-

cional.

Os problemas testados comprovaram sobre a necessidade da relaxação das restrições não-ativas para obtenção de menores gastos com operações quando do emprego da técnica da Aproximação Tangencial na resolução da "1ª FASE" do algoritmo proposto.

A utilização do algoritmo dual-simplex na resolução do Problema-Mestre exigiu, para os problemas com maior dimensão, maior número de reinversões da matriz-base, o que acarretou aumento na carga computacional do algoritmo em sua "1ª FASE". A necessidade de muitas reinversões está relacionada com problemas de precisão numérica devido a prováveis erros por arredondamentos causados pela máquina e que podem ser evitados.

Dessa forma, apesar da técnica proposta via Lagrangeano Simples possuir convergência teoricamente assegurada, é uma técnica que exige cuidados especiais na implementação do algoritmo para a resolução da "1ª FASE". Sua viabilidade computacional depende essencialmente da obtenção de eficiência computacional na resolução da "1ª FASE", dado que a eficiência da resolução da "2ª FASE" está garantida.

#### Análise Comparativa:

O Método do Lagrangeano Aumentado foi introduzido ( Hestenes e Powell ) com o propósito de se evitar problemas de mal-condicionamento no Método das Penalidades quando a este último são aplicados valores grandes ao fator de penalização. Dessa forma, é um método que deve ser explorado utilizando essa característica que lhe é essencial. Aplicado à programação linear dinâmica, seu algoritmo apresenta em seus níveis, dual e primal, problemas a serem otimizados cujas funções apresentam importantes propriedades que garantem seu bom desempenho computacional.

Por sua vez, o Método de Decomposição via Lagrangeano Simples

pode apresentar convergência lenta em sua "1ª FASE" devido à característica da técnica da Aproximação Tangencial de gerar vários hiperplanos para atingir a solução dual do problema. Por este motivo, a relaxação das restrições geradas pelo procedimento e não-ativas é necessária, tanto do ponto-de-vista de minimizar a utilização de memória quanto o de economizar o número de operações. A "2ª FASE" deve ser facilmente resolvida devido à característica do sistema, relacionado às colunas básicas da matriz das restrições do (P.L.D.), de apresentar "subsistemas" que podem ser resolvidos sequencialmente: essa FASE deve exigir poucos gastos operacionais.

Ambos os métodos trabalham com dualidade e não necessitam de uma fase preliminar para inicializar o procedimento. O Método do Lagrangeano Aumentado determina simultaneamente as soluções dual e primal do problema, ao contrário do Método da Decomposição via Lagrangeano Simples que as obtém separadamente.

Dos resultados computacionais obtidos, podemos observar que, distintamente do segundo método abordado, o Método do Lagrangeano Aumentado apresenta boa taxa de convergência em seu nível dual: sua carga computacional está relacionada principalmente com as operações necessárias para a avaliação da função dual em cada ponto, apresentando-se crescente à medida em que cresce o valor do parâmetro de penalização.

O Método da Decomposição via Lagrangeano Simples apresenta grande trabalho computacional na resolução da "1ª FASE" - determinação da solução ótima dual: a fase relacionada à determinação da solução ótima primal é bastante eficiente, apresentando tempo de CPU pequeno.

Numericamente, o algoritmo do Lagrangeano Aumentado apresentou-se bem comportado para valores "não muito grandes" do fator de penalização. Valores muito pequenos para o mesmo fator causaram convergência lenta para o algoritmo em seu nível dual e menor



precisão numérica para a solução encontrada.

O algoritmo da Decomposição via Lagrangeano Simples apresentou problemas de precisão numérica em sua "1ª FASE" nos testes com (PT-4) e (PT-5), requisitando frequentes reinversões das matrizes - bases e, conseqüentemente, tempo de CPU grande.

Dessa forma, os resultados computacionais comprovam a análise teórica sobre a estrutura e comportamento de ambos os métodos aqui abordados: exceto com relação aos problemas numéricos apresentados na "1ª FASE" do segundo método, os quais se sobrepueram às nossas expectativas.

Ambos os métodos abordados neste trabalho apresentam possibilidades de melhoria em seu desempenho computacional a serem exploradas. Para os testes apresentados, o método do Lagrangeano Aumentado mostrou tempos computacionais menores do que os apresentados pela técnica via Lagrangeano Simples.

## Contribuições do Trabalho:

Este trabalho teve como objetivo analisar e propor novas técnicas duais na resolução de problemas lineares dinâmicos, procurando a decomposição temporal do problema, tendo em vista o problema da indeterminação da solução ótima primal via aplicação do Lagrangeano simples .

Para isso, analisamos uma nova estrutura para o algoritmo do Lagrangeano Aumentado, verificando propriedades das funções nos seus vários níveis as quais garantiram sua boa convergência. Estas propriedades possibilitaram uma nova implementação para o método, cujo comportamento foi analisado através de testes computacionais.

Sobre a hipótese de não-degenerescência, foi possível mostrar que os valores das variáveis básicas pertencentes à solução ótima primal de problemas lineares dinâmicos podem ser obtidos através de decomposição temporal do sistema de equações, conhecendo os valores ótimos das variáveis não-básicas correspondentes.

Obtida a solução ótima dual, os valores das variáveis não-básicas foram facilmente obtidos através de um simples e rápido problema de minimizar o Lagrangeano associado ao (P.L.D.).

Análise sobre a distribuição das variáveis básicas da solução ótima primal de problemas lineares dinâmicos indicaram para a viabilidade da aplicação deste novo procedimento.

A técnica proposta foi implementada, utilizando Aproximação Tangencial para obtenção da solução ótima dual; testes computacionais mostraram sua eficiência computacional.

## A P E N D I C E 1

Seja  $F(x,u)$  definida para  $u \in E^m$  e  $x \in S$ ,  $S$  espaço topológico.

hipóteses:  $S$  é compacto,  $F$  e  $\frac{\partial F}{\partial u_i}$  são funções contínuas.

Definimos  $\phi(u) = \min_{x \in S} F(x,u)$

$X(u) = \{ x / x \text{ minimiza } F(x,u) \text{ sobre } S \}$

TEOREMA1 : ( Danskin [ 13 ] ) Sob as hipóteses acima, a derivada direcional de  $\phi$  existe em qualquer direção  $s$ , para qualquer  $u \in E^m$  e é dada por

$$D \phi(u,s) = \min_{x \in X(u)} \sum_{i=1}^m s_i \frac{\partial F(x,u)}{\partial u_i}$$

TEOREMA2 : ( ver Lasdon [ 38 ] ) Assumidas as hipóteses acima, seja  $\frac{\partial F}{\partial u_i} = g_i(x)$ . A função  $\phi$  é diferenciável em  $u^*$  se e somente

se cada  $g_i$  é constante sobre  $X(u^*)$ . Neste caso, a derivada parcial de  $\phi$  é

$$\frac{\partial \phi}{\partial u_i} \Big|_{u=u^*} = g_i(x(u^*)), \text{ para qualquer } x \in X(u^*).$$

COROLARIO : Sob as hipóteses acima, se  $F(x, u^*)$  é minimizado sobre  $S$  em um único ponto  $x(u^*)$ , então  $\phi$  é diferenciável em  $u^*$ , com derivadas parciais  $\frac{\partial F}{\partial u_i} \Big|_{u=u^*} = g_i(x(u^*))$ .

Logo, a garantia de convexidade estrita de  $F$  assegura a diferenciabilidade de  $\phi$ .

## A P E N D I C E 2

### SOBRE A CONVERGÊNCIA DO MÉTODO DO LAGRANGEANO AUMENTADO NA RESOLUÇÃO DE PROBLEMAS NÃO-LINEARES

Consideremos problemas não-lineares na forma

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{s.a.} && h(x) = 0 \end{aligned} \quad (1)$$

onde  $x \in E^n$ ,  $h(x) \in E^m$  e  $f, h \in C^2$ .

Seja  $x^*$  um mínimo local de (1). Assumindo  $x^*$  ponto regular, então existe  $\lambda^*$ , multiplicador de Lagrange, tal que

$$\nabla f(x) + (\lambda^*)^T \nabla h(x) = 0 \quad (2)$$

e a hessiana do Lagrangeano  $l(x) = f(x) + (\lambda^*)^T h(x)$

$$L(x^*) = F(x^*) + (\lambda^*)^T H(x^*)$$

é semidefinida positiva sobre o subspaço tangente

$$M = \{ x / \nabla h(x^*) x = 0 \}$$

Para desenvolvermos a teoria que segue, devemos supor  $L(x^*)$  definida positiva; assim, temos garantida a convexidade estrita do lagrangeano  $l(x)$  em  $x^*$ .

Dessa forma,  $x^*$  é também mínimo local para o problema

$$\text{minimize } f(x) + (\lambda^*)^T h(x) \quad (3)$$

tendo em vista que o mesmo satisfaz as condições suficientes de primeira e segunda ordem para mínimo local. Além do mais, para qualquer  $\lambda$  suficientemente próximo de  $\lambda^*$ , a função  $f(x) + \lambda^T h(x)$  terá um mínimo local  $x$  próximo a  $x^*$ . Isto segue do fato de que, pelo Teorema da Função Implícita, a equação

$$\nabla f(x) + \lambda^T h(x) = 0 \quad (4)$$

tem uma solução  $x$  próxima de  $x^*$  quando  $\lambda$  está próximo de  $\lambda^*$ , pois  $L(x^*)$  é não-singular e, na solução  $x$ , a hessiana  $F(x) + \lambda^T H(x)$  é definida positiva.

Assim, temos que, localmente, existe uma correspondência biunívoca entre  $\lambda$  e  $x$  ao solucionarmos o problema irrestrito

$$\text{minimize } f(x) + \lambda^T h(x) \quad (5)$$

Mais ainda, esta correspondência é continuamente diferenciável [39].

Definimos, na vizinhança de  $\lambda^*$ , a função dual

$$\phi(\lambda) = \underset{x}{\text{mínimo}} [ f(x) + \lambda^T h(x) ] \quad (6)$$

para  $x$  próximo de  $x^*$ .

Denotamos por  $x(\lambda)$  a única solução de (5) na vizinhança de  $x^*$ . Os seguintes lemas são demonstrados e Luenberger [39].

LEMA 1 : A função dual tem gradiente

$$\nabla \phi(\lambda) = h(x(\lambda))^T$$

LEMA 2 : A hessiana da função dual é

$$-\nabla h(x(\lambda)) L^{-1}(x(\lambda), \lambda) \nabla h(x(\lambda))^T \quad (7)$$

onde  $L(x(\lambda), \lambda)$  denota a dependência da hessiana do Lagrangeano sobre  $\lambda$ .

Desde que  $L^{-1}(x(\lambda))$  é definida positiva e desde que  $\nabla h(x(\lambda))$  possui posto completo na vizinhança de  $x^*$ , então, pelo LEMA 2, a hessiana (7) é definida negativa.

Definimos a função Lagrangeano Aumentado relacionada ao problema (1) na forma

$$I_{\Delta}(x, \lambda, \alpha) = f(x) + \lambda^T h(x) + \alpha h(x)^T h(x), \quad \alpha > 0$$

A hessiana de  $I_{\Delta}$  no par  $x^*, \lambda^*$  é

$$\begin{aligned} L_{\Delta}(x^*, \lambda^*) &= F(x^*) + (\lambda^*)^T H(x^*) + \alpha \nabla h(x^*)^T \nabla h(x^*) \\ &= L(x^*) + \alpha \nabla h(x^*)^T \nabla h(x^*) \end{aligned}$$

Existe  $\alpha^*$  tal que  $L_{\Delta}$  é definida positiva para todo  $\alpha > \alpha^*$ , pois  $L_{\Delta}$  é a soma de  $L(x^*)$ , que é definida positiva no subespaço tangente  $M$ , com a matriz  $\nabla h(x^*)^T \nabla h(x^*)$  que é semidefinida positiva (vide Luenberger [39]).

Logo, os LEMAS 1 e 2 podem ser aplicados ao problema equivalente ao problema (1)

$$\begin{aligned} & \text{minimize } f(x) + q h(x)^T h(x) \\ & \text{s.a. } h(x) = 0 \end{aligned} \quad (8)$$

tendo em vista  $L_{\Delta}(x, \lambda, q)$  ser o seu Lagrangeano e possuir hessiana  $L_{\Delta}$  definida positiva, para todo  $q > q^*$ , no par  $x^*, \lambda^*$ .

Definimos a função dual relacionada ao problema (8)

$$\phi_{\Delta}(\lambda) = \min_x \{ f(x) + \lambda^T h(x) + q h(x)^T h(x) \} \quad (9)$$

na vizinhança de  $x^*, \lambda^*$ .

Aplicando os LEMAS 1 e 2, temos que  $h(x(\lambda))$  é o gradiente de  $\phi_{\Delta}$  e sua hessiana é

$$\nabla h(x(\lambda)) [ L(x(\lambda), \lambda) + q \nabla h(x(\lambda))^T \nabla h(x(\lambda)) ]^{-1} \nabla h(x(\lambda)) \quad (10)$$

Os autovalores da matriz hessiana (10) na solução  $x^*, \lambda^*$  descrevem o "comportamento" da função dual (9) e, conseqüentemente, determinam a taxa de convergência dos métodos que podem ser aplicados na sua maximização.

Luenberger [ 39 ] mostra que a razão entre o maior e o menor autovalores da matriz em (10) é dada por



$\frac{1/W + q}{1/w + q}$  , onde  $W$  e  $w$  são o maior e o menor autovalores de

$\nabla h(x^*) L^{-1}(x^*, \lambda^*) \nabla h(x^*)$ , que é a hessiana da função dual do problema (1).

Dessa forma, temos que à medida que a constante  $q$  cresce, a razão acima tende para a unidade, implicando em rápida convergência para o problema de maximizar  $\phi_A$ .

### A P E N D I C E 3

#### ILUSTRAÇÃO ITERATIVA DA TÉCNICA DA APROXIMAÇÃO TANGENCIAL EM PROBLEMAS DO TIPO (P.L.D.)

No exemplo que segue desenvolvemos algumas iterações da técnica da Aproximação Tangencial com o propósito de verificarmos a influência da introdução de variáveis-de-folga na matriz-base, utilizada pelo algoritmo Dual-Simplex, nas questões sobre condicionamento numérico da matriz.

Seja o problema na forma  $\min_x \{ f(x) \text{ s.a. } g(x)=0 \}$  com  $f$  e  $g$  funções lineares :

$$\begin{array}{ll} \min & 2x_1 + x_2 + 3x_3 - 2x_4 + 10x_5 \\ \text{s.a.} & x_1 + x_3 - x_4 + 2x_5 = 5 \\ & x_2 + 2x_3 + 2x_4 + x_5 = 9 \\ & x_1 \leq 7 \quad ; \quad x_4 \leq 5 \\ & x_2 \leq 10 \quad ; \quad x_5 \leq 3 \\ & x_3 \leq 1 \quad ; \quad x_l \geq 0 \quad , \quad l=1, \dots, 5 \end{array}$$

( solução :  $x_1=7$  ,  $x_2=1$  ,  $x_3=1$  ,  $x_4=3$  ,  $x_5=0$  )

Sejam  $\lambda_1$  e  $\lambda_2$  os multiplicadores de Lagrange do problema. Logo, a função Lagrangeano associada é

$$L(x, \lambda) = ( 2x_1 + x_2 + 3x_3 - 2x_4 + 10x_5 ) + \lambda_1 ( x_1 + x_3 - x_4 + 2x_5 - 5 ) + \lambda_2 ( x_2 + 2x_3 + 2x_4 + x_5 - 9 )$$

$$L(x, \lambda) = (2 + \lambda_1)x_1 + (1 + \lambda_2)x_2 + (3 + \lambda_1 + 2\lambda_2)x_3 + \\ (-2 - \lambda_1 + 2\lambda_2)x_4 + (10 + 2\lambda_1 + \lambda_2)x_5 - 5\lambda_1 - 9\lambda_2$$

Inicialização:  $\lambda = (\lambda_1, \lambda_2) = (0, 0)$  e

1ª iteração : minimizando  $L(x, \lambda)$  na variável  $x$  obtemos

$$x_1 = x_2 = x_3 = x_5 = 0 \quad \text{e} \quad x_4 = 5$$

Dessa forma, o primeiro hiperplano gerado é

$$z^0 \leq f(x) + \lambda g(x) = -10 - 10\lambda_1 + \lambda_2 \quad (h1)$$

e devemos resolver o Problema-Mestre (PM)<sub>1</sub>

$$\begin{aligned} \min \quad & -z^0 \\ \text{s.a.} \quad & z^0 + 10\lambda_1 - \lambda_2 \leq -10 \\ & -10^9 \leq \lambda_1 \leq 10^9 \\ & -10^9 \leq \lambda_2 \leq 10^9 \end{aligned}$$

Fazendo  $z^0 = -10 - 10\lambda_1 + \lambda_2 - \lambda_3$  ( $\lambda_3 \geq 0$ ) encontramos a solução  $\lambda_1 = -10^9$ ,  $\lambda_2 = 10^9$  e  $\lambda_3 = 0$  e  $z^0 = -10 + 11 \times 10^9$ .

2ª iteração : minimizando  $L(x, \lambda)$  encontramos a solução

$$x_1 = 7, \quad x_2 = x_3 = x_4 = 0, \quad x_5 = 3$$

Logo, o segundo hiperplano gerado é  $z^0 \leq 44 + 8\lambda_1 - 6\lambda_2$  (h2) e devemos resolver (PM)<sub>2</sub>

$$\begin{aligned} \min \quad & 10\lambda_1 - \lambda_2 + \lambda_3 \\ \text{s.a.} \quad & -18\lambda_1 + 7\lambda_2 - \lambda_3 + \lambda_4 = 54 \\ & -10^9 \leq \lambda_1, \lambda_2 \leq 10^9 \\ & \lambda_3, \lambda_4 \geq 0 \end{aligned}$$

"Tableau" inicial:

-18	7	-1	1	54 - 25×10 <sup>9</sup>
10	-1	1	0	

Base inicial =  $x_3$  ;                   entra na base :  $x_2$

"Tableau" final :

1	-7/18	1/18	-1/18	[-54 - 7×10 <sup>9</sup> ] / 18
0	26/9	4/9	5/9	

cuja solução é  $\lambda_1 = \frac{-54}{18} - \frac{7}{18} * 10^9$  ,  $\lambda_2 = -10^9$  ,  $\lambda_3 = 0 = \lambda_4$

3ª iteração : minimizando  $L(x,\lambda)$  obtemos

$$x_1 = 7, x_2 = 10, x_3 = 1, x_4 = 5 \text{ e } x_5 = 3$$

e o novo hiperplano gerado é  $z^0 \leq 47 + 4\lambda_1 + 16\lambda_2$  (h3)

Resolvemos (PM)<sub>3</sub> :

$$\begin{aligned} \min \quad & 10\lambda_1 - \lambda_2 + \lambda_3 \\ \text{s.a.} \quad & -18\lambda_1 + 7\lambda_2 - \lambda_3 + \lambda_4 = 54 \\ & -14\lambda_1 - 15\lambda_2 - \lambda_3 + \lambda_5 = 57 \\ & -10^9 \leq \lambda_1, \lambda_2 \leq 10^9 \\ & \lambda_i \geq 0, \quad i = 3, 4, 5 \end{aligned}$$

"Tableau" inicial:

1	-7/8	1/18	-1/18	0		[-54 - 7x10 <sup>9</sup> ] / 18
0	-18/9	-2/9	-7/9	1		15 - $\frac{184 \times 10^9}{9}$
<hr/>						
0	269	4/9	5/9	0		

"Tableau" final:

1	0	11/184	-135/	-7/368		-10881/3512
0	1	1/92	7/184	-9/184		-135/184
<hr/>						
0	0	19/46	41/92	13/92		

4ª iteração : minimizando  $L(x,\lambda)$  obtemos

$$x_1 = 7, x_2 = x_5 = 0, x_3 = 1, x_4 = 5$$

e o novo hiperplano gerado é  $z^0 \leq 7 - 2\lambda_1 + 3\lambda_2$  (n4)

Dessa forma,  $(PM)_4$  é dado por

$$\begin{aligned}
 \min \quad & 10\lambda_1 - \lambda_2 + \lambda_3 \\
 \text{s.a.} \quad & -18\lambda_1 + 7\lambda_2 - \lambda_3 + \lambda_4 = 54 \\
 & -14\lambda_1 - 15\lambda_2 - \lambda_3 + \lambda_5 = 57 \\
 & -8\lambda_1 - 2\lambda_2 - \lambda_3 + \lambda_6 = 17 \\
 & -10^9 \leq \lambda_1, \lambda_2 \leq 10^9 \\
 & \lambda_i \geq 0, \quad i = 3, 4, 5, 6
 \end{aligned}$$

"Tableau" inicial :

1	0	11/184	-15/368	-7/368	0	-10881/3512
0	1	1/92	7/184	-9/184	0	-135/184
0	0	-1/2	-1/4	-1/4	1	-9.253268
0	0	19/46	41/92	13/92	0	

"Tableau" final :

1	0	9/92	-1/46	0	- 7/92	-2.3941816
0	1	5/46	2/23	0	- 9/46	1.0767264
0	0	2	1	1	-92/23	37.013073
0	0	3/23	7/23	0	13/23	

5ª iteração : minimizando  $L(x,\lambda)$  obtemos

$$x_1 = 7, \quad x_2 = x_3 = x_4 = x_5 = 0$$

e temos o novo hiperplano  $z^0 \leq 14 + 2\lambda_1 - 9\lambda_2$  (h5)

Assim,  $(PM)_5$  é  $(PM)_4$  adicionado da nova restrição.  
Atualizando (h5) no "tableau" anterior obtemos

"Tableau" inicial :

1	0	9/92	-1/46	0	- 7/92	0	-2.3941816
0	1	5/46	2/23	0	- 9/46	0	1.0767264
0	0	2	1	1	-92/23	0	37.013073
0	0	-21/23	-26/23	0	24/23	1	-15.497443
<hr/>							
0	0	3/23	7/23	0	13/23	0	

Verificamos o número-de-condição das matrizes-base dos "tableau's" apresentados no exemplo acima :

3<sup>a</sup> iteração : Seja A matriz-base ; COND : número de condição.

$$A(\text{inicial}) = \begin{bmatrix} -18 & 0 \\ -14 & 1 \end{bmatrix} \quad \text{COND}(A) = 28.873$$

$$A(\text{final}) = \begin{bmatrix} -18 & 7 \\ -14 & -15 \end{bmatrix} \quad \text{COND}(A) = 1.4835$$

4<sup>a</sup> iteração :

$$A(\text{inicial}) = \begin{bmatrix} -18 & 7 & 0 \\ -14 & -15 & 0 \\ -8 & -2 & 1 \end{bmatrix} \quad \text{COND}(A) = 26.279$$

$$A(\text{final}) = \begin{bmatrix} -18 & 7 & 0 \\ -14 & -15 & 1 \\ -8 & -2 & 0 \end{bmatrix} \quad \text{COND}(A) = 105.2891$$

5ª iteração :

$$A(\text{final}) = \begin{bmatrix} -18 & 7 & 0 & 0 \\ -14 & -15 & 1 & 0 \\ -8 & -2 & 0 & 0 \\ -12 & 10 & 0 & 1 \end{bmatrix} \quad \text{COND}(A) = 120.0927$$

Como podemos notar, a entrada de uma variável-de-folga como uma variável básica, no algoritmo Dual-Simplex, pode contribuir fortemente para piorar o número-de-condição da matriz-base.



## A P E N D I C E 4

Neste apêndice verificamos o número-de-condição para algumas matrizes-bases utilizadas pelo algoritmo Dual-Simplex durante a resolução do problema-mestre da técnica da Aproximação Tangencial. Nestas matrizes, a última linha sempre refere-se aos coeficientes das variáveis básicas na função objetivo.

1 - Para o problema-teste (PT-1), a matriz-base obtida da relaxação das restrições não-ativas na vigésima restrição gerada pelo algoritmo é :

350	140	-50	-50	-10	0	0	0	-1	0	0
230	-100	0	-100	-50	250	-20	110	-1	0	0
50	140	-50	-50	140	250	-50	50	-1	0	0
50	140	50	-100	50	150	0	0	-1	0	0
150	240	-100	-100	0	50	30	110	-1	0	0
350	140	-100	-50	-50	150	30	110	-1	0	0
30	100	0	0	0	200	-50	50	-1	0	0
30	100	-50	0	40	150	30	110	-1	0	0
350	140	100	-200	-10	200	-50	0	-1	0	0
330	100	-100	-200	-10	100	0	50	-1	1	0
-120	-105	40	35	-15	-145	-7	-82	1	0	1

O número-de-condição desta matriz-base é 2862,1 .

As 9<sup>a</sup> e 10<sup>a</sup> colunas são referentes a variáveis-de-folga; a 9<sup>a</sup> coluna é relativa à variável-de-folga da restrição que descreve z<sup>0</sup> e a 10<sup>a</sup> coluna é relativa à variável-de-folga da última

restrição gerada antes da relaxação.

O número-de-condição da matriz acima tomando-se apenas como elementos os coeficientes das restrições geradas é 28,446 .

2 - Para o problema-teste (PT-2) tomamos a matriz-base obtida da primeira relaxação na vigésima restrição gerada ; descremos essa

matriz por  $BM = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$  sendo

$$A = \begin{bmatrix} 210 & -1 & 180 & 0 & 90 & 60 & 120 \\ 340 & -1 & 60 & -70 & -220 & -120 & 180 \\ 270 & -1 & -180 & -70 & 0 & 70 & 60 \\ 200 & -1 & 60 & -210 & 0 & 130 & 50 \\ 200 & -1 & -60 & -140 & -100 & 70 & -70 \\ 200 & -1 & -120 & -20 & -50 & -60 & 50 \\ 200 & -1 & -120 & 0 & 0 & -60 & -10 \end{bmatrix}$$

$$C = \begin{bmatrix} 210 & -1 & 180 & 0 & -150 & 60 & 50 \\ 70 & -1 & 120 & -140 & 0 & -60 & 180 \\ 270 & -1 & 120 & -90 & 0 & 70 & 120 \\ 340 & -1 & 120 & 0 & -70 & 0 & 50 \\ 270 & -1 & 120 & -140 & 0 & 70 & 50 \\ 270 & -1 & 60 & 0 & 0 & -60 & 180 \\ 200 & -1 & 60 & -70 & -150 & 70 & 180 \\ -210 & 1 & -20 & 58 & 30 & -18 & -40 \end{bmatrix}$$

$$B = \begin{bmatrix} -160 & 140 & 50 & 10 & 0 & 0 & 0 & 0 \\ 0 & 90 & 50 & -40 & 40 & 0 & 0 & 0 \\ -120 & 90 & 180 & -40 & 40 & -120 & 0 & 0 \\ -40 & 280 & 100 & 0 & -40 & 40 & 0 & 0 \\ -120 & 140 & -50 & 10 & 0 & 0 & 0 & 0 \\ 0 & 280 & 50 & -140 & 40 & 0 & 0 & 0 \\ -40 & 90 & 150 & 0 & -40 & 40 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 210 & 40 & -40 & 40 & -40 & 0 & 0 \\ -80 & 140 & 140 & -40 & 40 & -80 & 0 & 0 \\ 0 & 140 & 140 & -140 & 0 & -40 & 0 & 0 \\ -160 & 210 & 140 & -40 & 0 & -80 & 0 & 0 \\ 0 & 210 & 100 & -40 & 40 & 0 & 0 & 0 \\ -120 & 140 & 40 & 50 & -40 & -120 & 0 & 0 \\ -120 & 70 & 0 & -90 & 0 & 0 & 1 & 0 \\ 80 & -155 & -65 & 20 & -5 & 10 & 0 & 1 \end{bmatrix}$$

A matriz  $BM$  assim definida possui número-de-condição igual a 3122,2 . Novamente temos a presença de duas colunas associadas às variáveis-de-folga : a que pertence à restrição que descreve  $z^0$  e aquela que pertence à última restrição gerada.

Ao eliminarmos as colunas que possuem apenas elementos unidades e zeros, obtemos uma matriz cujo número-de-condição é 16,3372.

3 - Para o problema-teste (PT-4) tomamos a matriz-base na 96<sup>ª</sup> restrição gerada e por ocasião da quarta relaxação das restrições

não-ativas. Definimos essa matriz por BM4, sendo

$$BM4 = \begin{bmatrix} A4 & B4 \\ C4 & D4 \end{bmatrix}$$

onde A4, B4, C4 e D4 estão definidas abaixo.

A matriz BM4 assim definida possui número-de-condição igual a 51097,00. Novamente temos a presença de três colunas associadas a variáveis-de-folga : a que pertence à restrição que descreve  $z^0$ , outra relacionada à variável-de-folga da 29 restrição e aquela que pertence à última restrição gerada.

Ao eliminarmos as colunas que possuem apenas elementos unidades e zeros, obtemos uma matriz cujo número-de-condição é 28,3405.

$$A4 = \begin{bmatrix} 280 & 80 & -320 & 50 & -80 & 0 & 50 & 50 & -100 & -300 & 0 & 240 & 70 & 100 & -1 \\ 280 & 240 & -320 & -50 & -80 & -80 & -100 & -130 & -150 & -200 & 60 & 80 & -100 & -170 & -1 \\ 350 & 180 & -400 & -280 & -80 & -240 & 0 & 110 & -100 & -300 & -120 & 320 & -300 & -170 & -1 \\ 0 & 160 & -240 & -230 & 0 & 160 & -50 & 50 & -100 & 0 & 60 & 80 & -100 & -170 & -1 \\ 70 & 80 & -160 & -180 & 0 & -160 & 0 & -240 & 50 & -300 & 0 & 320 & -300 & -100 & -1 \\ 0 & 160 & -400 & 0 & -160 & -80 & 50 & 60 & 150 & -300 & -120 & 240 & -230 & -170 & -1 \\ 0 & -80 & -160 & 0 & -160 & -80 & -50 & -180 & -150 & -300 & -120 & 320 & 0 & 100 & -1 \\ 0 & 260 & -400 & -80 & 0 & -80 & -50 & -180 & -100 & -300 & -120 & 80 & -200 & 30 & -1 \\ 140 & -80 & -80 & -50 & -80 & -80 & 0 & -60 & 50 & -300 & -180 & 80 & -30 & -170 & -1 \\ 70 & 340 & -320 & -180 & -80 & -80 & -100 & 50 & 0 & -300 & 60 & 240 & 0 & 30 & -1 \\ -70 & 0 & -240 & -80 & -80 & -240 & 50 & -10 & -50 & -100 & 0 & 240 & -100 & -170 & -1 \\ 0 & 80 & -400 & -80 & 80 & -240 & 50 & -70 & -150 & -300 & 60 & 80 & 70 & -70 & -1 \\ 70 & 80 & -320 & -180 & -80 & 80 & 50 & -60 & 50 & -200 & -180 & 240 & 0 & 100 & -1 \\ 140 & -80 & -160 & -100 & -160 & 0 & -100 & 0 & 150 & -300 & 60 & 240 & -300 & -100 & -1 \\ 280 & 0 & -240 & -80 & -80 & -240 & 50 & 50 & -150 & -100 & 0 & 240 & -200 & 30 & -1 \end{bmatrix}$$

C4 =

140	180	-180	-30	80	-80	0	-120	-50	-200	60	320	-100	-170	-1
0	180	-400	-80	0	-240	-50	-80	-50	-300	0	80	-100	-170	-1
140	0	-400	-30	320	-240	0	0	0	-200	60	160	-200	30	-1
70	0	-80	-130	0	0	-100	50	-150	-200	-120	240	-300	-170	-1
140	180	-320	50	-180	-180	50	0	0	0	60	320	70	30	-1
140	0	-320	-280	0	-80	-50	-60	50	-200	0	320	-30	-170	-1
140	0	-320	-80	0	0	150	-130	-200	-300	60	80	-300	-170	-1
210	80	-80	-80	-80	160	150	110	-50	-300	60	320	100	130	-1
140	80	-320	50	-80	0	-100	60	0	-100	-120	320	-300	-170	-1
70	180	-240	-80	-80	80	-100	110	-50	-300	-80	320	-300	-100	-1
70	160	-160	-80	80	-80	150	-60	-50	-200	0	80	-100	200	-1
0	80	-160	-180	-80	-160	100	-180	-50	-200	60	240	-130	-70	-1
70	80	-80	0	-160	80	100	-70	-150	-300	60	320	-300	-170	-1
70	100	-80	-180	80	0	0	0	100	-300	0	80	0	30	-1
-98	-91	284	97	45	86	15	50	55	235	40	-190	100	67	-1

B4 =

0	50	-50	0	0	0	-200	-50	-280	0	50	50	-210	0	0
100	0	0	100	200	-80	-100	50	-140	180	0	0	70	0	0
-130	50	0	100	-100	-80	0	200	-210	-60	0	-50	-70	0	0
-100	50	-170	100	0	-160	-100	50	-140	240	120	-150	140	0	0
-80	-50	-50	150	0	80	0	50	-210	120	120	-230	-140	0	0
-50	50	-170	50	100	-160	0	50	-210	-60	120	-100	140	0	0
-180	150	-100	150	100	-80	-200	100	-140	60	120	-200	0	0	0
0	0	0	50	-100	-160	-100	-50	-140	60	170	0	70	0	0
-180	-50	50	150	0	-80	-100	50	-210	60	0	-150	140	0	0
0	0	50	50	-100	-80	-200	-50	-140	180	0	-150	70	0	0
0	-50	-50	50	0	-80	-100	0	0	120	120	0	70	0	0
0	0	0	150	0	0	0	100	-210	0	70	0	70	0	0
100	-50	50	100	100	-160	-200	50	-140	60	0	0	0	0	0
100	100	0	100	0	-160	0	50	-210	240	0	-50	-140	0	0
-100	50	120	50	0	-80	-100	0	-210	0	140	-200	-70	0	0

D4 =

-130	50	0	0	100	-160	-100	0	-210	0	0	-100	-70	0	0
-100	100	70	150	-100	-80	-100	100	-140	120	140	-200	70	0	0
-130	-50	-100	50	200	-80	-100	150	-140	180	50	-100	140	0	0
100	50	-170	100	100	0	0	50	-140	80	120	0	210	0	0
-130	-50	-100	100	0	-80	-200	200	-280	180	120	-100	0	0	0
-130	100	-50	50	200	0	-100	-50	-280	120	120	0	0	0	0
-100	0	-100	-100	100	-180	0	200	-210	120	50	-150	-70	0	0
-180	-50	-100	50	0	-160	-100	200	-140	80	120	-150	70	0	0
-130	0	50	-50	-100	80	0	0	-140	-80	0	-230	140	0	0
-180	-50	-100	100	100	-80	0	50	-210	120	120	-150	-140	0	0
0	250	50	50	200	0	0	150	-140	120	0	-50	0	0	0
0	50	50	-150	100	-80	100	100	-210	120	70	-200	70	0	0
-80	50	50	100	0	-80	0	50	-140	0	0	-50	70	0	0
0	50	-100	-50	0	0	-200	200	-140	0	50	-200	70	1	0
37	-23	-5	-95	-35	90	95	-60	172	-80	-75	85	-27	0	1

## B I B L I O G R A F I A

- [ 1 ] Armentano, V.A.: "Programação Linear Dinâmica" : Tese de Mestrado apresentada à FEC, Unicamp, Campinas, 1979.
- [ 2 ] Bazaraa, M.; Goode, J.J.: "A Survey of Various Tactics for Generating Lagrangian Multipliers in the Context of Lagrangian Duality": European J.Op.Research 3: pp.322-338: 1979.
- [ 3 ] Bazaraa, M.; Jarvis, J.: "Linear Programming and Network Flows": John Wiley, 1989.
- [ 4 ] Bazaraa, M.; Shetty, C.M.: "Nonlinear Programming : Theory and Algorithms": New York, Wiley, 1979.
- [ 5 ] Bellman, R.; Kalaba, R.: "Dynamic Programming and Modern Control Theory": Academic Press, New York, 1965.
- [ 6 ] Bertsekas, D.P.: "Combined Primal-dual and Penalty Methods for Constrained Minimization": SIAM J. Control, vol.13, n<sup>o</sup> 3, 1975.
- [ 7 ] Bertsekas, D.P. : "Multiplier Methods : A Survey" : Automatica, vol.12, 133-145, 1976.
- [ 8 ] Bertsekas, D.P.: "On the Method of Multipliers for Convex Programming": IEEE Automatic Control, Junho 1975.
- [ 9 ] Bertsekas, D.P.: "On Penalty and Multiplier Methods for Constrained Minimization": SIAM J.Control, vol.14, n<sup>o</sup> 2, 1976.
- [10] Biagio, M.A.: "Análise de Algoritmos para Minimização de Funções sem Restrições": Tese de Mestrado apresentada ao IMECC, Unicamp, Campinas, 1981.
- [11] Boggs, P.T.; Tolle, J.W.: "Augmented Lagrangians Which are Quadratic in the Multiplier": J.O.T.A., vol.31, n<sup>o</sup> 1, 1980.
- [12] Brooks, R.; Geoffrion, A.: "Finding Everett's Lagrange Multipliers by Linear Programming": Op.Research, vol.14, 1149-1153, 1966.

- [13 ] Danskin, J.M.: "The Theory of Max-Min with Applications": J.SIAM Appl.Math., vol.14, n° 4, 1966.
- [14 ] Dantzig, G.B.; Wolfe, P.: "Decomposition Principle for Linear Programs": The Rand Corporation, Santa Monica, CA, 1960.
- [15 ] Demjanov, V.F.: "Algorithms for Some Minimax Problems": J.Comp.Systems Sci., vol.2, 342-380, 1968.
- [16 ] Demjanov, V.F.: "On the Maximization of a Certain Nondifferentiable Function": J.O.T.A., vol.7, n° 2, 1971.
- [17 ] Di Pillo, G.; Grippo, L.: "A New Class of Augmented Lagrangeans in Nonlinear Programming": SIAM J.Control and Opt., vol.17, n° 5, 1979.
- [18 ] Everett, H.: "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources": Op.Research, vol.11, 399-414, 1963.
- [19 ] Forsythe, G.E.; Moler, C.B.: "Computer Solutions of Linear Algebraic Systems": Prentice-Hall, New Jersey, 1967.
- [20 ] Forsythe, G.E.; Malcolm, M.A.; Moler, C.B.: "Computer Methods for Mathematical Computations", Prentice-Hall, New Jersey, 1977.
- [21 ] Friedlander, A.: "Otimização com Estrutura Escada nas Restrições": Tese de Doutorado apresentada à FEC, Unicamp, Campinas, 1986.
- [22 ] Geoffrion, A.M.: "Reducing Concave Programs with Some Linear Constraints": SIAM J.Appl.Math., vol.15, n° 3, 1967.
- [23 ] Geoffrion, A.M.: "Duality in Nonlinear Programming : A Simplified Applications - Oriented Development": SIAM Review, vol.13, n° 1, 1971.
- [24 ] Glad, S.T.: "A Combination of Penalty Function and Multiplier Methods for Solving Optimal Control Problems": J.O.T.A., vol.28, n° 3, 1979.
- [25 ] Glad, S.T.: "Properties of Updating Methods for the Multipliers in Augmented Lagrangians": J.O.T.A., vol.28, n° 2, 1979.



- [26 ] Glassey, C.R.: "Nested Decomposition and Multi-Stage Linear Programs": Management Science, vol.20, n° 3, 1973.
- [27 ] Granville, S.; Schechtman, J.: "An Application of the Augmented Lagrangian Approach to Decomposition in Linear Programming": Investigación Operativa, vol.1, n° 1, 1988.
- [28 ] Hestenes, M.R.: "Survey Paper - Multiplier and Gradient Methods": J.O.T.A., vol.4, n° 5, 1969.
- [29 ] Hestenes, M.R.: "Conjugate Direction Methods in Optimization": Springer-Verlag, New York, 1980.
- [30 ] Ho, J.K.; Lee, T.C.; Sundarval, R.P.: "Decomposition on Linear Programs Using Parallel Computation": Math. Programming, vol.42, 391-405, 1988.
- [31 ] Ho, J.K.; Manne, A.S.: "Nested Decomposition for Dynamic Models": Math. Programming, n° 6, 1974.
- [32 ] Jackson, P.L.; Lynch, D.F.: "Revised Dantzig-Wolfe Decomposition for Staircase - Structured Linear Programs": Math. Programming, vol.39, n° 2, 1987.
- [32'] Karmarkar, N.: "A New Polynomial-Time Algorithm for Linear Programming": Combinatorica, n° 4, 373-395, 1984.
- [33 ] Kelley, J.E.: "The Cutting-Plane Method for Solving Convex Programs": J.Soc. Indust. Appl. Math., vol.8, n° 4, 1960.
- [34 ] Kennington, J.L., Helgason, R.V.: "Algorithms for Network Programming": John Wiley & Sons, 1980.
- [35 ] Krivonozhko, V.E.: "A Decomposition Algorithm for Linear Dynamic Programming Problems": Engineering Cybernetics, vol.15, n° 3, 1977.
- [36 ] Krivonozhko, V.E.; Chebotarev, S.P.: "Factorization Method for Linear Dynamic Programming Problems": Aut.Remote Control, n° 7, 1976.
- [37 ] Krivonozhko, V.E.; Chebotarev, S.P.: "Triangular Factoring of Matrices in Penalty Function Method for Solving Problems of Linear Dynamic Programming": Aut.Remote Control, n° 6, 1977.
- [38 ] Lasdon, L.S.: "Optimization Theory for Large Systems": The Macmillan Company, London, 1970.

- [39 ] Luenberger, D.G.: "Introduction to Linear and Nonlinear Programming": Addison-Wesley, Massachusetts, 1984
- [40 ] Martensson, K.: "A New Approach to Constrained Function Optimization": J.O.T.A., vol.12, n<sup>o</sup> 6, 1973.
- [41 ] Miele, A.; Cragg, E.E.; Iyer, R.R.; Levy, A.V.: "Use of the Augmented Penalty Function in Mathematical Programming Problems. PART 1": J.O.T.A., vol.8, n<sup>o</sup> 2, 1971.
- [42 ] Miele, A.; Moseley, P.E.; Levy, A.V.; Coggins, G.M.: "On the Method of Multipliers for Mathematical Programming Problems": J.O.T.A., vol.10, n<sup>o</sup> 1, 1972.
- [42'] Oliveira, A.R.L.: "Métodos de Ponto Interior em Programa - ção Linear - Estudo e Implementação "; Tese de Mestrado apre - sentada na FEE/Unicamp , outubro , 1989.
- [43 ] O'Neill, R.P.: "Nested Decomposition of Multistage Convex Programs": SIAM J.Control and Opt., vol.14, n<sup>o</sup> 3, 1976.
- [44 ] Perlotto, A.J.: "Um Método Dual-Simplex para Problemas de Programação Linear com Estrutura Bloco-Angular e Variáveis Canalizadas": Tese de Mestrado apresentada à FEC, Unicamp, Campinas, 1983.
- [45 ] Propoi, A.I.; Yadykin, A.B.: "A Modification of Duality Relation in Dynamic Linear Programming": Aut.Remote Control, n<sup>o</sup> 5, 1975.
- [46 ] Propoi, A.I.; Yadykin, A.B.: "A Parametric Iterative Method for the Solution of Linear Dynamic Programming Problems": Aut.Remote Control, n<sup>o</sup> 12, 1975.
- [47 ] Propoi, A.I.; Krivonozhko, V.E.: "The Simplex Method for Dynamic Linear Programs": RR-78-14, International Institute for Applied Systems Analysis (IIASA), Luxemburg, Áustria, setembro 1978.
- [48 ] Rockafellar, R.T.: "A Dual Approach to Solving Nonlinear Programming Problems by Unconstrained Optimization": Math.Programming, vol.5, 354-373, 1973.
- [49 ] Rockafellar, R.T.: "Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming": SIAM J.Control, vol.12, n<sup>o</sup> 2, 1974.

- [50 ] Rockafellar, R.T.: "Convex Analysis": Princeton University, Princeton, New Jersey, 1970.
- [51 ] Rockafellar, R.T.: "The Multiplier Method of Hestenes and Powell Applied to Convex Programming": Math.Programming, vol.12, n° 6, 1973.
- [52 ] Santos, A.: "O Método da Lagrangeana Aumentada Aplicado ao Fluxo de Carga Ótimo": Tese de Doutorado apresentada à FEC, Unicamp, Campinas, 1986.
- [53 ] Sayama, H.; Fan, L.T.; Fan, L.S.: "On Lagrange Multipliers and Constraints - II. Augmented Lagrangian Approach": Int. J. Systems Sci., vol.7, n° 11, 1976.
- [54 ] Shanno, D.F.: "Conjugate Gradient Methods with Inexact Searches": Math.Op.Research, vol.3, n° 3, 1973.
- [55 ] Shanno, D.F.: "On the Convergence of a New Conjugate Gradient Algorithm": SIAM J. Numer.Anal., vol.15, n° 6, 1978.
- [56 ] Simonnard, M.: "Linear Programming", Prentice Hall, Englewood Cliffs, 1966.
- [57 ] Soares, S.: "Programação Matemática de Grande Porte - Uma Aplicação a --Sistemas Hidrotérmicos": Tese de Doutorado apresentada à FEC, Unicamp, Campinas, 1978.
- [58 ] Stephanopoulos, G.; Westerberg, A.W.: "The of Hestenes Method of Multipliers to Resolve Dual Gaps in Engineering Systems Optimizations": J.O.T.A., vol.15, n° 3, 1975.
- [59 ] Tamura, H.: "Decentralized Optimization for Distributed - Lag Models of Discrete Systems": Automatica, vol.11, 593-602, 1975.
- [60 ] Tamura, H.: "Multistage Linear Programming for Discrete Optimal Control with Distributed - Lags": Automatica, vol.13, 369-376, 1977.
- [61 ] Tripathi, S.S.; Narendra, K.S.: "Constrained Optimization Problems Using Multiplier Methods": J.O.T.A., vol.9, n° 1, 1972.
- [62 ] Watanabe, N.; Nishimura, Y.; Matsubara, M.: "Decomposition in Large System Optimization Using the Method of Multipliers": J.O.T.A., vol.25, n° 2, 1978.

[63 ] Wollmer, R.D.; "A Substitute Inverse for the Basis of a Staircase Structure Linear Program"; Math. Op. Research, vol.2, n° 3, 1977.