



UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**UM MODELO DE FLUXO EM REDE PARA SOLUÇÃO DE
PROBLEMAS DE DISTRIBUIÇÃO DE PRODUTOS COMPOSTOS**

TESE SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
PRODUÇÃO DA UNIVERSIDADE FEDERAL DE SANTA CATARINA, PARA
OBTENÇÃO DO TÍTULO DE “DOUTORA EM ENGENHARIA DE PRODUÇÃO”

CATIA MARIA DOS SANTOS MACHADO

FLORIANÓPOLIS (SC) – BRASIL, 2005.

TERMO DE APROVAÇÃO

UM MODELO DE FLUXO EM REDE PARA SOLUÇÃO DE PROBLEMAS DE DISTRIBUIÇÃO DE PRODUTOS COMPOSTOS

Tese de Doutorado defendida em 5 de abril de 2005

CATIA MARIA DOS SANTOS MACHADO

BANCA EXAMINADORA

Prof. Sérgio Fernando Mayerle, Dr.
Orientador

Prof. Rafael Carlos Vélez Benito, Dr.
Moderador – UNIVALI/SC

Prof^a. Elizabeth Wegner Karas, Dr^a.
Examinadora Externa –UFPR/PR

Prof. Nei Yoshihiro Soma, Ph.D.
Examinador Externo –ITA/SP

Prof. Antônio Sérgio Coelho, Dr.
Examinador – UFSC/SC

Prof. João Neiva de Figueiredo, Ph.D.
Examinador – UFSC/SC

Prof. Edson Pacheco Paladini
Coordenador

Aos meus filhos, Guilherme e Marina

Mais uma vez...

Perdoem a cara amarrada, perdoem a falta de abraço, perdoem a falta de espaço, perdoem por tantos perigos, perdoem a falta de abrigo, perdoem a falta de ar, perdoem a falta de escolha. Os dias eram assim... E quando brotarem as flores, e quando crescerem as matas, quando colherem os frutos, digam o gosto para mim.

(Ivan Lins)

Ao meu esposo Gilnei, pelo apoio, compreensão e paciência.

AGRADECIMENTOS

Ao meu Orientador *Prof. Dr. Sérgio Fernando Mayerle*, pelas lições de saber, orientação constante, dedicação e paciência no decorrer do trabalho e, também, pelo estímulo, sugestões e comentários oportunizando novos conhecimentos.

Ao Co-orientador *Prof. Dr. Rafael Carlos Vélez Benito*, por ter me conduzido até o *Caminho das Pedras* – o começo para a realização deste trabalho.

Ao *Prof. Dr. Antônio Sérgio Coelho*, pelo aceite no curso e pelo apoio no decorrer do trabalho.

Em especial ao *Prof. M.Sc. Rafael Machado Casali*, pela implementação do algoritmo desenvolvido no trabalho e por me fazer lembrar que vencemos porque não estamos sós, nossa vitória cabe também a outros.

À equipe do *LABTRANS*, pelo apoio computacional prestado.

À *família*, minha *mãe*, e em especial à minha tia *Sila*.

Aos meus colegas de curso, especialmente a *Elaine Corrêa Pereira* e *Junelene Costódio*.

À *Fundação Universidade Federal do Rio Grande*, pela oportunidade que me concedeu em realizar o curso de Pós-Graduação em Engenharia de Produção.

À *Banca Examinadora*, pelos comentários, críticas e sugestões.

E, finalmente, agradeço a *Deus* por ter-me concedido força suficiente para mais esta conquista de minha vida.

RESUMO

Neste trabalho é proposto um modelo linear de Fluxo em Redes para o problema de minimização de custos de produção e distribuição de Múltiplos Produtos Compostos. Neste modelo, restrições de acoplamento são consideradas para tratar a proporcionalidade existente entre os diversos fluxos que formam o produto composto, bem como as restrições de capacidade dos arcos pelos quais estes fluxos percorrem. A metodologia utilizada para solucionar o problema é baseada na estratégia de particionamento da matriz básica, e na implementação de uma especialização do método simplex dual para solucionar o problema particionado primal. Como solução inicial, é utilizada uma base construída por meio de um método heurístico que aloca fluxos em caminhos de custo mínimo. Para realização das operações de troca de base, a matriz ciclo é armazenada na forma produto da inversa, de modo a manter a esparsidade e a dimensão. Testes computacionais, contendo em torno de 200.000 restrições e 370.000 variáveis, aplicados à distribuição de produtos compostos de uma indústria do setor petroquímico, foram realizados com sucesso. Os resultados obtidos demonstram a eficiência computacional do algoritmo desenvolvido e a aplicabilidade do modelo formulado. Finalmente, recomendações são apresentadas para desenvolvimento de trabalhos futuros.

Palavras-chave: múltiplos produtos; particionamento; primal-dual; programação linear; fluxos em redes.

ABSTRACT

This dissertation presents a linear model of flows in networks to solve the cost minimization problem of the production and distribution of compound multi-commodities. Coupling constraints are employed to permit the use of arc capacity upper bounds and to guarantee proportionality among compound multi-commodity network flows. The methodology used to solve the problem is based on partitioning the basic matrix and on implementing a variant of the dual simplex method to solve the partitioned primal problem. The first iteration solution is a base constructed through a heuristic method that allocates flows through least-cost paths. A fixed-dimension and sparse cycle matrix which is stored in its inverse product form is used for base-changing procedures. Test runs of compound multi-commodity distribution problems in the petrochemical sector with 370.000 variables and 200.000 constraints were executed successfully. The results confirmed the computational efficiency of the algorithm as well as the broad applicability of the formulation. Future research opportunities are suggested.

Key-words: Compound multi-commodities; partitioning; primal-dual methods; linear programming; flows in networks.

LISTA DE FIGURAS

Figura III-1 Representação de uma rede através de um grafo G	32
Figura III-2 Representação de um caminho	34
Figura III-3 Árvore com nó raiz	39
Figura III-4 Orientação dos arcos no ciclo formado com o arco fora da base	40
Figura III-5 Base genérica representando o produto K_1 contendo dois arcos complementares	62
Figura III-6 Base genérica representando o produto K_2 contendo um arco complementar	62
Figura III-7 Representação da nova base para o produto K_1 , após pivoteamento sobre o caso 1	63
Figura III-8 Representação da nova base para o produto K_1 , após pivoteamento sobre o caso 2a (entra e sai arcos de um mesmo produto)	64
Figura III-9 Representação da nova base para o produto K_1 , após pivoteamento sobre o caso 2a (entra e sai arco de produtos diferentes)	65
Figura III-10 Representação da nova base para o produto K_2 , após pivoteamento sobre o caso 2a (entra e sai arcos de produtos diferentes)	65
Figura III-11 Representação da nova base para o produto K_1 , após pivoteamento sobre o caso 3	66
Figura III-12 Representação da nova base para o produto K_1 , após pivoteamento sobre o caso 5 (a folga que entra na base está associada ao arco complementar que sai)	68
Figura III-13 Representação da nova base para o produto K_1 , após pivoteamento sobre o caso 5 (a folga que entra na base não está associada ao arco complementar que sai)	69

Figura III-14	Representação da nova base para o produto K_2 , após pivoteamento sobre o caso 5 (a folga que entra na base não está associada ao arco complementar que sai)	69
Figura III-15	Representação da nova base para o produto K_1 , após pivoteamento sobre o caso 6	70
Figura IV-1	Modelo lógico representando múltiplos produtos com misturas em distribuidores ou terminais	76
Figura IV-2	Modelo lógico representando múltiplos produtos compostos em nós consumidores (super nó i)	79
Figura IV-3	Modelo lógico representando um produtor com múltiplos produtos (super nó i)	80
Figura IV-4	Modelo lógico representando o transporte entre dois super nós através de um super arco (i, j)	81
Figura IV-5	Modelo lógico representando a rede completa para um produto (Produto A) incluído a fração correspondente do fluxo dos produtos compostos derivados (mistura Produto A \times Produto B)	83
Figura IV-6	Rede completa para os produtos dos tipos A, B e produtos compostos	88
Figura IV-7	Rede em equilíbrio completa, representando o produto do tipo A, incluído a fração do fluxo do produto composto derivado (mistura adubo A \times adubo B)	89
Figura IV-8	Rede em equilíbrio completa, representando o produto do tipo B, incluído a fração do fluxo do produto composto derivado (mistura adubo A \times adubo B)	89
Figura IV-9	Representação da matriz básica	92
Figura V-1	Orientação dos arcos básicos nos ciclos formados pelos arcos complementares e pelo arco que entra na base	104
Figura V-2	Árvore básica para o produto A dividida em duas sub-árvores	108
Figura V-3	Árvore básica para o produto B	108
Figura VI-1	Valores duais dos nós para o produto 1 sem restrição de agregação	111
Figura VI-2	Valores duais dos nós para o produto 2 sem restrição de agregação	111
Figura VI-3	Valores duais para os nós do produto 2 na segunda iteração dual	113

LISTA DE TABELAS

Tabela IV-1	Especificação dos arcos (Adubo do tipo <i>A</i>)	90
Tabela IV-2	Especificação dos arcos (Adubo do tipo <i>B</i>)	90
Tabela VI-1	Especificação dos arcos para o produto 1, cujo fluxo inicial é a solução ótima obtida no problema sem restrição de capacidade agregada	110
Tabela VI-2	Especificação dos arcos para o produto 2, cujo fluxo inicial é a solução ótima obtida no problema sem restrição de capacidade agregada	110
Tabela VI-3	Tempo de CPU para os dois Estágios do Algoritmo	115

SUMÁRIO

RESUMO	v
ABSTRACT	vi
LISTA DE FIGURAS	vii
LISTA DE TABELAS	x
CAPÍTULO I	13
1. INTRODUÇÃO	13
1.1 Considerações Iniciais	13
1.2 Objetivos	14
1.3 Importância do Trabalho	15
1.4 Motivação	15
1.5 Estrutura do Trabalho	16
CAPÍTULO II	18
2. REVISÃO DA LITERATURA	18
2.1 Visão Geral dos Métodos de Programação Linear	18
2.2 Métodos de Solução para Problemas Lineares de Fluxo em Rede	20
2.3 Aplicações	28
CAPÍTULO III	31
3. FUNDAMENTAÇÃO TEÓRICA	31
3.1 Notação e Convenção	31
3.2 Método Simplex para o Problema de Fluxo em Rede	32
3.2.1 Caracterização de uma Base para o Problema de Fluxo em Rede	37
3.2.2 Método Simplex Primal no Grafo	38
3.3 Método Simplex Primal Particionado para o Problema de Fluxo em Rede com Múltiplos Produtos	41
3.3.1 Formulação Matemática	42
3.3.2 Particionamento da Base	42
3.3.3 Determinação da matriz ciclo Q	46
3.3.4 Determinação das Variáveis Duais e Atualização da Coluna no Algoritmo com Particionamento Primal	47

3.4	Algoritmo com Particionamento Primal para MP	52
3.4.1	Estrutura Geral do Algoritmo	52
3.4.2	Operações de Pivoteamento sobre a matriz Ciclo Q	54
3.4.3	Considerações sobre a Matriz Ciclo Q	55
3.5	Matriz Ciclo Y	56
3.6	Obtenção dos Valores Duais utilizando a Inversa da matriz Ciclo Y	57
3.7	Obtenção dos Valores y 's no Teste de Razão utilizando a Inversa da matriz Ciclo Y	60
3.8	Operações de Pivoteamento sobre a Matriz Ciclo Y	61
3.9	Algumas Considerações	70
CAPÍTULO IV		73
4.	MODELO DE REDE PARA O PROBLEMA DA DISTRIBUIÇÃO DE PRODUTOS COMPOSTOS	73
4.1	Reformulação do Modelo de Fluxo em Rede para Múltiplos Produtos	73
4.2	Formulação Matemática	74
4.2.1	Caracterização de Variáveis de Decisão e de Parâmetros do Modelo	74
4.2.2	Restrições Estruturais para Super Nós de Transbordo e Mistura	76
4.2.3	Restrições Estruturais para Super Nós Consumidores	78
4.2.4	Restrições Estruturais para Super Nós de Produção	80
4.2.5	Restrições Estruturais para Super Arcos de Transporte	81
4.2.6	Função Objetivo	82
4.3	Modelo e Estrutura da Rede	83
4.4	Operacionalização da Busca de uma Base Inicial Viável	93
CAPÍTULO V		94
5.	MÉTODO SIMPLEX PRIMAL-DUAL ESPECIALIZADO PARA O PROBLEMA DE FLUXO EM REDE COM MÚLTIPLOS PRODUTOS	94
5.1	Estratégia de Solução	94
5.1.1	Estágio I	96
5.1.2	Estágio II	97
5.2	Método Simplex com Particionamento Dual	99
5.3	Considerações sobre a Realização das Iterações Duais	106
CAPÍTULO VI		109
6.	RESULTADOS NUMÉRICOS E COMPUTACIONAIS	109
6.1	Realização das iterações Duais sobre uma Rede com Super Nós de Transbordo	109
6.2	Testes Numéricos	115
6.3	Considerações Finais	117
CAPÍTULO VII		118
7.	COMENTÁRIOS, CONCLUSÕES E RECOMENDAÇÕES	118
7.1	Comentários	118
7.2	Conclusões	119
7.3	Recomendações	121
BIBLIOGRAFIA		122
GLOSSÁRIO		128

APÊNDICE	131
I. Rede com super nós de transbordo para ilustrar passo a passo o algoritmo particionado primal que utiliza a matriz ciclo Y	132
II. Rede com distribuição de produtos compostos para ilustrar passo a passo três iterações do algoritmo particionado primal que utiliza a matriz ciclo Y	151
ANEXOS	155

CAPÍTULO I

1. INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

A abordagem por programação linear desfruta de ampla aceitação devido à habilidade de modelar problemas importantes e complexos. Alguns dos mais extraordinários modelos de programação linear são aqueles que possuem uma estrutura especial que permitem o desenvolvimento de algoritmos eficientes e especializados em sua solução. Nessa família de problemas, destaca-se o modelo para o problema de fluxo em rede para múltiplos produtos *MP*. Os problemas *MP* são caracterizados basicamente por vários produtos compartilharem arcos em uma rede capacitada. Esses problemas têm sido extensivamente estudados, devido suas numerosas aplicações e estrutura de rede que exibem. O problema da distribuição de produtos compostos *MPC* pode ser caracterizado como uma extensão do problema *MP* no qual produtos distintos são utilizados para formular novos compostos demandados pelo mercado. A solução deste problema implica na identificação do local onde o composto deve ser formulado bem como as respectivas quantidades envolvidas, de modo a assegurar a proporcionalidade.

O *MPC* surge pela necessidade de estabelecer uma solução eficiente para este problema de produção e distribuição, por parte, principalmente, da indústria petroquímica. Este problema torna-se, ao mesmo tempo, mais complexo e relevante, na medida em que os pontos de produção se distanciam geograficamente dos pontos de demanda e consumo, e o processo de distribuição não é realizado, obrigatoriamente, de um ponto de produção a um ponto de demanda, permitindo que pontos intermediários, como bases de misturas, centros de concentração e distribuição sejam utilizados. Também contribuem para esta complexidade, o fato das interligações possuírem restrições de capacidade e custos variados. Contudo, o

problema *MPC* serve de modelo para inúmeras outras situações absolutamente diversas que lhe são assemelhadas por abstração.

1.2 OBJETIVOS

Este trabalho tem como objetivo geral verificar a viabilidade de implantação de um algoritmo de programação linear para o problema de múltiplos produtos, do tipo *MP* e *MPC*, capaz de resolver problemas de médio e grande porte, utilizando uma estratégia primal-dual sobre o problema particionado, e a estrutura do problema a fim de acelerar o processo de busca da solução ótima.

Para alcançar este objetivo geral, os seguintes objetivos específicos serão considerados no desenvolvimento deste trabalho:

- i) Modelar a rede, caracterizando as entidades de produção, distribuição e consumo através de conjuntos de nós elementares, por onde trafegam os produtos e são formulados os compostos.
- ii) Equacionar um modelo matemático de Programação Linear, de modo a permitir a quantificação dos volumes transportados, produzidos e formulados, ao longo de toda a cadeia logística, desde os pontos de produção até os mercados consumidores.
- iii) Enquadrar o modelo matemático formulado na classe de problemas de fluxo em redes com múltiplos produtos, de modo a caracterizar os fluxos das frações correspondentes aos produtos compostos.
- iv) Desenvolver uma estratégia de solução para o problema de fluxo em redes para múltiplos produtos e compostos, baseado na especialização do algoritmo simplex primal-dual.
- v) Desenvolver uma solução básica inicial avançada, através de técnicas heurísticas, de modo a reduzir o número de iterações e o tempo computacional gastos na resolução de algumas instâncias típicas deste problema.
- vi) Avaliar a viabilidade da estratégia de solução e o desempenho do algoritmo proposto, através da solução de um conjunto problemas reais de uma indústria do setor petroquímico.

1.3 IMPORTÂNCIA DO TRABALHO

A escala de globalização e busca da eficácia organizacional tem forçado cada vez mais o desenvolvimento de métodos quantitativos avançados na tomada de decisões com solução em tempo razoável. Hoje em dia, problemas com milhares de restrições e variáveis são resolvidos em microcomputadores. Nesse contexto, a metodologia desenvolvida possibilita minimizar os custos logísticos envolvidos na distribuição e produção de produtos compostos. A rede lógica permite indicar o local e a quantidade de formulação de produtos compostos, bem como o transporte desses até os mercados consumidores.

O trabalho tem contribuições importantes em relação ao estudo dos problemas que envolvem múltiplos produtos. Além da reformulação do modelo tradicional para o problema *MP* por meio de uma tradução contextual conveniente do problema *MPC*, um estudo profundo da estrutura do algoritmo simplex particionado para múltiplos produtos implicou em resultados matemáticos, que serão apresentados nos capítulos 3 e 5, cujas propriedades levaram às relações de dualidade e sensibilidade que, devidamente aproveitadas, contribuirão para uma eficiência numérica do algoritmo.

1.4 MOTIVAÇÃO

As atividades de logística e transporte de múltiplos produtos têm sido amplamente pesquisadas e os métodos de solução exata oferecem grandes dificuldades computacionais. Entretanto, o método apresentado por Castro e Nabona [Cas96] para solucionar problemas de múltiplos produtos de grande porte é exato e eficiente. Trata-se do método simplex particionado especializado para o caso de redes com múltiplos produtos, cuja técnica de solução é realizada em duas etapas. A possibilidade de aplicação da técnica de particionamento ao problema *MPC* surge quando um novo modelo de rede é desenhado.

Muitas são as aplicações práticas que fazem uso desse modelo e o problema da distribuição de produtos compostos pode ser visto como uma extensão do problema *MP*.

Exemplos da distribuição de produtos compostos aparecem na indústria de fertilizantes, celulose, petrolífera, petroquímica, siderúrgica e outras.

1.5 ESTRUTURA DO TRABALHO

O presente trabalho está dividido em sete capítulos, sendo o primeiro capítulo introdutório.

O Capítulo II apresenta uma revisão histórica sobre os métodos que solucionam os problemas de programação linear, incluindo o caso especial dos problemas de fluxo em redes com múltiplos produtos.

O Capítulo III apresenta a fundamentação teórica necessária ao desenvolvimento deste trabalho, resgatando os principais resultados apresentados em [Ken80] em relação à especialização do algoritmo simplex aplicado aos problemas de fluxo em rede com custo mínimo. Ainda neste capítulo, algumas considerações são realizadas sobre a técnica de resolução proposta originalmente por Kennington e Helgason [Ken80], e também por Castro e Nabona [Cas96]. Em face destas considerações são sugeridas algumas modificações em relação ao tratamento da matriz ciclo, que passa a ter dimensão fixa, contrariamente ao que é feito pelos autores referenciados. Dado as modificações sugeridas, são acrescentadas, a este capítulo de fundamentação teórica, as extensões e adaptações pertinentes.

No Capítulo IV, é apresentada a descrição do problema de distribuição de produtos compostos, sendo formalizado, para o mesmo, um modelo de programação linear. Tal modelo apresenta a estrutura de um problema de fluxo em redes com múltiplos produtos, no qual cada entidade envolvida na logística de distribuição, caracterizada por super-nós, é representada por um conjunto de nós elementares sobre os quais se aplicam as restrições de conservação de fluxo. Aos arcos, que conectam estes nós elementares, representando os fluxos de produtos, incluindo os compostos, são associadas restrições de capacidade. Restrições adicionais de acoplamento entre as diversas correntes de produtos são agregadas ao modelo para restringir as capacidades dos arcos que são compartilhados por diversos produtos, bem como para manter a proporcionalidade entre os produtos que formam o composto.

No Capítulo V é apresentada uma estratégia de solução para o modelo desenvolvido no capítulo IV. Esta estratégia é composta de dois estágios. No primeiro, uma alocação de fluxo inicial é realizada através de uma técnica heurística primal. No segundo estágio, partindo da solução inicial obtida, é aplicado o método simplex dual especializado para o problema de fluxo em rede com particionamento primal. Ainda neste capítulo, são apresentados os algoritmos referentes ao desenvolvimento desta estratégia, bem como a descrição detalhada da implementação dos passos algorítmicos, para o aproveitamento eficiente da estrutura de rede do problema.

O Capítulo VI apresenta resultados numéricos e computacionais. Um exemplo numérico ilustra passo a passo o algoritmo com particionamento dual, correspondente ao segundo estágio do algoritmo apresentado no capítulo V. Além disso, resultados computacionais para problemas reais de uma empresa do setor petroquímico, tais como dimensões de problemas resolvidos, tempos computacionais e número de iterações realizadas, são apresentados.

Finalmente, no Capítulo VII são apresentadas algumas considerações, conclusões e recomendações para trabalhos futuros.

O trabalho ainda contém um apêndice ilustrando passo a passo o algoritmo com particionamento primal para o problema MP e MPC , onde efetivamente são aplicados todos os resultados algébricos obtidos.

CAPÍTULO II

2. REVISÃO DA LITERATURA

Este capítulo apresenta uma breve retrospectiva sobre alguns métodos de soluções para problemas lineares de otimização. Técnicas têm sido aplicadas a problemas de fluxo em rede solucionando problemas lineares e não lineares com o objetivo de investigação e compreensão do desempenho computacional na implementação aos problemas de custo mínimo. Uma razão importante para estudar modelos de fluxo em rede para múltiplos produtos é a grande variedade de aplicações que podem ser modeladas.

2.1 VISÃO GERAL DOS MÉTODOS DE PROGRAMAÇÃO LINEAR

O primeiro algoritmo para resolver o Problema de Programação Linear (*PPL*) foi o Simplex desenvolvido pelo matemático americano George Dantzig em 1947, (ver [Dan63]). Após vários anos de aprimoramento tornou-se muito eficiente, apesar de pesquisar a solução ótima percorrendo de vértice em vértice. De 1950 até 1965 foram desenvolvidos algoritmos específicos para modelos lineares de fluxo em rede. A estrutura especial do modelo permitia a utilização de métodos específicos muito eficientes a esses casos, não se aplicando a um caso geral. Nessa época, foram desenvolvidos o método primal simplex especializado e o método primal-dual. A especialização do primal simplex começou com Dantzig e culminou com o trabalho de Ellis Johnson [Joh65], baseado em [Dan63] e [Cha61]. O método primal-dual teve origem com Harold Kuhn's (algoritmo húngaro) para o problema de atribuição e culminou com Delbert Fulkerson's (algoritmo out-of-kilter [Ful71]).

No final da década de 1970, Khachian [Kha79] apresentou o algoritmo dos Elipsóides para programação linear, com tempo polinomial. Embora o algoritmo tivesse

propriedades teóricas satisfatórias, o seu desempenho na prática era muito inferior quando comparado com o método simplex.

O algoritmo Project, desenvolvido por Rödder e Blauth [Röd80], utilizando a idéia de procurar o ótimo na direção do gradiente da função objetivo projetado sobre o espaço nulo da matriz de restrições, mostrou uma boa eficiência numérica a modelos de programação linear que apresentavam estruturas quaisquer.

Rödder [Röd82a] demonstra que o algoritmo é finito, através do uso de um conceito análogo da perturbação, também utilizado na demonstração da convergência do simplex. Rödder et al. [Röd82b] sugerem um novo algoritmo conceitual, que utiliza a forma produto da inversa para efetuar mudanças na base, a ser desenvolvido a partir do algoritmo revisado.

Mais tarde, Coelho [Coe83] desenvolve uma teoria permitindo uma Análise de Pós-Otimilidade de um problema resolvido pelo algoritmo Project com bastante sucesso. Cria uma teoria e implementação algorítmica para os principais parâmetros do problema de programação linear.

Mayerle [May84] implanta e avalia um algoritmo de programação linear capaz de resolver problemas de médio e grande porte, utilizando o gradiente da função objetivo para efetuar trocas múltiplas de base. A complexidade do algoritmo é analisada fazendo um paralelo com o simplex.

Karmarkar [Kar84] publicou um *método de pontos interiores*, capaz de encontrar uma solução ótima em tempo polinomial, para o pior caso. Além disso, o autor afirmou que seu método era mais rápido do que o método simplex, o que estimulou uma imensa pesquisa na área de otimização linear, incluindo-se questões desafiadoras sobre a eficiência computacional de métodos do tipo simplex. Sousa (ver [Sou00]) cita em seu trabalho sobre uma observação interessante de Dantzig e Thapa (1997) de que, nos primórdios da otimização linear, Von Neumann propôs um método que hoje seria chamado de pontos interiores, porém as implementações feitas na época deram grande vantagem computacional ao método simplex, inibindo as pesquisas nessa direção.

Sousa, baseado no trabalho de Freund e Mizuno [Fre96], apresenta um panorama dos métodos de pontos interiores no campo da otimização linear. Métodos dos pontos interiores têm mudado permanentemente o panorama da teoria, da prática e da computação em programação matemática. Programação linear não é mais sinônimo do clássico método simplex, e muitos pesquisadores tendem a ver agora a programação linear mais como um caso especial da programação não linear devido ao desenvolvimento desses métodos.

Até os meados da década de 1990, métodos de pontos interiores atraíram alguns dos melhores pesquisadores em pesquisa operacional, matemática aplicada e ciência da computação. Muitos artigos já foram escritos, seguindo o trabalho de [Kar84].

Os métodos de pontos interiores utilizam ferramentas matemáticas mais avançadas quando comparadas com métodos do tipo simplex, sua análise matemática é tipicamente complicada e os métodos são menos receptivos a intuição geométrica. Na opinião de alguns autores, a maioria dos métodos de pontos interiores está incluída numa das três principais categorias: Métodos A fim-Escala, Métodos de Redução Potencial e Métodos da Trajetória Central. Os resumos dessas três categorias são apresentados em [Sou00].

O algoritmo original de Karmakar [Kar84] utiliza uma forma muito específica de Problema de Redução Potencial e também o procedimento de transformações projetivas no algoritmo e na prova de garantia do desempenho do algoritmo [Gon92]. Transformação projetiva é conteúdo de suprema importância, pelo menos conceitualmente, na compreensão de métodos de pontos interiores em geral. Embora o Método de Redução Potencial apresente desempenho de complexidade polinomial, não recebe muita atenção em termos de testes computacionais. Métodos de pontos interiores baseados em Trajetória Central são sem dúvida os mais utilizados na prática.

Na tentativa de tirar o maior proveito desses trabalhos, implementações eficientes dessas técnicas foram sendo desenvolvidas e toda tecnologia foi estendida aos problemas lineares de fluxo em rede para múltiplos produtos.

2.2 MÉTODOS DE SOLUÇÃO PARA PROBLEMAS LINEARES DE FLUXO EM REDE

O excepcional trabalho de Bradley et al. [Bra77] enriquece a literatura que envolve problemas lineares de fluxo em rede através de uma completa descrição dos modelos e implementação aos problemas de rede de custo mínimo (simplex primal). É utilizada uma família de programas de rede muito rápidos e eficientes na resolução dos problemas de minimização em grande escala. O mais generalizado dos modelos de fluxo em rede de custo mínimo é a classe de problemas de transbordo capacitado, que incluem problemas de transporte capacitados e não capacitados e o problema de designação clássico. Estes modelos são utilizados em um número variado de aplicações para determinar como (ou em que nível) um bom fluxo deve circular através de uma rede de forma a minimizar custos totais de

embarque. A estrutura matemática unificada de programação linear para redes é apresentada com ênfase especial na estrutura de dados a qual parece ser fundamental na programação matemática geral. Uma revisão pertinente da literatura de otimização acompanha os testes computacionais. Relatos importantes como a exploração da estrutura especial do problema, técnicas de geração de elementos, análise de pós otimalidade, operação com geradores de problemas, arquivos de problema externo e um procedimento de seleção do pivô simples, não cíclico, garantindo um limite para o algoritmo são apresentados.

O modelo de rede para múltiplos produtos difere dos modelos anteriores por conter restrições adicionais. A dificuldade encontrada para solucionar os problemas de múltiplos produtos era, e ainda é, em relação à restrição adicional e o tamanho do problema com respeito a um número considerável de produtos. Esse tipo de problema tem recebido muita atenção nos últimos anos e muitos algoritmos que apresentam a estrutura do problema são sugeridos e implementados. Duas técnicas básicas são aplicadas aos problemas de múltiplos produtos; decomposição e particionamento. As técnicas de decomposição são caracterizadas como *price-directive* e *resource-directive*.

A decomposição *price-directive*, também chamada Dantzig-Wolfe *Decomposition*, é um procedimento em que existe uma coordenação entre o problema mestre e cada um dos vários subproblemas mudando a função objetivo (preços) dos subprogramas a cada iteração. O objetivo é obter um conjunto de preços (variáveis duais) tais que a solução combinada com todos os subproblemas assegura o ótimo para o problema global.

A decomposição *resource-directive* é um procedimento que distribui a capacidade do arco entre os k produtos de modo a resolver k subprogramas assegurando o fluxo ótimo para o problema de acoplamento. A cada iteração, uma alocação é feita e k problemas de monoproduto são resolvidos. Desde muito cedo referências apresentam esta aproximação, surgida em 1956 (Robacker citado em [Ken77]).

As técnicas de particionamento são especializações do método simplex, onde a base corrente é particionada em uma estrutura especial. Essas técnicas são especializações do método simplex primal, dual, ou primal-dual. Os trabalhos de [Sai66], [Har72], [Gra73] e [Ken75] apresentam a técnica primal, enquanto [Gri72] apresenta a técnica dual.

Os métodos *price-directive* e *primal partitioning* são baseados no algoritmo simplex e, portanto existe um critério de otimalidade. Esses dois métodos são vistos propriamente como técnicas exatas. Em relação ao método *resource-directive*, como não existe efetivamente um critério de otimalidade, o método é visto como uma heurística.

Muitas vezes, sob o ponto de vista computacional, os métodos exatos são considerados muito dispendiosos e alternativas mais econômicas têm sido procuradas. Deste esforço, uma variedade de heurísticas de alta eficiência prática tem surgido.

De qualquer modo, é presumível que, no universo dos possíveis problemas, existam aqueles que apresentam um comportamento melhor quando utilizam um método de resolução exato, enquanto que em outros a heurística é utilizada com maior sucesso.

Os métodos que surgem em anos mais recentes são baseados tanto em métodos heurísticos como exatos, e abordam os mais variados assuntos.

Kennington e Shalaby [Ken77] apresentam uma técnica heurística para obter boas soluções para problemas grandes de fluxo em rede de múltiplos produtos. A abordagem geral é alocar as capacidades dos arcos entre produtos individuais e, portanto, decompor o problema em um conjunto de problemas de monoproduto. Os problemas de monoproduto são resolvidos e a solução combinada é comparada com um limite inferior. Se a solução estiver dentro de um desvio aceitável do limite inferior, o procedimento acaba. Em caso contrário, as capacidades dos arcos são realocadas e os subprogramas são resolvidos. A realocação é baseada em uma abordagem de otimização pelo subgradiente. Portanto, a técnica heurística não envolve operações matriciais e os requisitos de armazenagem são modestos, quase todas as operações são efetuadas diretamente nas redes de monoproduto.

Chen e Chin [Che92], pesquisadores de problemas de múltiplos produtos de custo mínimo, afirmam que existem dois tipos de valores associados a um arco. O primeiro é a capacidade do arco e o segundo é o custo do fluxo ao longo do arco. O artigo adiciona um terceiro atributo que é o grau de dificuldade em transpor o arco dentro de um modelo convencional de múltiplos produtos. Dessa forma, o novo problema que é mais geral e difícil que o problema convencional de minimização de custos é apresentado. Entretanto, cada um dos problemas pode ser convertido no outro através de um algoritmo polinomial. Com a equivalência, a técnica desenvolvida para resolver o problema convencional de múltiplos produtos de custo mínimo pode ser utilizada para resolver o novo problema mais geral.

Ross, Kimberly e Lynn [Ros93] investigam o impacto da formulação dos problemas de fluxo em rede para múltiplos produtos que se aproveitam do aspecto combinatorial da decomposição Dantzig-Wolfe. O algoritmo implementado é testado em um conjunto de problemas de múltiplos produtos de custo mínimo baseados em caminhos e em árvores, encontrando a melhor performance do algoritmo aos problemas baseados em caminhos. Resultados computacionais são apresentados para quatro formulações diferentes de problemas de múltiplos produtos a custo mínimo. Os resultados demonstram que para problemas de

natureza distinta, cuja estrutura subjacente é uma rede dinâmica, a formulação baseada em caminhos é superior à baseada em árvores. O algoritmo baseado em caminhos é comparado com (OB1 e CPLEX), onde é verificado que para muitos problemas de grande escala a abordagem pela decomposição Dantzig-Wolfe apresenta boas soluções. Também é implementado e testado um algoritmo híbrido (decomposição de ponto interior) para um conjunto de problemas de múltiplos produtos. Resultados indicam que o algoritmo de decomposição Dantzig-Wolfe é muito superior quando comparado com o algoritmo híbrido.

Castro e Nabona [Cas96] apresentam um código para resolver problemas de fluxo em rede para múltiplos produtos com função objetivo linear e não linear, considerando restrições acopladas que unem arcos de um mesmo ou diferentes produtos. Na resolução dos problemas de múltiplos produtos é utilizado o particionamento primal e vários testes são realizados em problemas reais e gerados aleatoriamente, contendo acima de 150.000 variáveis e 45.000 restrições. Os resultados obtidos são comparados com metodologias alternativas como: método de pontos interiores, métodos especializados para fluxo em rede e pacotes que otimizam problemas com restrições lineares e não lineares.

Yamakawa, Matsubara e Fukushima [Yam96] apresentam uma implementação do método primal-dual de pontos interiores para problemas de múltiplos produtos com custos quadráticos separáveis. A estrutura de bloco é aproveitada de forma que os sistemas lineares de equações são resolvidos a cada iteração sendo decompostos em sistemas independentes de equações lineares. Um sistema está associado a cada produto e o outro corresponde às restrições de acoplamento. Os sistemas são resolvidos pela aplicação do método do gradiente conjugado pré-condicionado. Resultados computacionais indicam que a abordagem proposta é eficiente.

Gondzio, Sarkissian e Vial [Gon97] tratam de questões que surgem quando o método de pontos interiores é utilizado para lidar com o problema mestre em uma abordagem de decomposição. Os pontos principais se utilizam eficientemente da estrutura especial do problema mestre para reduzir os custos em uma iteração de ponto interior. O método pode ser utilizado em casos generalizados, em qualquer abordagem de decomposição, toda vez que o problema mestre seja resolvido pelo método de pontos interiores. Resultados computacionais indicam vantagens por uma abordagem particular de decomposição. O método *Analytic Center Cutting Plane (ACCPM)* é aplicado para resolver problemas de fluxo em rede de múltiplos produtos não lineares de grande escala, contendo em torno de 5000 arcos e 10.000 produtos.

Liu [Liu97] motivado por [For92], apresenta um algoritmo baseado no *dual steepest-edge* de duas fases, para resolver problemas de múltiplos produtos em uma rede capacitada. Na primeira fase uma solução inicial avançada, de acordo com um método dual *steepest-edge*, é aplicada para resolver cada problema de rede de monoproduto capacitado. A cada iteração a inviabilidade primal é diminuída ou a função objetivo dual é aumentada. Na segunda fase, o critério de seleção do *steepest-edge* é utilizado para determinar a restrição mais inviável. Ao manter a viabilidade dual enquanto melhora-se a função objetivo, o número de restrições inviáveis de acoplamento é monotonicamente reduzido a zero. A propriedade de convergência finita deste algoritmo é mostrada. O algoritmo é codificado em linguagem pascal e testado em vários problemas de pequeno porte.

De acordo com Larsson e Liu [Lar97], problemas lineares geralmente não são resolvidos por relaxação lagrangeana e pelo método do subgradiente, dado que as soluções dos problemas relaxados são normalmente inviáveis nos programas lineares. É mostrado que a solução ótima para o problema linear pode ser obtida pelo simples cálculo médio das soluções dos problemas relaxados, os quais são resolvidos durante a pesquisa do subgradiente. O algoritmo foi implementado para resolver problemas de transporte de múltiplos produtos e mostrou boa performance. Os resultados computacionais indicaram ser possível utilizar relaxação lagrangeana e otimização do problema dual pelo subgradiente construindo eficientemente um método de solução baseado numa proposta especial para estrutura de problemas lineares de grande escala.

Lin e Lin [Lin97] propõem um método para resolver problemas não lineares de múltiplos produtos com função objetivo convexa. O método combina o método de Jacobi projetado e um dual *pseudo-quasi-Newton* projetado, que envolve subproblemas quadráticos de fluxo induzido dentro do método de Jacobi projetado. O método dual pseudo Newton difere muito a partir do método convencional Newton. O método aproveita completamente as vantagens das restrições lineares da rede para obter, constantemente, aproximações da matriz Hessiana esparsa de estrutura desacoplada; inclui uma nova iteração finita de projeções sucessivas e aplica o algoritmo (truncado) para resolver a dificuldade causada pelas restrições de capacidade agregada. É mostrada a convergência do método, incluindo a convergência finita das iterações de projeções sucessivas e da aplicação do algoritmo (truncado). O método quando comparado com Frank-Wolfe, com o algoritmo PARTAN e o método da decomposição *price-directive*, na resolução dos problemas lineares de fluxo em rede, mostrou uma significativa eficiência computacional em termos de tempo de CPU em uma estação de

trabalho que soluciona numerosos exemplos não lineares de múltiplos produtos de fluxo em redes.

McBride [Mcb98] relata que muitos problemas como: logística, transporte e comunicação são formulados como problemas de múltiplos produtos de grande escala e poucos pesquisadores já haviam obtido avanços na resolução de problemas desse tipo. Este aperfeiçoamento pode ser verificado para ambos, algoritmo e hardware. O método simplex apresentado utiliza uma abordagem de particionamento da matriz básica. O tempo computacional na obtenção da solução mostrou ser excelente mesmo em um modesto hardware e os resultados indicaram que problemas de múltiplos produtos de grande escala podem ser resolvidos eficientemente. O autor cita, ainda, que profissionais utilizando o EMNET, um algoritmo com particionamento primal da base, solucionaram problemas logísticos contendo 600.000 restrições e 7.000.000 de variáveis.

Schneur e Orlin [Sch98] apresentam um algoritmo baseado em penalidades que também soluciona o problema de fluxo em rede de múltiplos produtos. A base do algoritmo é simples e consiste em, iterativamente, descobrir e enviar fluxo ao redor de ciclos de custos negativos. Dois parâmetros controlam o algoritmo: o parâmetro de penalidades e o parâmetro de escalas. Na fase escala epsilon, onde epsilon é uma função dos parâmetros, o algoritmo determina o ϵ -ótimo; a solução nas quais as condições para as folgas complementares são satisfeitas dentro de um epsilon. A performance do algoritmo é analisada a partir da teoria e perspectivas práticas. Os resultados práticos suportam a teoria para o algoritmo. Eles mostram também a eficiência do algoritmo para solucionar exemplos de problemas de diferentes estruturas e tamanhos.

Gabrel, Knippel e Minoux [Gab99] descrevem um procedimento de solução exata, baseado na utilização de um software padrão de programação linear e na otimização de problemas de múltiplos produtos. A classe de problemas inclui monoprodutos e múltiplos produtos em redes capacitadas. O procedimento proposto é uma especialização do conhecido procedimento particionado de BENDERS, resolvendo iterativamente um programa linear inteiro 0-1 relaxado. A cada passo do procedimento, novas restrições são incluídas dentro do problema corrente diminuindo significativamente o número de iterações.

Em Castro [Cas00], o algoritmo de eliminação de ciclos é um dos primeiros algoritmos para resolver problemas de múltiplos produtos. Este algoritmo mantém uma solução viável na rede e procede aumentando o fluxo ao longo dos ciclos negativos e conseqüentemente pela eliminação de tais ciclos. O algoritmo de eliminação de ciclos não

possui tempo polinomial, no entanto várias implementações específicas têm comportamento polinomial em relação à escolha dos ciclos a serem eliminados.

Castro e Frangioni [Cas01] apresentam uma implementação do algoritmo de pontos interiores para os problemas de múltiplos produtos. Nesta implementação do algoritmo, os sistemas definidos positivos a cada iteração são resolvidos continuamente num esquema que combina uma fatoração e um método pré-condicionado do gradiente conjugado. Um conjunto de problemas é testado contendo em torno de 2.5 milhões de variáveis e 600.000 restrições. Os resultados mostraram que a abordagem é especialmente competitiva em problemas difíceis de fluxo em rede para múltiplos produtos.

McBride e Mamer [McB01] encontram uma solução para problemas de programação de transportes relacionados com redes de comunicação, computação, ferroviárias, entre outras. Esses problemas são formulados como um problema de otimização linear por partes, no qual a função objetivo é convexa. O particionamento primal da base no algoritmo simplex é modificado eficientemente para resolver problemas utilizando um procedimento de *pricing*, baseados em caminhos mais curtos. As soluções não são somente obtidas rapidamente, como também, o número de pivoteamentos utilizados é uma fração das iterações necessárias para o método simplex padrão.

Detlefsen e Wallace [Del02] discutem uma abordagem do algoritmo simplex, onde somente é necessária a inversão de uma matriz básica de trabalho com dimensão menor ou igual ao número de arcos da rede, e independe do número de produtos. O custo reduzido de uma variável não básica é encontrado utilizando somente a matriz básica de trabalho que é invertida apenas em algumas iterações simplex. Também é discutido como é realizado o pivoteamento na matriz básica de trabalho. O principal detalhe está na interpretação da estrutura da rede em termos dos ciclos que surgem nos produtos individuais.

Cappanera e Frangioni [Cap03] estudam um algoritmo eficiente baseado na decomposição dos custos para solução de problemas de fluxo de custo mínimo de múltiplos produtos. O código utiliza o paralelismo natural próprio na abordagem de decomposição dos custos, resolve os subproblemas de fluxo de custo mínimo em paralelo, obtendo eficiências satisfatórias, com o uso de muitos processadores, em problemas difíceis e com muitos produtos. Esta é exatamente a classe de exemplos onde a abordagem de decomposição alcança seus melhores resultados. O código paralelo desenvolvido é altamente portátil e flexível, podendo ser utilizado em diferentes máquinas.

Castro [Cas03] relata que devido ao avanço no desenvolvimento de solucionadores de programação linear, alguns dos problemas de múltiplos produtos, antigamente

considerados difíceis, podem hoje ser solucionados em poucos minutos, até mesmo mais rápido do que com métodos especializados. Entretanto, para outros exemplos de múltiplos produtos, os solucionadores lineares comuns podem ainda ser bastante ineficientes. O artigo apresenta uma revisão do estado da arte na resolução de problemas de múltiplos produtos de grande escala, comparando um algoritmo especializado que utiliza pontos interiores com CPLEX 6.5, resolvendo problemas difíceis com até um milhão de variáveis e 300.000 restrições.

Holmberg e Yuan [Hol03] caracterizam o modelo de fluxo em rede para múltiplos produtos relacionando-o ao direcionamento de um número de produtos através de uma rede capacitada com custo mínimo. No modelo básico presume-se que para cada produto o fluxo pode ser direcionado a qualquer caminho conectando sua origem a seu destino. Em aplicações para telecomunicações, onde um produto é representado por pares de comunicação, freqüentemente existem atrasos (tempos adicionais) ou requisitos de confiabilidade em caminhos que sejam utilizados para direcionamento. Estes requisitos podem variar por pares de comunicação, representados por classe de prioridades diferentes. O estudo também é estendido a modelos básicos de fluxo em rede de múltiplos produtos incluindo restrições agregadas em caminhos. O problema estendido *NP* é o problema de menor caminho restrito a um caso especial. Para resolver o problema estendido é utilizada uma abordagem por geração de colunas na qual a solução é construída sucessivamente por geração de caminhos. As restrições de agregação são eficientemente manipuladas no subproblema de geração de caminhos. Também, são discutidas várias melhorias nesta abordagem, e resultados computacionais mostram que a abordagem de geração de colunas fornece uma maneira eficiente de resolver o problema estendido, até mesmo para redes muito grandes.

Gabrel, Knippel e Minoux [Gab03] apresentam um algoritmo de solução aproximada para custos discretos na otimização de problemas de múltiplos produtos. Uma extensão da classe de heurísticas gulosas é apresentada em detalhes, e um novo algoritmo de solução aproximada, que basicamente consiste na implementação heurística do método de geração de plano de corte tipo BENDERS, é proposto. Os algoritmos são comparados amplamente com grafos gerados aleatoriamente contendo 50 nós e 90 arcos.

Larsson e Dy [Lar04] ressaltam que o problema linear de fluxo em rede para múltiplos produtos tem muitas aplicações nas áreas de transporte e telecomunicações e, portanto, tem recebido muita atenção. Muitos algoritmos que exploram a estrutura do problema foram sugeridos e implementados. A dificuldade prática de resolver modelos de múltiplos produtos a custo mínimo aumenta rapidamente com relação ao tamanho do

problema especialmente com relação ao número de produtos. Aplicações em telecomunicações levam, tipicamente, a problemas com um grande número de produtos, e utilizar tais exemplos é computacionalmente desafiador. O artigo descreve a avaliação de um procedimento de limite inferior rápido e convergente, baseado em uma reformulação do problema lagrangeano aumentado, isto é, combinando a relaxação lagrangeana com a abordagem de penalidades. O algoritmo é especialmente projetado para solucionar exemplos de grande escala e é comparado com uma abordagem de relaxamento lagrangeano padrão, tendo características de convergência mais favoráveis. Para resolver o subproblema lagrangeano não linear aumentado é aplicado um esquema de decomposição simples que explora a estrutura do subproblema e tem boa capacidade de reotimização. O algoritmo lagrangeano aumentado também pode ser utilizado para suprir heurísticas limitadas superiormente. Em particular ele fornece soluções quase ótimas para exemplos acima de 14.000 arcos e 80.000 produtos dentro de um tempo computacional razoável.

2.3 APLICAÇÕES

Barnhart, Hane e Vance [Ban97] apresentam uma abordagem do modelo de geração de colunas para solucionar problemas inteiros de fluxo de múltiplos produtos. O método utilizado é o *branch-and-bound*, dentro de limites fornecidos por programas lineares para cada nó no caminho origem-destino. Desde que o modelo contenha uma variável para cada caminho origem-destino, para todos os produtos, o problema linear relaxado é resolvido utilizando geração de colunas, isto é, com base nas variáveis duais, novas colunas são geradas de modo a fornecer uma solução melhorada do problema linear. Métodos de solução avançados para problemas lineares são também apresentados. Novas regras de ramificação são obtidas permitindo gerar colunas eficientemente. Computacionalmente, é apresentado um conjunto de problemas testes aplicados ao transporte.

Mesquita e Paixão [Mes99] comparam um programa linear relaxado com uma formulação matemática diferente para problemas de escala de veículos com múltiplos depósitos. Como resultado desta análise teórica, optam por desenvolver um procedimento de pesquisa em árvores baseado numa formulação de fluxo em rede para múltiplos produtos envolvendo dois tipos diferentes de variáveis de decisão: um tipo é utilizado para descrever as conexões entre as viagens, definindo um conjunto de viagens ordenadas, enquanto que o outro tipo é utilizado para definir a retirada de veículos dos depósitos. É também desenvolvido um

algoritmo *branch-and-bound* baseado em relaxação linear e na formulação de fluxo em rede para múltiplos produtos. Experiências computacionais são apresentadas para comparar os dois algoritmos.

Mamer e McBride [Mam00] propõem um novo procedimento de operação de *pricing* para solucionar problemas com estrutura linear de grande escala. O procedimento resolve um subproblema relaxado identificando uma coluna em potencial que entra na base. A eficiência do procedimento consiste na redução do número de pivoteamentos necessários para resolver o problema. A abordagem é motivada pela geração de colunas do método de decomposição Dantzig-Wolfe. O procedimento é testado em dois conjuntos de problemas de múltiplos produtos, um associado a redes de telecomunicações e o outro a problemas logísticos.

M. Hadjiat et al. [Had00] estudam o problema de custo mínimo para problemas de múltiplos produtos onde a demanda de tráfego é satisfeita através de rotas que são limitadas por um dado número de arcos. Tem uma proposta baseada no simplex utilizando um particionamento primal refinado baseado em [Far93]. O programa foi testado com exemplos reais fornecidos por uma operadora de telecomunicações nacional francesa France Telecom e também em problemas de fluxo de múltiplos produtos gerados aleatoriamente. O algoritmo testado é capaz de resolver problemas contendo em torno de 150 nós, 1500 arcos e 11.000 demandas. A segunda parte do artigo generaliza as técnicas descritas previamente para lidar com restrições adicionais relativa à confiabilidade do sistema. Operadores das maiores companhias de telecomunicações buscam projetar redes confiáveis de múltiplos produtos, isto é, que são capazes de redirecionar todas as demandas de tráfego, mesmo que algum nó ou arco esteja danificado. É apresentado um modelo simples do problema baseado em fluxos de múltiplos produtos não simultâneos.

Chardaire e Lisser [Cha02] abordam modelos de fluxo em rede para múltiplos produtos e sugerem uma grande variedade de contextos típicos no dimensionamento de redes. Várias são as abordagens baseadas na especialização do algoritmo simplex e pontos interiores para resolver os problemas de múltiplos produtos não orientados. Algoritmos são testados com dados da rede de transmissão distrital da Telecom Paris França. Algoritmos simplex primais e duais aplicados na estrutura particular da matriz de restrições são apresentados e comparados. Resultados numéricos são fornecidos por problemas de até 800.000 restrições e 6.000.000 variáveis. Entretanto, uma abordagem muito mais potente, baseada em métodos de decomposição especializada pode ser implementada para resolver o problema. Um método de decomposição Dantzig-Wolfe projetado é comparado com a implementação especializada do

método *Analytic Center Cutting Plane (ACCPM)*. Técnicas de partição são utilizadas para aproveitar a estrutura dos programas mestres envolvidos nestes métodos.

Moz e Pato [Mos04] apresentam um novo modelo de fluxo em rede para múltiplos produtos, associado ao problema de reprogramação da escala de enfermeiras em hospitais, quando no mínimo uma enfermeira informa que não poderá trabalhar nos turnos designados para ela em um ou mais dias de trabalho. Como resultado, a escala atual tem que ser refeita de acordo com as regras contratuais de trabalho e requisitos institucionais. Todas essas restrições são consideradas como rígidas. Entretanto, maiores alterações, nas escalas de enfermeiras previamente designadas, devem ser evitadas. Este artigo é baseado em um estudo de caso de um hospital público em Portugal. Ele apresenta duas novas formulações de fluxo inteiro para múltiplos produtos no problema de reprogramação, além do experimento computacional realizado utilizando dados reais. O primeiro modelo é baseado em uma rede acíclica multi-direcionada. A agregação dos nós da rede leva a um segundo modelo. Os resultados obtidos mostram que a segunda formulação de fluxo inteiro para múltiplos produtos tem um desempenho melhor que a primeira, tanto em termos de qualidade de solução, como em tempo computacional.

Como pode ser observado, nessa breve revisão, o problema de otimização linear é, sem dúvida, freqüentemente utilizado seja no planejamento de produção, na alocação de recursos, bem como em muitos outros aspectos da tomada de decisão em indústrias, economia, transporte, engenharia, etc. Um resumo sobre os problemas de programação linear pode ser encontrado em [Len91].

Cabe salientar que no estudo particular dos problemas lineares de fluxo em rede para múltiplos produtos, e alguns de suas variantes, como observado através dos artigos revisados, a ênfase dada ainda consiste na importância da investigação na busca de soluções para suas aplicações e na compreensão do seu desempenho computacional. Hoje em dia, a necessidade do mercado por meios que permitam a movimentação dos produtos de forma eficiente, reduzindo perdas, tempo e custos logísticos aumenta a pesquisa para modelagem de problemas de fluxo de múltiplos produtos. Dessa forma, o trabalho apresentado permite uma contribuição a mais nesse contexto, no que diz respeito ao transporte de múltiplos produtos compostos.

CAPÍTULO III

3. FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo resgatar os resultados apresentados em [Ken80] em relação à especialização do algoritmo simplex aplicado aos problemas de fluxo em rede com custo mínimo. Resultados algébricos são mostrados, obtendo as fórmulas necessárias que determinam as variáveis duais e as colunas atualizadas no algoritmo simplex com particionamento primal para problemas de múltiplos produtos. Uma contribuição deste trabalho é a forma de aplicação do algoritmo simplex especializado para o problema *MP* em função da inversa de uma matriz de dimensão fixa denominada matriz ciclo *Y*.

3.1 NOTAÇÃO E CONVENÇÃO

As notações e convenções utilizadas no decorrer do trabalho são as mesmas adotadas por [Ken80]. Uma notação especial é utilizada para representar o produto de uma matriz por um vetor ou vice versa. Adota-se que um vetor é um vetor linha (matriz de dimensão $1 \times n$) quando pré-multiplica uma matriz e um vetor coluna (matriz de dimensão $n \times 1$) quando pós-multiplica uma matriz. O produto interno de dois vetores a e b é simplesmente denotado por ab , isto é, o produto de um vetor linha por um vetor coluna. No contexto, de acordo com a conveniência, um vetor pode ser tratado como vetor linha ou vetor coluna. Sigma é um escalar utilizado para denotar o sinal de uma função, definida por:

$$\sigma(x) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x = 0 \\ -1 & \text{se } x < 0. \end{cases}$$

Outras notações matemáticas são as usuais.

3.2 MÉTODO SIMPLEX PARA O PROBLEMA DE FLUXO EM REDE

A especialização do algoritmo simplex primal de George Dantzig para o problema de fluxo em rede com custo mínimo PL é de grande importância, pois o método simplex primal pode ser realizado diretamente na rede eliminando o peso computacional na atualização da inversa da matriz básica.

Antes da formulação para problema PL especializado, alguns conceitos de rede serão apresentados.

Uma rede pode ser representada por um grafo G que consiste de um conjunto finito $[\mathcal{N}, \mathcal{R}]$ de nós e arcos respectivamente.

Uma rede contendo \bar{I} nós e \bar{J} arcos deve possuir nós e arcos ordenados de forma que exista uma correspondência biunívoca entre os nós da rede e os números inteiros, com $1, \dots, \bar{I}$. A mesma correspondência biunívoca deve existir para os arcos da rede, com $1, \dots, \bar{J}$.

A estrutura da rede é ilustrada através da figura III-1, onde os nós são representados por círculos e os arcos por segmentos de linha orientados que conectam dois nós. A seta no segmento de linha indica a direção de fluxo no arco da rede. Por exemplo, o primeiro arco está direcionado a partir do nó 1 na direção do nó 5.

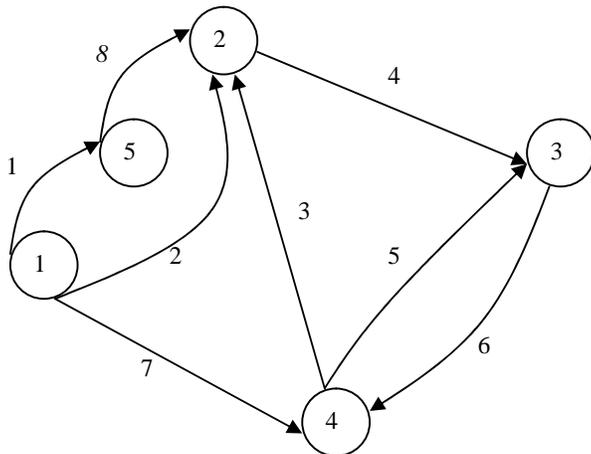


Figura III-1 – Representação de uma rede através de um grafo G .

A estrutura da rede pode também ser descrita por uma matriz A de dimensão $\bar{I} \times \bar{J}$, definida como segue

$$A_{ij} = \begin{cases} +I & \text{direção do arco } j \text{ a partir do nó } i \\ -I & \text{direção do arco } j \text{ na direção do nó } i \\ 0 & \text{outros casos} \end{cases}$$

A matriz A definida acima é chamada de matriz de incidência nó-arco. A matriz de incidência A , correspondente à rede da figura III-1, é mostrada a seguir:

$$A = \begin{array}{cccccccc|c} & \text{Arcos} & & & & & & & & \\ & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \text{Nós} \\ \left[\begin{array}{cccccccc} I & I & & & & & & I & & 1 \\ & -I & -I & I & & & & & -I & 2 \\ & & & -I & -I & I & & & & 3 \\ & & & I & & I & -I & -I & & 4 \\ -I & & & & & & & & I & 5 \end{array} \right] \end{array}$$

A característica dessa matriz é que cada coluna possui exatamente duas entradas não nulas, uma entrada (+1) e a outra (-1).

As definições de outras quantidades associadas à rede são também necessárias. A variável de decisão x_j denota a quantidade de fluxo através do arco j , com o vetor x de dimensão \bar{J} denotando o fluxo em todos os arcos da rede. O custo unitário de fluxo no arco j denotado por c_j , com o correspondente vetor c de dimensão \bar{J} . As capacidades mínima e máxima de fluxo através do arco j denotada por l_j e u_j respectivamente, com os correspondentes vetores l e u de dimensão \bar{J} . As quantidades de oferta e demanda para o nó i denotadas por r_i , com o correspondente vetor r de dimensão \bar{I} . Se $r_i > 0$, o nó i é um *ponto de oferta*, com oferta igual a r_i . Se, $r_i < 0$, o nó i é um *ponto de demanda*, com demanda igual a $|r_i|$. Nós da rede possuindo $r_i = 0$ são chamados *pontos de transbordo*. Matematicamente o problema de rede de mínimo custo pode ser formulado como segue:

$$\text{Min } cx \tag{3.1}$$

$$\text{s.a. } Ax = r \tag{3.2}$$

$$l \leq x \leq u \tag{3.3}$$

onde, A é a matriz de incidência nó-arco, x a quantidade de fluxo através dos arcos, r o vetor de ofertas e demandas associados aos nós da rede, c o vetor de custos, l e u representando as capacidades mínimas e máximas de fluxo nos arcos da rede.

Resultados a partir da teoria de grafos serão utilizados no desenvolvimento do algoritmo especializado para redes.

Considera-se uma rede contendo \bar{I} nós e \bar{J} arcos e uma matriz de incidência A associada à rede. Para cada arco j , tem-se $F(j) = i$, onde $A_{ij} = 1$ e $T(j) = k$, onde $A_{kj} = -1$. O arco j é formalmente descrito como um par $(F(j), T(j))$ tal que $F(j) \neq T(j)$.

Seja $R = \{e_1, e_2, \dots, e_{\bar{J}}\}$ o conjunto de arcos, onde $e_j = (F(j), T(j))$. Seja $N = \{1, 2, \dots, \bar{I}\}$ o conjunto de nós. Então, o grafo G consiste de um conjunto $[N, R]$. Para a rede apresentada na figura III-1 tem-se $R = \{e_1 = (1,5), e_2 = (1,2), e_3 = (4,2), e_4 = (2,3), e_5 = (4,3), e_6 = (3,4), e_7 = (1,4), e_8 = (5,2)\}$.

Um grafo G é dito ser um grafo próprio se $\bar{I} \geq 2$ e $\bar{J} \geq 1$. Um grafo $\hat{G} = [\hat{N}, \hat{R}]$ é subgrafo de G se $\hat{N} \subset N$ e $\hat{R} \subset R$. Se $\hat{N} = N$, então \hat{G} gera G . Estas definições formais e outros resultados descritos a seguir são encontrados em [Ken80].

Uma seqüência finita $P = \{s_1, e_{j_1}, s_2, e_{j_2}, s_3, e_{j_3}, \dots, s_n, e_{j_n}, s_{n+1}\}$ possuindo pelo menos um arco é definida por ser um caminho no grafo G , onde o arco $e_{j_i} \in \{(s_i, s_{i+1}), (s_{i+1}, s_i)\}$. Um exemplo de um caminho no grafo ilustrado na figura III-1 é $P = \{5, e_1, 1, e_7, 4, e_5, 3\} = \{5, (1,5), 1, (1,4), 4, (4,3), 3\}$ com a correspondente representação através da figura III-2.

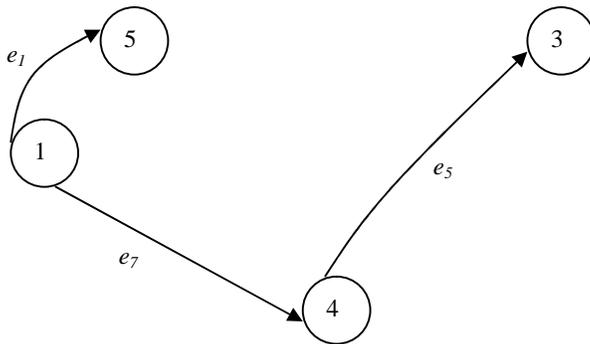


Figura III-2 – Representação de um caminho

Uma seqüência finita $C = \{s_1, e_{j_1}, s_2, e_{j_2}, s_3, e_{j_3}, \dots, s_n, e_{j_n}, s_{n+1}\}$ tendo pelo menos dois arcos é chamado de um ciclo em G se a subsequência $\{s_1, e_{j_1}, s_2, e_{j_2}, \dots, s_n\}$ é um caminho em G , $e_{j_n} \in \{(s_n, s_{n+1}), (s_{n+1}, s_n)\}$, $s_1 = s_{n+1}$, e $e_{j_n} \neq e_{j_1}$. Um exemplo de um ciclo no grafo representado na figura III-1 é $C = \{2, e_8, 5, e_1, 1, e_2, 2\}$.

O comprimento de um caminho ou ciclo é o número de arcos no caminho ou ciclo. Para cada caminho ou ciclo, com comprimento n , a orientação da seqüência $O(P)$ de n elementos é definida como segue:

$$O_i(P) = \begin{cases} +1, & \text{se } e_{j_i} = (\mathbf{s}_i, \mathbf{s}_{i+1}) \\ -1, & \text{se } e_{j_i} = (\mathbf{s}_{i+1}, \mathbf{s}_i) \end{cases}.$$

Para o caminho $\{5, e_1, 1, e_7, 4, e_5, 3\}$ ilustrado na figura III-2 a orientação da seqüência é $O(P) = \{-1, 1, 1\}$.

Algumas proposições demonstradas em [Ken80] serão enunciadas a seguir. Essas proposições servem de suporte para resultados teóricos posteriores.

Proposição 1 Se uma seqüência finita $P = \{\mathbf{s}_1, e_{j_1}, \mathbf{s}_2, e_{j_2}, \dots, \mathbf{s}_n, e_{j_n}, \mathbf{s}_{n+1}\}$ é um caminho ou ciclo em um grafo próprio G com matriz de incidência A , então:

$$\sum_{i=1}^n O_i(P) A(j_i) = \mathbf{e}^{s_i} - \mathbf{e}^{s_{n+1}}. \quad (3.4)$$

onde \mathbf{e}^{s_i} é o vetor coluna canônico com entrada 1 na i -ésima linha e entrada zero nas demais linhas.

Para o caminho $P = \{5, e_1, 1, e_7, 4, e_5, 3\}$ do grafo representado na figura III-2 tem-se:

$$e_1 = e_{j_1} = (\mathbf{s}_2, \mathbf{s}_1) = (1, 5), \text{ então } O_1(P) A(j_1) = -1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

$$e_7 = e_{j_2} = (\mathbf{s}_3, \mathbf{s}_2) = (4, 1), \text{ então } O_7(P) A(j_2) = -1 \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$e_5 = e_{j_3} = (\mathbf{s}_3, \mathbf{s}_4) = (4, 3), \text{ então } O_5(P) A(j_3) = 1 \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix}$$

Portanto

$$\sum_{i=1}^3 O_i(P)A(j_i) = -1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} -1 \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} +1 \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \mathbf{e}^{s_j} - \mathbf{e}^{s_d}.$$

Proposição 2 Se $C = \{s_1, e_{j_1}, s_2, e_{j_2}, \dots, s_n, e_{j_n}, s_{n+1}\}$ é ciclo de um grafo próprio G com matriz de incidência A , então:

$$\sum_{i=1}^n O_i(C)A(j_i) = 0 \quad (3.5)$$

Corolário 3 Se $C = \{s_1, e_{j_1}, s_2, e_{j_2}, \dots, s_n, e_{j_n}, s_{n+1}\}$ é ciclo de um grafo próprio G com matriz de incidência A , então $\{A(j_i): i=1, \dots, n\}$ é linearmente dependente.

Um grafo $G = [\mathcal{N}, \mathcal{R}]$ é acíclico se não possui ciclos podendo ser formado a partir de \mathcal{N} e \mathcal{R} . Um grafo $G = [\mathcal{N}, \mathcal{R}]$ é conexo se, para cada par de nós distintos (i, j) um caminho pode ser formado a partir de i até j . Uma árvore τ é um grafo conexo e acíclico. Uma árvore τ é um subgrafo gerador de G , também chamada de árvore geradora para G .

Proposição 4 Seja A uma matriz de incidência nó-arco para um grafo próprio G . Seja $\tau = [\mathcal{N}, \mathcal{R}]$ um subgrafo de G , isto é, uma árvore contendo pelo menos dois nós. Então $\{A(j): e_j \in \mathcal{R}\}$ é linearmente independente.

Proposição 5 Seja A uma matriz de incidência nó-arco para um grafo próprio G conexo e com n nós. Então o posto da matriz A é $n-1$.

Proposição 6 Seja A uma matriz de incidência nó-arco para um grafo próprio $G = [\mathcal{N}, \mathcal{R}]$, onde G possui n nós. Seja $\widehat{\mathcal{R}}$ um subconjunto de \mathcal{R} tal que $\{A(j): e_j \in \widehat{\mathcal{R}}\}$ é linearmente independente com $\widehat{\mathcal{R}}$ tendo $n-1$ arcos. Então, $\tau = [\mathcal{N}, \widehat{\mathcal{R}}]$ é uma árvore.

3.2.1 Caracterização de uma Base para o Problema de Fluxo em Rede

Assume-se que, a matriz de restrições A possui posto completo. Pela Proposição 5, a matriz de restrição para o problema possui o posto menor que o número de linhas da matriz. Com o objetivo de tornar o posto da matriz A completo, reescreve-se o problema (3.1)-(3.3) como:

$$\text{Min } cx \quad (3.6)$$

$$\text{s.a. } Ax + ae^1 = r \quad (3.7)$$

$$l \leq x \leq u \quad (3.8)$$

$$0 \leq a \leq \theta \quad (3.9)$$

onde, $\mathbf{1}$ é um inteiro positivo não maior que o número de nós n com a restrito a zero. A solução ótima para (3.1) também será solução ótima para (3.6).

Observe que o vetor canônico e^1 possui sua entrada não nula (+1) correspondente ao nó raiz $\mathbf{1}$ do grafo. A coluna j , acrescida a matriz de incidência A , corresponde ao vetor canônico e^1 tornando a matriz A com composto completo.

Proposição 7 Seja A uma matriz de incidência nó-arco para um grafo próprio conexo $G=[\mathcal{N},\mathcal{R}]$ possuindo n nós. Seja $\tau=[\mathcal{N},\widehat{\mathcal{R}}]$ uma árvore geradora para G . Então, $\Omega = \{A(j): e_j \in \widehat{\mathcal{R}}\} \cup \{e^1\}$ gera E^n , isto é, existe um conjunto de n colunas a partir de $[A|e^1]$ que gera E^n .

Proposição 8 Seja A uma matriz de incidência nó-arco para um grafo próprio conexo $G=[\mathcal{N},\mathcal{R}]$ com nó raiz $\mathbf{1}$. Se Ω é uma base para $[A|e^1]$, então $e^1 \in \Omega$ e $\tau=[\mathcal{N},\widehat{\mathcal{R}}]$ é uma árvore geradora para G , onde $\widehat{\mathcal{R}}=\{e_j: A(j) \in \Omega\}$.
A partir das proposições acima, pode-se caracterizar bases para (3.6).

Proposição 9 Seja A uma matriz de incidência nó-arco para um grafo próprio, conexo e com um nó raiz $\mathbf{1}$. Uma base para $[A|e^1]$ é um conjunto de arcos que compõem as colunas da matriz A correspondente à árvore geradora τ para G .

Proposição 10 Seja A uma matriz de incidência nó-arco para um grafo próprio, conexo e com um nó raiz $\mathbf{1}$. Seja B uma base a partir de $[A|e^1]$. Então, B é triangular.

A estratégia utilizada na demonstração, da Proposição 10, que determina a linha e coluna ordenada a fim de se obter uma base para a rede na forma triangular superior fez com que o algoritmo de triangulação fosse desenvolvido, ver [Ken80].

3.2.2 Método Simplex Primal no Grafo

Considerando que as bases para o problema (3.6) sejam triangulares, as operações do algoritmo primal simplex especializado que envolve a base B (a partir de $[A|e^1]$) ou B^{-1} podem ser realizadas diretamente no grafo que representa a rede. Para determinar uma variável não básica candidata a entrar na base (operação de *pricing*), deve-se determinar a componente correspondente de $[c^N - c^B B^{-1} N]$. Considere somente $c^B B^{-1} N$, e faça $\pi = c^B B^{-1}$, onde π_i 's são geralmente chamados de variáveis duais. Mas $\pi = c^B B^{-1}$ é a solução para o sistema de equações lineares $\pi B = c^B$. Como B é triangular, π pode ser obtido por substituição simples. Entretanto, pode-se fazer uso da árvore associada com B , isto é, τ_B para efetuar os cálculos. Em geral, para a base B e a correspondente árvore básica τ_B com nó raiz $\mathbf{1}$, $\pi B = c^B$ resume-se a determinar

$$\left\{ \begin{array}{l} \pi_{\mathbf{1}} = 0 \\ \pi_{F(j)} - \pi_{T(j)} = c_j \quad \forall e_j \in \tau_B \end{array} \right\}$$

quando B é triangular. O algoritmo que pode ser utilizado iterativamente para determinar π é apresentado no Anexo A.1.

Todas as variáveis duais são unicamente determinadas iterativamente porque a árvore tem um nó raiz, é conexa e não possui ciclos. Quando $\pi = c^B B^{-1}$ é utilizado, os conjuntos que determinam os candidatos a entrarem na base no simplex especializado são:

$$\begin{aligned} \psi_1 &= \{ e_j: x_j = l_j \text{ e } \pi_{F(j)} - \pi_{T(j)} - c_j > 0 \} \\ &\quad \text{e} \\ \psi_2 &= \{ e_j: x_j = u_j \text{ e } \pi_{F(j)} - \pi_{T(j)} - c_j < 0 \}. \end{aligned}$$

A componente $c^B B^{-1} N$ pode também ser determinada do seguinte modo:

Seja B a base com correspondente árvore τ_B . A i -ésima componente de $c^B B^{-1} N$ é dada por $c^B B^{-1} N(i)$. Suponha que $A(k) = N(i)$. Então, pode-se determinar $y = B^{-1} A(k)$ e calcular $c^B y$, onde $e_k \notin \tau_B$. Mas isto, equivale determinar $By = A(k) = e^{F(j)} - e^{T(j)}$. Desde que B seja triangular, y

pode ser obtido diretamente e, portanto o cálculo de B^{-1} não é necessário. Novamente, utiliza-se τ_B para resolver o sistema triangular. Seja, $P = \{s_1, e_{j_1}, s_2, e_{j_2}, \dots, s_n, e_{j_n}, s_{n+1}\}$ o único caminho na árvore τ_B unindo $F(k)$ ao $T(k)$. Pela Proposição 1

$$\sum_{i=1}^n O_i(P) A(j_i) = \mathbf{e}^{F(k)} - \mathbf{e}^{T(k)}. \quad (3.10)$$

O cálculo de $c^B B^{-1} N(i) = c^B y$ pode ser determinado como:

$$\sum_{i=1}^n c_{j_i} O_i(P). \quad (3.11)$$

A equação (3.10) também indica que o teste de razão pode ser especializado, no cálculo de $y = B^{-1} A(k)$. Se os arcos na árvore τ_B são ordenados como: $e_{k_1}, e_{k_2}, \dots, e_{k_{j\bar{I}}}$, correspondendo às colunas \bar{I} de B , então, as componentes \bar{I} de y podem ser determinadas pela orientação da seqüência:

$$y_n = \begin{cases} O_i(P) & \text{se } e_{k_n} = e_{j_i} \in P, \\ 0 & \text{caso contrário} \end{cases} \quad (3.12)$$

e o cálculo de Δ_1 e Δ_2 no teste de razão é especializado como:

$$\Delta_1 \leftarrow \min_{O_i(P)=\delta} \{x_{j_i} - l_{j_i}, \infty\} \quad \text{e} \quad \Delta_2 \leftarrow \min_{-O_i(P)=\delta} \{u_{j_i} - x_{j_i}, \infty\}$$

Considere a árvore com nó raiz representada na figura III-3 com sua matriz básica correspondente:

$$B = \begin{bmatrix} e_1 & e_4 & e_2 & e_3 & e^1 \\ 1 & & & & 1 \\ & -1 & & & 4 \\ & & -1 & & 2 \\ -1 & & 1 & -1 & 3 \\ & 1 & & 1 & 1 \\ & & & & 5 \end{bmatrix} \begin{matrix} \text{Nós} \\ 1 \\ 4 \\ 2 \\ 3 \\ 5 \end{matrix}$$

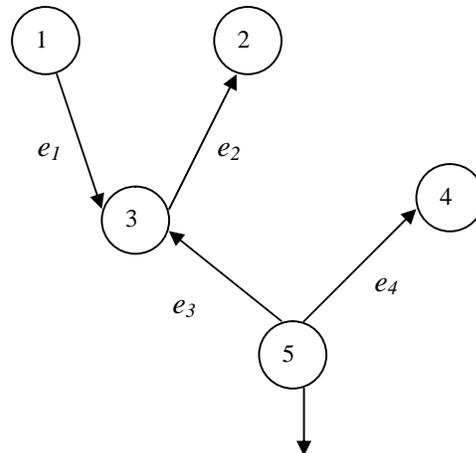


Figura III-3 – Árvore com nó raiz

Suponha que o arco fora da base seja $e_k = (2,4)$. Então $P = \{2, e_2, 3, e_3, 5, e_4, 4\}$ e $O_i(P) = \{-1, -1, 1\}$. O ciclo formado e o cálculo para y são ilustrados na figura III-4.

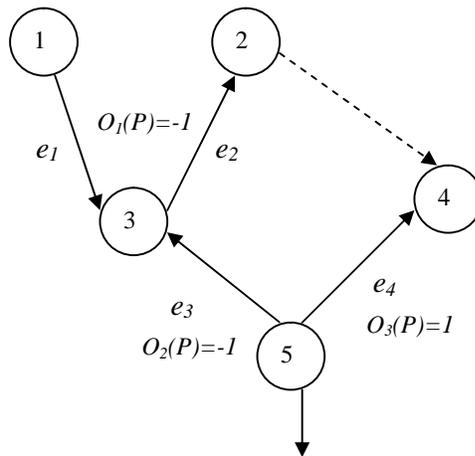


Figura III-4 – Orientação dos arcos no ciclo formado com o arco fora da base.

O algoritmo simplex primal especializado é então enunciado:

ALGORITMO 3.1 Algoritmo Simplex Primal Especializado para NP

P1. Inicialização

Seja $[x^B | x^N]$ uma solução básica viável com árvore τ_B .

P2. Variável candidata a entrar na base.

Sejam os conjuntos:

$$\psi_1 = \{ e_j: x_j = l_j \text{ e } \pi_{F(j)} - \pi_{T(j)} - c_j > 0 \}$$

e

$$\psi_2 = \{ e_j: x_j = u_j \text{ e } \pi_{F(j)} - \pi_{T(j)} - c_j < 0 \}.$$

Se $\psi_1 \cup \psi_2 = \emptyset$, o algoritmo termina com a solução ótima. Caso contrário, seleciona-se

$e_k \in \psi_1 \cup \psi_2$ para entrar na base.

$$\text{Assim, } \delta \leftarrow \begin{cases} +1, & \text{se } e_k \in \psi_1 \\ -1, & \text{se } e_k \in \psi_2 \end{cases}.$$

P3. Teste de Razão.

Seja, $P = \{s_1, e_{j_1}, s_2, e_{j_2}, \dots, s_n, e_{j_n}, s_{n+1}\}$, o caminho na árvore τ_B unindo $F(k)$ ao $T(k)$.

$$\text{Faça } \Delta_1 \leftarrow \min_{O_i(P)=\delta} \{x_{j_i} - l_{j_i}, \infty\},$$

$$\Delta_2 \leftarrow \min_{-O_i(P)=\delta} \{u_{j_i} - x_{j_i}, \infty\},$$

$$\Delta \leftarrow \min \{ \Delta_1, \Delta_2, u_k - l_k \}.$$

P4. Atualização dos Fluxos.

Faça $x_k \leftarrow x_k + \Delta\delta$. Para todo $e_j \in P$, faça $x_{j_i} \leftarrow x_{j_i} - \Delta\delta O_i(P)$. Se $\Delta = u_k - l_k$, retorne ao

P2.

P5. *Atualização da Árvore e Variáveis Duais.*

Sejam os conjuntos:

$$\Psi_3 = \{ e_{j_i} : x_{j_i} = l_{j_i} \text{ onde } O_i(P) = \delta \}$$

e

$$\Psi_4 = \{ e_{j_i} : x_{j_i} = u_{j_i} \text{ onde } -O_i(P) = \delta \}.$$

Selecione algum $e_m \in \Psi_3 \cup \Psi_4$. Permute e_m em τ_B com e_k , atualize as variáveis duais e retorne a P2.

3.3 MÉTODO SIMPLEX PRIMAL PARTICIONADO PARA O PROBLEMA DE FLUXO EM REDE COM MÚLTIPLOS PRODUTOS

O método simplex com particionamento primal para múltiplos produtos é descrito por Kennington e Helgason [Ken80] e mais tarde por Castro e Nabona [Cas96] que, baseados na especialização de Murtagh e Saunders [Mur78], utilizando a estratégia de dividir o conjunto de variáveis dentro de básicas, não básicas e superbásicas, implementam o método primal particionado. A formulação matemática do problema na sua forma geral é enunciada e toda técnica de solução é mostrada posteriormente. Os resultados algébricos mostrados não envolvem grandes dificuldades matemáticas, no entanto, a notação utilizada é bastante complexa.

3.3.1 Formulação Matemática

A formulação matemática¹ para o problema de rede para múltiplos produtos na sua forma geral é

$$MP \quad \text{Min } z = \sum_k c^k x^k \quad (3.13)$$

$$s.a: \quad A^k x^k = r \quad k=1, \dots, K \quad (3.14)$$

$$\sum_k D^k x^k \leq b \quad (3.15)$$

$$l^k \leq x^k \leq u^k \quad k=1, \dots, K \quad (3.16)$$

onde:

x^k vetor de fluxos do produto k ;

¹ A notação utilizada na formulação matemática para MP segue a notação utilizada por [Ken80].

- z custo total de distribuição de todos os produtos (função objetivo);
 A^k matriz de incidência nó-arco associada ao produto k ;
 D^k matriz de coeficientes das restrições de agregação de fluxos para o produto k ;
 b vetor de capacidades associadas às restrições de agregação de fluxos;
 c^k vetor de custos associados ao produto k ;
 r^k vetor de ofertas e demandas associados ao produto k ;
 u^k capacidade máxima de fluxo nos arcos associadas ao produto k ;
 l^k capacidade mínima de fluxo nos arcos associadas ao produto k ;

São chamadas de restrições de agregação de fluxos às restrições que podem unir arcos de um mesmo produto ou de produtos diferentes.

3.3.2 Particionamento da Base

Pela adição de variáveis artificiais e de folgas (s) ao problema formulado anteriormente obtém-se um problema com restrições de igualdade, da seguinte forma:

$$MP \quad \text{Min } z = \sum_k c^k x^k \quad (3.17)$$

$$s.a. \quad Ax^k + \mathbf{e}^1 a_k = r^k \quad \forall k \quad (3.18)$$

$$\sum_k D^k x^k + s = b \quad (3.19)$$

$$l^k \leq x^k \leq u^k \quad 0 \leq a_k \leq 0 \quad \forall k, s \geq 0 \quad (3.20)$$

onde, $\mathbf{1}$ é o nó raiz para o grafo G associado ao produto k garantindo que $[A^k \mid \mathbf{e}^1]$ seja uma matriz com posto completo.

A matriz de restrição associada com (3.18) e (3.19) pode ser colocada sob a seguinte forma

$$\hat{A} = \left[\begin{array}{c|c|c|c|c} A^1 / \mathbf{e}^1 & & & & \\ \hline & A^2 / \mathbf{e}^1 & & & \\ \hline & & \ddots & & \\ \hline & & & A^K / \mathbf{e}^1 & \\ \hline D^1 / 0 & D^2 / 0 & \dots & D^K / 0 & I \end{array} \right] = \left[\begin{array}{c|c|c|c|c} \bar{A}^1 & & & & \\ \hline & \bar{A}^2 & & & \\ \hline & & \ddots & & \\ \hline & & & \bar{A}^K & \\ \hline \bar{D}^1 & \bar{D}^2 & & \bar{D}^K & I \end{array} \right] \quad (3.21)$$

Nesta matriz \hat{A} , pode-se obter uma base particionada com a seguinte estrutura:

$$\bar{B} = \left[\begin{array}{ccc|ccc|c} B^1 & & & R^1 & & & 0 \\ & \ddots & & & \ddots & & \\ & & B^K & & & R^K & \\ \hline P^1 & \dots & P^K & T^1 & \dots & T^K & 0 \\ \hline S^1 & \dots & S^K & U^1 & \dots & U^K & I \end{array} \right] = \left[\begin{array}{c|c} B & C \\ \hline D & F \end{array} \right] \quad (3.22)$$

para o grafo $G = [\mathcal{N}, \mathcal{R}]$. A demonstração para obter \bar{B} pode ser encontrada em [Ken80],

onde se tem:

$$B = \begin{bmatrix} B^1 & 0 & \dots & 0 \\ 0 & B^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B^K \end{bmatrix} \quad C = \begin{bmatrix} R^1 & 0 & \dots & 0 & | & 0 \\ 0 & R^2 & \dots & 0 & | & 0 \\ \vdots & \vdots & \ddots & \vdots & | & \vdots \\ 0 & 0 & \dots & R^K & | & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} P^1 & P^2 & \dots & P^K \\ S^1 & S^2 & \dots & S^K \end{bmatrix} \quad F = \begin{bmatrix} T^1 & T^2 & \dots & T^K & | & 0 \\ U^1 & U^2 & \dots & U^K & | & I \end{bmatrix}$$

com:

- B^k uma matriz de incidência nó-arco que forma uma base para \bar{A}^k associada ao produto k ;
- R^k matriz de incidência nó-arco associada aos arcos superbásicos² definidas para o produto k ;
- P^k matriz de coeficientes associados aos arcos básicos do produto k , nas restrições ativas de agregação de fluxo;
- S^k matriz de coeficientes associados aos arcos básicos do produto k , nas restrições não ativas de agregação de fluxo;
- T^k matriz de coeficientes associados aos arcos superbásicos (complementares) do produto k , nas restrições ativas de agregação de fluxo;
- U^k matriz de coeficientes associados aos arcos superbásicos (complementares) do produto k , nas restrições não ativas de agregação de fluxo;
- I matriz identidade associada às folgas nas restrições não ativas de agregação.

² Arcos superbásicos são aqueles que participam das restrições de agregação que se encontram na base.

Proposição 11 Seja \bar{A}^k uma matriz de incidência nó-arco associada a um grafo G^k com nó raiz $\mathbf{1}$. Se B^k é uma base para $[A^k / \mathbf{e}^1]$ com partição R^k vazia, então $\mathbf{e}^1 \in B^k$ e $\tau^k = [\mathcal{N}, \hat{\mathcal{R}}]$ é uma árvore geradora para G^k , onde $\hat{\mathcal{R}} = \{e_j : \bar{A}^k(j) \in B^k\}$.

Justificativa: \bar{A}^k uma matriz de incidência associada a um grafo $G^k[\mathcal{N}, \mathcal{R}]$, onde G^k possui n nós. Se B^k é uma base para $[A^k / \mathbf{e}^1]$ então existe um subconjunto de \mathcal{R} linearmente independente tal que $\{e_j : \bar{A}^k(j) \in B^k\}$ com $n-1$ arcos. Portanto $\tau^k = [\mathcal{N}, \hat{\mathcal{R}}]$ é uma árvore geradora para G^k , onde $\hat{\mathcal{R}} = \{e_j : \bar{A}^k(j) \in B^k\}$.

A demonstração para a Proposição 11 pode ser encontrada em [Ken80].

Proposição 12

Se \bar{B} é inversível, então \bar{B}^{-1} será dada por:

$$\bar{B}^{-1} = \left[\begin{array}{c|c} B^{-1} + B^{-1}CY^{-1}DB^{-1} & -B^{-1}CY^{-1} \\ \hline -Y^{-1}DB^{-1} & Y^{-1} \end{array} \right]$$

onde $Y = F - DB^{-1}C$.

Prova

(i) $B = \begin{bmatrix} B^1 & 0 & \dots & 0 \\ 0 & B^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B^K \end{bmatrix}$ é inversível, pois B^1, B^2, \dots, B^K são bases para os PL 's.

(ii) $Y = F - DB^{-1}C$ é inversível, pois:

- se \bar{B} admite inversa, então $\det(\bar{B}) \neq 0$;
- sendo $W = \left[\begin{array}{c|c} I & -B^{-1}C \\ \hline \vdots & I \end{array} \right]$ então $\det(W) \neq 0$, por construção;
- com isto tem-se $\bar{B} \cdot W = \left[\begin{array}{c|c} B & \\ \hline D & F - DB^{-1}C \end{array} \right]$, que possui posto completo, visto que $\det(\bar{B} \cdot W) = \det(\bar{B}) \cdot \det(W) \neq 0$;
- assim, se $\bar{B} \cdot W$ tem posto completo, então $Y = F - DB^{-1}C$ também o terá, e Y será inversível.

(iii) $\bar{B} \cdot \bar{B}^{-1} = \left[\begin{array}{c|c} B & C \\ \hline D & F \end{array} \right] \left[\begin{array}{c|c} B^{-1} + B^{-1}CY^{-1}DB^{-1} & -B^{-1}CY^{-1} \\ \hline -Y^{-1}DB^{-1} & Y^{-1} \end{array} \right] = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$. ■

Assim:

$$\begin{aligned}
Y &= \left[\begin{array}{cccc|c} T^1 & T^2 & \dots & T^K & 0 \\ U^1 & U^2 & \dots & U^K & I \end{array} \right] - \left[\begin{array}{cccc} P^1 & P^2 & \dots & P^K \\ S^1 & S^2 & \dots & S^K \end{array} \right] \cdot \left[\begin{array}{cccc} (B^1)^{-1} & 0 & \dots & 0 \\ 0 & (B^2)^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (B^K)^{-1} \end{array} \right] \cdot \left[\begin{array}{cccc|c} R^1 & 0 & \dots & 0 & 0 \\ 0 & R^2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & R^K & 0 \end{array} \right] \\
Y &= \left[\begin{array}{cccc|c} T^1 & T^2 & \dots & T^K & 0 \\ U^1 & U^2 & \dots & U^K & I \end{array} \right] - \left[\begin{array}{cccc} P^1(B^1)^{-1} & P^2(B^2)^{-1} & \dots & P^K(B^K)^{-1} \\ S^1(B^1)^{-1} & S^2(B^2)^{-1} & \dots & S^K(B^K)^{-1} \end{array} \right] \cdot \left[\begin{array}{cccc|c} R^1 & 0 & \dots & 0 & 0 \\ 0 & R^2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & R^K & 0 \end{array} \right] \\
Y &= \left[\begin{array}{cccc|c} T^1 & T^2 & \dots & T^K & 0 \\ U^1 & U^2 & \dots & U^K & I \end{array} \right] - \left[\begin{array}{cccc} P^1(B^1)^{-1}R^1 & P^2(B^2)^{-1}R^2 & \dots & P^K(B^K)^{-1}R^K \\ S^1(B^1)^{-1}R^1 & S^2(B^2)^{-1}R^2 & \dots & S^K(B^K)^{-1}R^K \end{array} \right] \cdot \left[\begin{array}{cccc|c} 0 \\ 0 \\ \vdots \\ 0 \end{array} \right] \\
Y &= \left[\begin{array}{cccc|c} T^1 - P^1(B^1)^{-1}R^1 & T^2 - P^2(B^2)^{-1}R^2 & \dots & T^K - P^K(B^K)^{-1}R^K & 0 \\ U^1 - S^1(B^1)^{-1}R^1 & U^2 - S^2(B^2)^{-1}R^2 & \dots & U^K - S^K(B^K)^{-1}R^K & I \end{array} \right] = \left[\begin{array}{c|c} Q & 0 \\ \hline V & I \end{array} \right]. \quad (3.23)
\end{aligned}$$

A matriz Q é denominada matriz ciclo ou (matriz de trabalho), pois ciclos são gerados na árvore básica. Mais especificamente, quando se refere à matriz ciclo, refere-se à matriz Q , que gera ciclos na árvore básica a partir dos arcos complementares (ou superbásicos), cujos arcos correspondem às restrições ativas de agregação.

3.3.3 Determinação da Matriz Ciclo Q

Seja $\Omega^k = \{e_{m_1}, \dots, e_{m_q}\} \subset \mathcal{R}$ o conjunto indexado de arcos correspondentes as r -ésimas linhas de $[T^1 \dots T^k]$. Para cada $e_{m_q} \in \Omega^k$ existe uma linha indexada r tal que o arco e_{m_r} denota o arco correspondente a r -ésima linha. Cabe salientar que a linha indexada r corresponde a uma restrição ativa de agregação que poderá unir arcos de um mesmo produto ou de diferentes produtos.

Seja $\nu^k = \{e_{j_1}, \dots, e_{j_k}\} \subset \mathcal{R}$ o conjunto indexado correspondente às colunas de R^k . O conjunto indexado ν^k é ordenado tal que e_{j_s} denota o arco correspondente a s -ésima coluna de R^k . A matriz ciclo Q pode ser obtida a partir da árvore básica τ^k , R^k e dos conjuntos indexados.

Observa-se em (3.23), que a k -ésima partição da matriz Q é dada por $T^k - P^k (B^k)^{-1} R^k$, onde R^k expressa cada um dos arcos complementares para o k -ésimo produto e sua união com a árvore geradora mínima. Se τ^k denota a árvore enraizada geradora correspondente a B^k , então para cada arco complementar $e_{j_s} \in \nu^k$, existe um único caminho na árvore τ^k , denotado P_s , unindo $T(j_s)$ ao $F(j_s)$. Se $O_i(s)$ denota a orientação na seqüência correspondente a P_s , então dado um arco $e_{j_i} \in P_s$, diz-se que o arco e_{j_i} possui orientação normal se aponta para a origem do nó do arco complementar e_{j_s} ; caso contrário terá orientação reversa.

Se P_r^k denota a r -ésima linha de P^k e β^s denota a s -ésima coluna de $(B^k)^{-1} R^k$. Então cada elemento Q_{rs} pode ser determinado da seguinte forma:

$$Q_{rs}^k = T_{rs}^k - P_r^k \beta^s \quad (3.24)$$

onde para um produto indexado k , T_{rs}^k indica o coeficiente do arco complementar na restrição ativa de agregação, com $m_r = j_s$, onde $e_{m_q r} \in \Omega^k$ e $e_{j_s} \in \nu^k$. O produto matricial $P_r^k \beta^s$ indica o somatório do produto dos coeficientes dos arcos básicos na restrição ativa de agregação pela sua orientação na seqüência P_s , ou seja, com e_{m_r} um arco da t -ésima coluna de B^k e $e_{m_r} \in P_s$ (com orientação normal ou reversa) onde $e_{m_q r} \in \Omega^k$.

Para resolver o algoritmo simplex especializado com particionamento primal para o problema MP , é necessário que sejam resolvidos os sistemas que determinam as variáveis duais e a atualização da coluna. Entretanto, é visto a seguir que, os sistemas de equações são reduzidos pelos fatores da matriz de operação Q ao invés da base \bar{B} e pelo procedimento de atualização dos fatores dessa matriz a cada iteração.

3.3.4 Determinação das Variáveis Duais e Atualização da Coluna na Resolução do Algoritmo com Particionamento Primal

Através de resultados algébricos obtém-se a matriz \bar{B}^{-1} necessária para a resolução dos sistemas que determinam as variáveis duais e a atualização da coluna não básica candidata a entrar na base.

Seja a base particionada com a seguinte estrutura:

$$\bar{B} = \left[\begin{array}{ccc|cc} B^l & & & R^l & \\ & \ddots & & & \\ & & B^K & & R^K \\ \hline P^l & & P^K & T^l & T^K \\ \hline S^l & & S^K & U^l & U^K \quad I \end{array} \right]$$

e, seja a matriz de transformação

$$W = \left[\begin{array}{ccc|cc} I & & & (-B^l)^{-1}R^l & \\ & \ddots & & & \\ & & I & & (-B^K)^{-1}R^K \\ \hline & & & I & \\ \hline & & & & I \\ \hline & & & & I \end{array} \right] \text{ tal que}$$

$$\xi = \bar{B}W = \left[\begin{array}{ccc|cc} B^l & & & & \\ & \ddots & & & \\ & & B^K & & \\ \hline P^l & \dots & P^K & T^l - P^l(B^l)^{-1}R^l & \dots & T^K - P^K(B^K)^{-1}R^K & 0 \\ \hline S^l & \dots & S^K & U^l - S^l(B^l)^{-1}R^l & \dots & U^K - S^K(B^K)^{-1}R^K & I \end{array} \right] \quad (3.25)$$

$$\xi = \bar{B}W = \left[\begin{array}{ccc|cc} B^l & & & & \\ & \ddots & & & \\ & & B^K & & \\ \hline P^l & \dots & P^K & Q & 0 \\ \hline S^l & \dots & S^K & V & I \end{array} \right] = \left[\begin{array}{c|c} B & 0 \\ \hline D & Y \end{array} \right].$$

Pela Proposição 12, tem-se

$$\xi^{-1} = \left[\begin{array}{c|c} B^{-1} & 0 \\ \hline -Y^{-1}DB^{-1} & Y^{-1} \end{array} \right], \text{ onde}$$

a partição

$$Y^{-1} = \left[\begin{array}{c|c} Q^{-1} & 0 \\ \hline -VQ^{-1} & I \end{array} \right] \text{ e}$$

a partição

$$\begin{aligned}
-Y^{-1}DB^{-1} &= -\left[\begin{array}{c|c} Q^{-1} & 0 \\ \hline -VQ^{-1} & I \end{array} \right] \left[\begin{array}{c|c|c} P^1 & \dots & P^K \\ \hline S^1 & \dots & S^K \end{array} \right] \left[\begin{array}{c|c} (B^1)^{-1} & \\ \hline & (B^K)^{-1} \end{array} \right] \\
&= -\left[\begin{array}{c|c} Q^{-1} & 0 \\ \hline -VQ^{-1} & I \end{array} \right] \left[\begin{array}{c|c|c} P^1(B^1)^{-1} & \dots & P^K(B^K)^{-1} \\ \hline S^1(B^1)^{-1} & \dots & S^K(B^K)^{-1} \end{array} \right] \\
&= -\left[\begin{array}{c|c|c} Q^{-1}P^1(B^1)^{-1} & \dots & Q^{-1}P^K(B^K)^{-1} \\ \hline -VQ^{-1}P^1(B^1)^{-1} + S^1(B^1)^{-1} & \dots & -VQ^{-1}P^K(B^K)^{-1} + S^K(B^K)^{-1} \end{array} \right].
\end{aligned}$$

Então,

$$\xi^{-1} = (\bar{B}W)^{-1} = \left[\begin{array}{c|c|c|c|c} & (B^1)^{-1} & & & \\ & & \ddots & & \\ & & & (B^K)^{-1} & \\ \hline & -Q^{-1}P^1(B^1)^{-1} & \dots & -Q^{-1}P^K(B^K)^{-1} & Q^{-1} & 0 \\ \hline & VQ^{-1}P^1(B^1)^{-1} - S^1(B^1)^{-1} & \dots & VQ^{-1}P^K(B^K)^{-1} - S^K(B^K)^{-1} & -VQ^{-1} & I \end{array} \right].$$

Logo

$$(\bar{B}W)^{-1} = \xi^{-1}$$

$$(\bar{B})^{-1} = W\xi^{-1}$$

$$\bar{B}^{-1} = \left[\begin{array}{c|c|c|c|c} (B^1)^{-1} + (B^1)^{-1}R^1Q^{-1}P^1(B^1)^{-1} & \dots & (B^1)^{-1}R^1Q^{-1}P^K(B^K)^{-1} & -(B^1)^{-1}R^1Q^{-1} & 0 \\ & \vdots & \vdots & \vdots & \vdots \\ (B^K)^{-1}R^KQ^{-1}P^1(B^1)^{-1} & \dots & (B^K)^{-1} + (B^K)^{-1}R^KQ^{-1}P^K(B^K)^{-1} & -(B^K)^{-1}R^KQ^{-1} & 0 \\ \hline -Q^{-1}P^1(B^1)^{-1} & \dots & -Q^{-1}P^K(B^K)^{-1} & Q^{-1} & 0 \\ \hline VQ^{-1}P^1(B^1)^{-1} - S^1(B^1)^{-1} & \dots & VQ^{-1}P^K(B^K)^{-1} - S^K(B^K)^{-1} & -VQ^{-1} & I \end{array} \right] \quad (3.26)$$

A estrutura da base \bar{B} pode ser utilizada eficientemente para calcular as variáveis duais $\pi = c_B(\bar{B})^{-1}$, obtendo-se assim, a solução para o primeiro sistema que resolve o simplex particionado.

$$[\pi^1 \mid \pi^2 \mid \pi^3] = [c_B^1 \mid c_B^2 \mid 0] \left[\begin{array}{c|c|c|c|c} (B^1)^{-1} + (B^1)^{-1}R^1Q^{-1}P^1(B^1)^{-1} & \dots & (B^1)^{-1}R^1Q^{-1}P^K(B^K)^{-1} & -(B^1)^{-1}R^1Q^{-1} & 0 \\ & \vdots & \vdots & \vdots & \vdots \\ (B^K)^{-1}R^KQ^{-1}P^1(B^1)^{-1} & \dots & (B^K)^{-1} + (B^K)^{-1}R^KQ^{-1}P^K(B^K)^{-1} & -(B^K)^{-1}R^KQ^{-1} & 0 \\ \hline -Q^{-1}P^1(B^1)^{-1} & \dots & -Q^{-1}P^K(B^K)^{-1} & Q^{-1} & 0 \\ \hline VQ^{-1}P^1(B^1)^{-1} - S^1(B^1)^{-1} & \dots & VQ^{-1}P^K(B^K)^{-1} - S^K(B^K)^{-1} & -VQ^{-1} & I \end{array} \right]$$

Para solucionar o sistema acima as partições c_B^1 e c_B^2 que representam os custos para os arcos básicos passam a ser denotadas por:

$c_B^1 = [c_B^{11} \mid c_B^{12} \mid \dots \mid c_B^{1K}] \forall_k$, indicando os custos associados aos arcos básicos e_j ;

$c_B^2 = [c_B^{21} \mid c_B^{22} \mid \dots \mid c_B^{2K}] \forall_k$, indicando os custos associados aos arcos complementares e_j .

Resolvendo o sistema por substituição tem-se

$$\pi^3 = 0$$

$$\begin{aligned} \pi^2 &= [-c_B^{11}(B^1)^{-1}R^1Q^{-1} \quad \dots \quad -c_B^{1K}(B^K)^{-1}R^KQ^{-1}] + [c_B^{21}Q^{-1} \quad \dots \quad c_B^{2K}Q^{-1}] \\ &= [c_B^{21} - c_B^{11}(B^1)^{-1}R^1 \quad \dots \quad c_B^{2K} - c_B^{1K}(B^K)^{-1}R^K] Q^{-1} \end{aligned} \quad (3.27)$$

$$\pi^1 = \begin{bmatrix} c_B^{11}(B^1)^{-1} - [c_B^{21} - c_B^{11}(B^1)^{-1}R^1 \quad \dots \quad c_B^{2K} - c_B^{1K}(B^K)^{-1}R^K] Q^{-1}P^1(B^1)^{-1} : \dots : c_B^{1K}(B^K)^{-1} + \\ -[c_B^{21} - c_B^{11}(B^1)^{-1}R^1 \quad \dots \quad c_B^{2K} - c_B^{1K}(B^K)^{-1}R^K] Q^{-1}P^K(B^K)^{-1} \end{bmatrix}$$

Na expressão acima, substituindo (3.27) tem-se:

$$\pi^1 = [c_B^{11}(B^1)^{-1} - \pi^2 P^1(B^1)^{-1} \quad \dots \quad c_B^{1K}(B^K)^{-1} - \pi^2 P^K(B^K)^{-1}]$$

$$\pi^1 = \left\{ \begin{array}{l} \pi^{11} = (c_B^{11} - \pi^2 P^1)(B^1)^{-1} \\ \vdots \\ \pi^{1K} = (c_B^{1K} - \pi^2 P^K)(B^K)^{-1} \end{array} \right\}. \quad (3.28)$$

A estrutura da base \bar{B} pode também ser utilizada eficientemente para atualização de uma coluna, isto é, $y = \bar{B}^{-1}\bar{a}_j$, onde \bar{a}_j é uma coluna não básica de (3.18)- (3.19). Então, o segundo sistema que resolve o simplex particionado resulta em:

$$\begin{bmatrix} y^1 \\ y^2 \\ y^3 \end{bmatrix} = \left[\begin{array}{ccc|ccc} (B^1)^{-1} + (B^1)^{-1}R^1Q^{-1}P^1(B^1)^{-1} & \dots & (B^1)^{-1}R^1Q^{-1}P^K(B^K)^{-1} & -(B^1)^{-1}R^1Q^{-1} & 0 & \alpha^1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ (B^K)^{-1}R^KQ^{-1}P^1(B^1)^{-1} & \dots & (B^K)^{-1} + (B^K)^{-1}R^KQ^{-1}P^K(B^K)^{-1} & -(B^K)^{-1}R^KQ^{-1} & 0 & \alpha^K \\ \hline -Q^{-1}P^1(B^1)^{-1} & \dots & -Q^{-1}P^K(B^K)^{-1} & Q^{-1} & 0 & \beta \\ \hline VQ^{-1}P^1(B^1)^{-1} - S^1(B^1)^{-1} & \dots & VQ^{-1}P^K(B^K)^{-1} - S^K(B^K)^{-1} & -VQ^{-1} & I & \gamma \end{array} \right]$$

Assim, resolvendo o sistema por substituição, obtém-se:

$$y^2 = -Q^{-1}P^K(B^K)^{-1}\alpha^K + Q^{-1}\beta^K = Q^{-1}[\beta^K - P^K(B^K)^{-1}\alpha^K]$$

ou sob a forma

$$y^2 = \begin{bmatrix} y^{2l} \\ \vdots \\ y^{2K} \end{bmatrix} = Q^{-1} \left\{ \begin{bmatrix} \beta^l \\ \vdots \\ \beta^K \end{bmatrix} - (P^l / \dots / P^K) \begin{bmatrix} (B^l)^{-1} & & \\ & \ddots & \\ & & (B^K)^{-1} \end{bmatrix} \begin{bmatrix} \alpha^l \\ \vdots \\ \alpha^K \end{bmatrix} \right\} = Q^{-1} \begin{bmatrix} \beta^l - P^l (B^l)^{-1} \alpha^l \\ \vdots \\ \beta^K - P^K (B^K)^{-1} \alpha^K \end{bmatrix} \quad (3.29)$$

Para obter y^l faz-se necessária a substituição de (3.29) (relativo aos arcos complementares das restrições ativas de agregação), logo

$$\begin{aligned} y^l &= (B^K)^{-1} \alpha^K + (B^K)^{-1} R^K Q^{-1} P^K (B^K)^{-1} \alpha^K - (B^K)^{-1} R^K Q^{-1} \alpha^K \\ &= (B^K)^{-1} \alpha^K - (B^K)^{-1} R^K Q^{-1} [\beta^K - P^K (B^K)^{-1} \alpha^K] \\ &= (B^K)^{-1} \alpha^K - (B^K)^{-1} R^K y^2 \end{aligned}$$

ou sob a forma

$$\begin{aligned} y^l &= \begin{bmatrix} y^{ll} \\ \vdots \\ y^{lK} \end{bmatrix} = \begin{bmatrix} (B^l)^{-1} & & \\ & \ddots & \\ & & (B^K)^{-1} \end{bmatrix} \begin{bmatrix} \alpha^l \\ \vdots \\ \alpha^K \end{bmatrix} - \begin{bmatrix} (B^l)^{-1} & & \\ & \ddots & \\ & & (B^K)^{-1} \end{bmatrix} \begin{bmatrix} R^l & & \\ & \ddots & \\ & & R^K \end{bmatrix} \begin{bmatrix} y^{2l} \\ \vdots \\ y^{2K} \end{bmatrix} \\ &= \begin{bmatrix} (B^l)^{-1} \alpha^l - (B^l)^{-1} R^l y^{2l} \\ \vdots \\ (B^K)^{-1} \alpha^K - (B^K)^{-1} R^K y^{2K} \end{bmatrix}. \end{aligned} \quad (3.30)$$

Para obter y^3 faz-se necessária a substituição de (3.29) e (3.30), portanto:

$$\begin{aligned} y^3 &= VQ^{-1} P^K (B^K)^{-1} \alpha^K - S^K (B^K)^{-1} \alpha^K - VQ^{-1} \beta^K + \gamma^K \\ &= -VQ^{-1} [\beta^K - P^K (B^K)^{-1} \alpha^K] + \gamma^K - S^K (B^K)^{-1} \alpha^K \end{aligned}$$

que, pode ser reescrita sob a forma

$$\begin{aligned} y^3 &= -Vy^2 + \gamma^K - S^K (B^K)^{-1} \alpha^K \\ &= - \left[U^l - S^l (B^l)^{-1} R^l \quad \dots \quad U^K - S^K (B^K)^{-1} R^K \right] \begin{bmatrix} y^{2l} \\ \vdots \\ y^{2K} \end{bmatrix} + \begin{bmatrix} \gamma^l - S^l (B^l)^{-1} \alpha^l \\ \vdots \\ \gamma^K - S^K (B^K)^{-1} \alpha^K \end{bmatrix} \\ &= - \begin{bmatrix} U^l y^{2l} - S^l (B^l)^{-1} R^l y^{2l} \\ \vdots \\ U^K y^{2K} - S^K (B^K)^{-1} R^K y^{2K} \end{bmatrix} + \begin{bmatrix} \gamma^l - S^l (B^l)^{-1} \alpha^l \\ \vdots \\ \gamma^K - S^K (B^K)^{-1} \alpha^K \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
&= \begin{bmatrix} \gamma^l - S^l (B^l)^{-1} \alpha^l - U^l y^{2l} + S^l (B^l)^{-1} R^l y^{2l} \\ \vdots \\ \gamma^K - S^K (B^K)^{-1} \alpha^K - U^K y^{2K} + S^K (B^K)^{-1} R^K y^{2K} \end{bmatrix} \\
&= \begin{bmatrix} \gamma^l - U^l y^{2l} - S^l [(B^l)^{-1} \alpha^l - (B^l)^{-1} R^l y^{2l}] \\ \vdots \\ \gamma^K - U^K y^{2K} - S^K [(B^K)^{-1} \alpha^K - (B^K)^{-1} R^K y^{2K}] \end{bmatrix} \\
&= \begin{bmatrix} \gamma^l - U^l y^{2l} - S^l y^{1l} \\ \vdots \\ \gamma^K - U^K y^{2K} - S^K y^{1K} \end{bmatrix}. \tag{3.31}
\end{aligned}$$

Através das expressões (3.27)-(3.30) que envolvem a inversa da matriz ciclo Q , o algoritmo abaixo pode ser enunciado.

3.4 ALGORITMO PARTICIONADO PRIMAL PARA MP

3.4.1 Estrutura Geral do Algoritmo

O algoritmo com particionamento primal para múltiplos produtos baseia-se no método simplex, que apresenta os seguintes passos básicos:

- P1. Obtenção de uma solução inicial viável.
- P2. Determinação das variáveis duais.
- P3. Determinação da variável candidata a entrar na base. Se existir, ir ao P4. Em caso contrário, Parar, pois o problema não tem solução viável.
- P4. Teste de razão para determinação da variável candidata a sair da base.
- P5. Realização da troca de base e atualização da solução. Retornar ao P2.

A seguir este algoritmo será detalhado para o caso particular do problema (3.17) que pode ser reescrito da forma

ALGORITMO 3.2 Algoritmo Simplex Primal Particionado para o problema MP

P1. Uma solução básica viável $\begin{bmatrix} x^B \\ \vdots \\ x^N \end{bmatrix}$ com uma base inicial assumindo a forma \bar{B} (com partições R^1, \dots, R^k vazias) e colunas não básicas N deve ser obtida. Arco raiz deve ser adicionado a $Ax^k = r^k$ e variáveis de folgas devem ser adicionadas a $\sum_k D^k x^k \leq b$ a fim de obter uma solução inicial viável. Os passos do algoritmo particionado são dados a seguir:

P2. *Determinação da variável candidata a entrar na base*

Para verificar a otimalidade de uma solução básica corrente, são definidos dois conjuntos de índices:

$$\Psi_1 = \{i/x_i^N = l_i^N \quad e \quad c^B \bar{B}^{-1} N(i) - c_i^N > 0\}$$

e

$$\Psi_2 = \{i/x_i^N = u_i^N \quad e \quad c^B \bar{B}^{-1} N(i) - c_i^N < 0\}$$

Se $\Psi_1 \cup \Psi_2 = \emptyset$, o algoritmo termina com a solução ótima. Caso contrário seleciona-se uma variável de índice $k \in \Psi_1 \cup \Psi_2$ para entrar na base.

$$\text{Assim, } \delta \leftarrow \begin{cases} 1, & \text{se } k \in \Psi_1 \\ -1, & \text{se } k \in \Psi_2 \end{cases}.$$

No cálculo de $c^B \bar{B}^{-1} N(i) - c_i^N$ determina-se $\pi = c^B \bar{B}^{-1}$. Utilizando π obtém-se $\pi N(i) - c_i^N$. Para o problema de múltiplos produtos MP , π é particionado em $[\pi^{11}, \dots, \pi^{1K} \quad \vdots \quad \pi^2 \quad \vdots \quad \pi^3]$ correspondente à partição (3.24) onde $\pi^3 = 0$. Utiliza-se (3.27) para obter π^2 e (3.28) para obter $\pi^1 = [\pi^{11}, \dots, \pi^{1K}]$.

P3. *Teste de razão para determinação da variável candidata a sair da base*

No método simplex, para garantir a manutenção da viabilidade da solução, calcula-se, para uma variável x_k candidata a entrar na base, o vetor auxiliar $y = \bar{B}^{-1} N(k)$ para realizar o seguinte teste de razão:

$$\Delta_1 \leftarrow \min_{\sigma(y_j)=\delta} \left\{ \frac{x_j^B - l_j^B}{|y_j|}, \infty \right\}$$

$$\Delta_2 \leftarrow \min_{-\sigma(y_j)=\delta} \left\{ \frac{u_j^B - x_j^B}{|y_j|}, \infty \right\}$$

$$\Delta \leftarrow \min\{\Delta_1, \Delta_2, u_k^N - l_k^N\}$$

Para o problema MP , y é particionado em

$$y = \begin{bmatrix} y^1 \\ \vdots \\ y^2 \\ \vdots \\ y^3 \end{bmatrix} = \begin{bmatrix} y^{1I} \\ \vdots \\ \frac{y^{1K}}{y^{2I}} \\ \vdots \\ \frac{y^{2K}}{y^3} \end{bmatrix} \text{ correspondente à partição apresentada em (3.22).}$$

Calcula-se (3.29) para obter $y^2 = [y^{2I}, \dots, y^{2K}]$. Em seguida, calcula-se (3.30) para obter $y^1 = [y^{1I}, \dots, y^{1K}]$ e finalmente (3.31) para obter y^3 .

P4. *Atualização dos Fluxos*

A variação a ser realizada na variável não básica escolhida determina uma nova solução para o problema e que, no caso de $\Delta > 0$ (solução não degenerada), é melhor que a solução anteriormente conhecida. Portanto, faz-se:

$x_k^N \leftarrow x_k^N + \Delta\delta$, para a variável do produto k candidata a entrar na base. Na atualização das variáveis básicas em relação a cada produto k , faz-se $x^B \leftarrow x^B - \Delta\delta y$. Se $\Delta = u_k^N - l_k^N$, retorna-se para determinar uma nova variável candidata a entrar na base. Em caso contrário, segue-se com o pivoteamento.

P5. *Pivoteamento*

Para efetuar a operação de pivoteamento faz-se:

$$\Psi_3 = \{j / \sigma(y_j) = \delta \text{ e } x_j^B = 0\}$$

$$\text{E } \Psi_4 = \{j / -\sigma(y_j) = \delta \text{ e } x_j^B = u_j^B\}$$

Seleciona-se algum $l \in \Psi_3 \cup \Psi_4$. Substitui-se $\bar{B}(l)$ por $N(k)$ e todas as etapas do processo são repetidas até que se tenha uma solução ótima ou uma solução ilimitada.

As fórmulas básicas necessárias para executar o algoritmo acima, envolvem a inversa da matriz ciclo Q . A eficiente implementação para este algoritmo exige que se tenha um bom procedimento para manipular e atualizar a matriz inversa Q^{-1} . As fórmulas necessárias para atualização de Q^{-1} são encontradas em [Ken80].

3.4.2 Operações de Pivoteamento sobre a Matriz Ciclo Q

O algoritmo descrito acima considera a $\dim(Q)$ igual ao número de restrições ativas, isto é, restrições que estão saturadas. Durante a operação de pivoteamento a dimensão da matriz Q pode ser modificada, dependendo da variável que sai da base. Considerando que as variáveis do problema podem ser arcos ou folgas (e os arcos da base \bar{B} são subdivididos dentro de k árvores geradoras e ou arcos complementares) então, dependendo do tipo de variável que entra (E) e que sai (S) da base, seis casos podem ser observados:

E: folga - S: folga. A linha da matriz Q associada com a folga que entra na base é removida e substituída por uma nova linha para a folga que parte da base. A $\dim(Q)$ não é modificada.

E: folga - S: arco complementar. A linha e coluna da matriz Q associada com a folga que entra e o arco complementar que sai são removidos. A $\dim(Q)$ deve ser atualizada como $\dim(Q)-1$.

E: folga - S: arco da k -ésima árvore. O arco complementar do k -ésimo produto (j -ésimo arco complementar) é buscado para substituir a saída do arco na k -ésima árvore. Este arco complementar sempre existirá (de outro modo a base se tornaria singular). A linha e coluna da matriz Q associada com a folga que entra e o j -ésimo arco complementar são removidos. A dimensão de Q deve ser atualizada como $\dim(Q)-1$.

E: arco - S: folga. Uma nova linha associada com a folga que parte da base é adicionada à matriz Q . Para manter a não singularidade de Q uma nova coluna para o arco que entra na base é também adicionada à matriz Q o qual se torna um arco complementar. A $\dim(Q)$ deve ser atualizada para $\dim(Q)+1$.

E: arco – S: arco complementar . A coluna de Q associada com o arco que parte é removida e substituída por uma coluna correspondente ao arco que entra na base o qual se torna um arco complementar. A $\dim(Q)$ não é modificada.

E: arco – S: arco da k -ésima árvore. O arco complementar do k -ésimo produto (j -ésimo arco complementar) deve ser buscado para substituir o arco que sai. Se o arco complementar é encontrado, ele substituirá o arco que parte na k -ésima árvore e o arco que entra na base torna-se um arco complementar. Se nenhum arco complementar é encontrado então o arco que entra, irá substituir o arco que parte na k -ésima árvore. Uma das duas possibilidades descritas sempre ocorrerá senão a base se tornaria singular. A $\dim(Q)$ não é modificada.

3.4.3 Considerações sobre a Matriz Ciclo Q

A dimensão da matriz de ciclo Q é em geral, nos casos práticos, muito menor quando comparada com a dimensão total da base \bar{B} . Então, espera-se que o algoritmo se comporte de modo eficiente utilizando o simplex com particionamento primal. Por outro lado, a dimensão da matriz ciclo Q é variável, depende do número de restrições de agregação ativas de fluxo. Isto pode implicar que, no processo de atualização dos elementos dessa matriz, a cada iteração, poderá aumentar a dificuldade do algoritmo. Uma outra implicação é a manipulação com duas matrizes ciclos de tamanhos variáveis como propõem Castro e Nabona (ver [Cas96]) quando uma matriz é obtida em função das restrições de capacidade mútua nos arcos (somente coeficientes unitários) e a outra quando é obtida em função de restrições gerais.

A partir das considerações, sobre a matriz ciclo Q , surge uma proposta de aplicação do algoritmo sobre uma matriz de dimensão fixa (denominada de matriz ciclo Y). Nas próximas seções serão mostrados os resultados matemáticos obtidos em função dessa matriz de dimensão fixa bem como suas propriedades.

3.5 MATRIZ CICLO Y

Neste trabalho é proposto o algoritmo sobre a matriz de dimensão fixa diferentemente do que foi proposto por outros autores. Será mostrado como a matriz \bar{B}^{-1} pode ser expressa em função da inversa de uma matriz de dimensão não variável e como os valores duais e os valores de y 's no teste de razão são obtidos a partir da inversa dessa matriz. Será apresentada também a forma de manipulação da matriz de dimensão fixa durante as operações de pivoteamento.

Como o problema a ser resolvido é de grande porte e possui um número pequeno de restrições de capacidade agregada, caso comum, na situação em que a rede representa sistemas logísticos de distribuição de produtos compostos³, é conveniente desenvolver uma forma eficiente de manipulação da matriz ciclo, mantendo fixa sua dimensão. Portanto, o Algoritmo 3.2, ao invés de operar com a inversa da matriz ciclo Q passa operar com a inversa da matriz ciclo Y de dimensão não variável. A seguir é apresentada a forma de particionamento que possibilita aplicar o Algoritmo 3.2 a partir de uma matriz ciclo de dimensão não variável.

A estrutura da matriz ciclo Y , apresentada na expressão (3.23), possui as colunas da matriz composta pelos arcos complementares e folgas. As linhas correspondem às restrições ativas e não ativas de agregação. Portanto, computar Y é obter as partições representadas pelas matrizes Q , V , θ e I .

O mesmo procedimento utilizado na obtenção da matriz ciclo Q é utilizado para gerar a matriz V , onde V é a matriz ciclo gerada a partir dos arcos complementares associados às restrições de agregação não ativas. Portanto, a matriz ciclo V pode ser obtida a partir da árvore básica, R^k e dos conjuntos indexados.

Seja $v^k = \{e_{j_1}, \dots, e_{j_k}\} \subset \mathcal{R}$ o conjunto indexado correspondente às colunas de R^k . O conjunto indexado v^k é ordenado tal que e_{j_s} denota o arco correspondente a s -ésima coluna de R^k .

³ Em problemas reais, apenas alguns arcos do sistema de distribuição apresentam restrições de capacidade relevantes. Na maioria dos arcos que representam modais (rodovias, ferrovias e hidrovias) tais restrições de capacidade são irrelevantes. De fato, em sistemas logísticos, apenas arcos específicos associados à estrutura física da empresa costumam apresentar limitantes de capacidade que precisam ser consideradas. Em geral, tais arcos representam uma parcela muito pequena em relação ao número total de arcos da rede.

Seja um conjunto $\overline{\Omega}^k = \{e_{m_1}, \dots, e_{m_{\bar{r}}}\} \subset \mathcal{R}$ indexado correspondente as \bar{r} -ésimas linhas de $[U^1, \dots, U^k]$. Para cada $e_{m_{\bar{r}}} \in \overline{\Omega}^k$ existe uma linha indexada \bar{r} tal que o arco $e_{m_{\bar{r}}}$ denota o arco correspondente a \bar{r} -ésima linha. Cabe salientar que a linha indexada \bar{r} corresponde a uma restrição não ativa de agregação que poderá unir arcos de um mesmo produto ou de diferentes produtos.

Se $S_{\bar{r}}^k$ denota a \bar{r} -ésima linha de S^k e β^s denota a s -ésima coluna de $(B^k)^{-1}R^k$.

Então cada elemento $V_{\bar{r}s}^k$ pode ser determinado da seguinte forma:

$$V_{\bar{r}s}^k = U_{\bar{r}s}^k - S_{\bar{r}}^k \beta^s \quad (3.32)$$

onde para um produto indexado k , $U_{\bar{r}s}^k$ indica o coeficiente do arco complementar na restrição não ativa de agregação, com $m_{\bar{r}} = j_s$, onde $e_{m_{\bar{r}}} \in \overline{\Omega}^k$ e $e_{j_s} \in \nu^k$. O produto matricial $S_{\bar{r}}^k \beta^s$ indica o somatório do produto dos coeficientes dos arcos básicos na restrição não ativa de agregação pela sua orientação na seqüência P_s , ou seja, com $e_{m_{\bar{r}}}$ um arco da t -ésima coluna de B^k e $e_{m_{\bar{r}}} \in P_s$ (com orientação normal ou reversa) onde $e_{m_{\bar{r}}} \in \overline{\Omega}^k$.

Observa-se que, a forma de computar a matriz V é análoga a forma de computar Q .

As partições correspondentes as matrizes O e I são facilmente computadas, pois estão diretamente relacionadas com as variáveis de folga não básicas e básicas. A partição nula é devido ao fato das restrições serem ativas e as folgas correspondentes não se encontrarem na base. Já a partição definida pela matriz I é formada tendo em vista os coeficientes das variáveis de folga das restrições não ativas encontrarem-se na base.

A estrutura do Algoritmo 3.2 é mantida, diferindo apenas na forma de obtenção das variáveis duais e dos valores y 's no teste de razão.

Como se pretende manipular com uma matriz de dimensão fixa, o produto $\overline{B}W$ apresentado em (3.25) pode ser particionado da seguinte forma:

$$\overline{B}W = \left[\begin{array}{ccc|cc} B^1 & & & & \\ & \ddots & & & \\ & & B^K & & \\ \hline P^1 & \dots & P^K & Q & O \\ S^1 & \dots & S^K & V & I \end{array} \right]$$

$$\bar{B}W = \left[\begin{array}{ccc|c} B^1 & & & \\ & \ddots & & \\ & & B^K & \\ \hline D_B^1 & \cdots & D_B^K & Y \end{array} \right] \quad (3.33)$$

onde

$$[D_B^K] = \left[\begin{array}{c} P^K \\ S^K \end{array} \right].$$

A matriz \bar{B}^{-1} obtida em (3.26) pode ser reescrita em função da inversa da matriz ciclo Y como:

$$\left[\begin{array}{ccc|c} (B^1)^{-1} + (B^1)^{-1} C^1 Y^{-1} D^1 (B^1)^{-1} & (B^1)^{-1} C^1 Y^{-1} D^2 (B^2)^{-1} & \cdots & (B^1)^{-1} C^1 Y^{-1} D^K (B^K)^{-1} \\ \hline (B^2)^{-1} C^2 Y^{-1} D^1 (B^1)^{-1} & (B^2)^{-1} + (B^2)^{-1} C^2 Y^{-1} D^2 (B^2)^{-1} & \cdots & (B^2)^{-1} C^2 Y^{-1} D^K (B^K)^{-1} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline (B^K)^{-1} C^K Y^{-1} D^1 (B^1)^{-1} & (B^K)^{-1} C^K Y^{-1} D^2 (B^2)^{-1} & \cdots & (B^K)^{-1} + (B^K)^{-1} C^K Y^{-1} D^K (B^K)^{-1} \\ \hline -Y^{-1} D^1 (B^1)^{-1} & -Y^{-1} D^2 (B^2)^{-1} & \cdots & -Y^{-1} D^K (B^K)^{-1} \end{array} \right] \quad (3.34)$$

No anexo A.2 é apresentada a forma produto da inversa utilizada no processo da inversão da matriz ciclo Y . A utilização da forma produto é indicada devido à grande esparsidade da matriz ciclo Y podendo armazená-la em forma de listas através da memorização dos elementos não nulos. A técnica de inversão da matriz é detalhada em [May84].

3.6 OBTENÇÃO DOS VALORES DUAIS UTILIZANDO A INVERSA DA MATRIZ CICLO Y

O sistema que determina as variáveis duais, considerando a matriz ciclo proposta Y , passa ter o seguinte particionamento:

$$[\pi^1 \mid \pi^2] = [c_1^B \mid c_2^B] \bar{B}^{-1}.$$

Para solucionar o sistema, as partições c_B^1 e c_B^2 que representam os custos para os arcos básicos passam a ser denotados por:

$$c_B^1 = [c_B^{11} \mid c_B^{12} \mid \cdots \mid c_B^{1K}] , \text{ indicando os custos associados aos arcos básicos } e;$$

$$c_B^2 = [c_B^{21} \mid c_B^{22} \mid \cdots \mid c_B^{2K} \mid 0] , \text{ indicando os custos associados aos arcos complementares } e_{js} \text{ e folgas } S.$$

A equação (3.27) pode ser reescrita da forma:

$$\pi^2 = [c_B^{2l} - c_B^{1l} (B^l)^{-1} R^l \mid \cdots \mid c_B^{2K} - c_B^{1K} (B^K)^{-1} R^K \mid 0] Y^{-1}. \quad (3.35)$$

As equações (3.28) podem ser expressas através de (3.35) da seguinte forma:

$$\pi^l = \left\{ \begin{array}{l} \pi^{1l} = (c_B^{2l} - \pi^2 D_B^l) (B^l)^{-1} \\ \vdots \\ \pi^{lk} = (c_B^{2K} - \pi^2 D_B^K) (B^K)^{-1} \end{array} \right\} \quad (3.36)$$

ou sob forma

$$\pi^l = \left\{ \begin{array}{l} \pi^{1l} = \theta_l (B^l)^{-1} \\ \vdots \\ \pi^{lk} = \theta_k (B^K)^{-1} \end{array} \right\}, \text{ onde } \theta_k = (c^{1k} - \pi^2 D_B^k). \quad (3.37)$$

3.7 OBTENÇÃO DOS VALORES Y'S NO TESTE DE RAZÃO UTILIZANDO A INVERSA DA MATRIZ CICLO Y

Os preços π^2 e π^3 foram armazenados em um único vetor quando nas operações matriciais foi utilizada a inversa da matriz ciclo proposta Y. O mesmo ocorre na determinação de y^2 e y^3 necessários no cálculo do teste de razão.

As expressões (3.29) e (3.31), podem ser obtidas da seguinte forma:

$$\begin{bmatrix} y^2 \\ \vdots \\ y^3 \end{bmatrix} = \begin{bmatrix} Q^{-1} & \mid & 0 \\ \hline -VQ^{-1} & \mid & I \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \beta^l \\ \vdots \\ \beta^K \end{bmatrix} - \begin{bmatrix} P^l (B^l)^{-1} \alpha^l \\ \vdots \\ P^K (B^K)^{-1} \alpha^K \end{bmatrix} \\ \hline \begin{bmatrix} \gamma^l \\ \vdots \\ \gamma^K \end{bmatrix} - \begin{bmatrix} S^l (B^l)^{-1} \alpha^l \\ \vdots \\ S^K (B^K)^{-1} \alpha^K \end{bmatrix} \end{bmatrix} \quad (3.38)$$

As diferenças $[\beta^k - P^k (B^k)^{-1} \alpha^k]$ e $[\gamma^k - S^k (B^k)^{-1} \alpha^k]$ que aparecem em (3.38) correspondem à partição das restrições ativas e não ativas de agregação, podendo ser unicamente representadas por $[\mu^K - D^K (B^K)^{-1} \alpha^K]$,

$$\text{onde } \mu^k = \begin{bmatrix} \beta^k \\ \dots \\ \gamma^k \end{bmatrix} \text{ e } D^k = \begin{bmatrix} P^k \\ \dots \\ S^k \end{bmatrix}.$$

Portanto, computar o vetor $\begin{bmatrix} y^2 \\ \dots \\ y^3 \end{bmatrix}$ é obter os valores y 's no teste de razão

correspondente aos arcos complementares e às folgas que se encontram na base.

A equação (3.38) pode ser reescrita como

$$y^2 = Y^{-1}[\mu^k - D^k(B^k)^{-1}\alpha^k]. \quad (3.39)$$

O principal detalhe da implementação do algoritmo simplex especializado, que utiliza a inversa da matriz ciclo Y , está baseado nas operações de pivoteamento.

3.8 OPERAÇÕES DE PIVOTEAMENTO SOBRE A MATRIZ CICLO Y

Tendo em vista que o algoritmo não trabalha com toda base \bar{B} , é necessário refletir como a escolha da base afeta as k árvores geradoras e a matriz ciclo Y . As variáveis do problema podem ser arcos ou folgas, de modo que, os arcos da base \bar{B} podem ser subdivididos dentro de k árvores geradoras e arcos complementares.

Os casos considerados para o tipo de variável que entra e sai da base foram apresentados anteriormente, com referência na matriz ciclo Q . A idéia agora é apresentar os mesmos casos considerados com referência na matriz ciclo Y , que durante as operações de pivoteamento permanece com dimensão fixa igual ao número de restrições agregadas.

Os diversos casos serão mostrados através de um exemplo considerando dois produtos K_1 e K_2 , uma base genérica para cada um dos produtos e uma matriz ciclo de dimensão fixa com índices sc_i representando as restrições de acoplamento. As operações de pivoteamento sobre matriz ciclo Y , bem como a representação das bases serão realizadas sobre uma dada iteração.

As figuras III-5 e III-6 representam as bases genéricas para os produtos K_1 e K_2 , com x_d representando o arco que entra na base na iteração corrente.

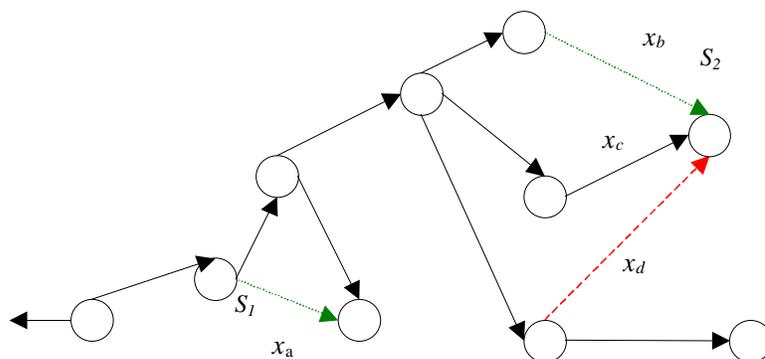


Figura III-5 – Base genérica para o produto K_1 contendo dois arcos complementares. Nesta figura os arcos x_a e x_b estão associados às folgas S_1 e S_2 respectivamente, onde x_d representa o arco candidato a entrar na base em uma iteração qualquer.

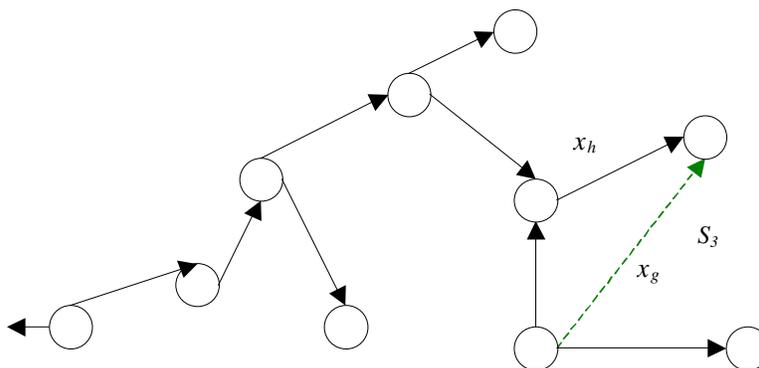


Figura III-6 – Base genérica para o produto K_2 contendo um arco complementar. Nesta figura o arco x_g é associado à folga S_3 .

Consideram-se, inicialmente, os casos em que a variável candidata a entrar na base é um arco. No exemplo proposto, os casos serão mostrados sobre o arco x_d candidato a entrar na base em relação ao produto K_1 .

Caso 1 Entra: arco – Sai: Folga

Nesse caso, a linha da matriz ciclo Y , associada com a folga que parte da base é ativada. A coluna para o arco que entra, o qual se torna um arco complementar fica associado à folga que parte da base. A coluna modificada é a coluna cuja folga pertencia à restrição não ativa antes do pivoteamento.

A figura III-7 representa a nova base para o produto K_1 depois do pivoteamento. Para exemplificar, a folga S_5 pertencente à restrição sc_5 foi ativada pela entrada do arco x_d . As matrizes abaixo representam a matriz ciclo antes e depois do pivoteamento para o caso 1.

$$Y = \begin{array}{cccc|cccc}
 s_1 & s_3 & s_2 & & & & & \\
 x_a & S_4 & x_g & x_b & S_5 & \cdots & S_n & \\
 \times & 0 & \times & \times & 0 & \cdots & 0 & SC_1 \\
 \times & 0 & \times & \times & 0 & \cdots & 0 & SC_2 \\
 \times & 0 & \times & \times & 0 & \cdots & 0 & SC_3 \\
 \times & 1 & \times & \times & 0 & \cdots & 0 & SC_4 \\
 \times & 0 & \times & \times & 1 & \cdots & 0 & SC_5 \\
 \cdots & \vdots \\
 \times & 0 & \times & \times & 0 & \cdots & 1 & SC_n
 \end{array}$$

Matriz Y antes do pivoteamento

$$Y = \begin{array}{cccc|cccc}
 s_1 & s_3 & s_2 & s_5 & & & & \\
 x_a & S_4 & x_g & x_b & x_d & \cdots & S_n & \\
 \times & 0 & \times & \times & \times & \cdots & 0 & SC_1 \\
 \times & 0 & \times & \times & \times & \cdots & 0 & SC_2 \\
 \times & 0 & \times & \times & \times & \cdots & 0 & SC_3 \\
 \times & 1 & \times & \times & \times & \cdots & 0 & SC_4 \\
 \times & 0 & \times & \times & \times & \cdots & 0 & SC_5 \\
 \cdots & \cdots \\
 \times & 0 & \times & \times & \times & \cdots & 1 & SC_n
 \end{array}$$

Matriz Y depois do pivoteamento

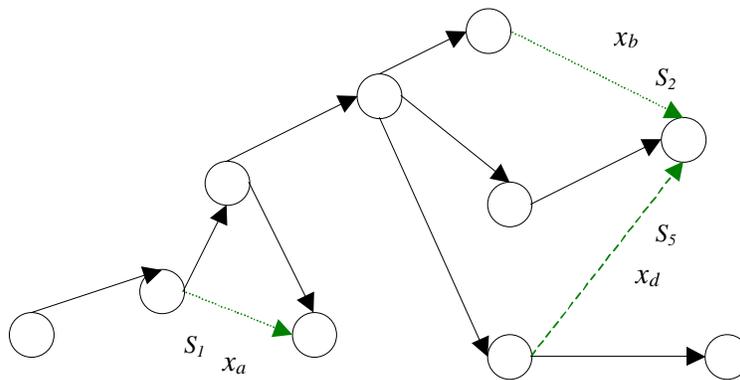


Figura III-7 – Representação da nova base para o produto K_1 após pivoteamento sobre o caso 1. Nesta figura o arco x_d torna-se complementar associado à folga S_5 .

Caso 2 *Entra*: arco – *Sai*: arco da k -ésima árvore.

Nesse caso podem ocorrer duas possibilidades:

- O arco complementar do k -ésimo produto (o j -ésimo arco complementar) é buscado porque o caminho P_j é desconexo pelo arco que sai; esse arco substituirá o arco que parte na k -ésima árvore e o arco que entra torna-se um arco complementar.
- O arco que entra substitui o arco que parte na k -ésima árvore.

Uma das duas possibilidades descritas acima sempre ocorrerá, senão matriz básica associada se tornaria singular.

A figura III-8 mostra o caso 2a, representando a nova base para o produto K_1 depois do pivoteamento. O arco x_b substitui o arco que parte da árvore básica, pois o caminho P_j é

desconexo pela saída do arco x_c . A matriz ciclo antes e depois do pivoteamento é mostrada abaixo.

$$Y = \begin{array}{c} \left[\begin{array}{cccc|ccc} s_1 & & s_3 & s_2 & & & \\ x_a & S_4 & x_g & x_b & S_5 & \cdots & S_n \\ \times & 0 & \times & \times & 0 & \cdots & 0 \\ \hline \times & 0 & \times & \times & 0 & \cdots & 0 \\ \hline \times & 0 & \times & \times & 0 & \cdots & 0 \\ \times & 1 & \times & \times & 0 & \cdots & 0 \\ \times & 0 & \times & \times & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \times & 0 & \times & \times & 0 & \cdots & 1 \end{array} \right] \begin{array}{l} SC_1 \\ SC_2 \\ SC_3 \\ SC_4 \\ SC_5 \\ \vdots \\ SC_n \end{array} \\ \text{Matriz } Y \text{ antes do pivoteamento} \end{array}$$

$$Y = \begin{array}{c} \left[\begin{array}{cccc|ccc} s_1 & & s_3 & s_2 & & & \\ x_a & S_4 & x_g & x_d & S_5 & \cdots & S_n \\ \times & 0 & \times & \times & 0 & \cdots & 0 \\ \hline \times & 0 & \times & \times & 0 & \cdots & 0 \\ \hline \times & 0 & \times & \times & 0 & \cdots & 0 \\ \times & 1 & \times & \times & 0 & \cdots & 0 \\ \times & 0 & \times & \times & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \times & 0 & \times & \times & 0 & \cdots & 1 \end{array} \right] \begin{array}{l} SC_1 \\ SC_2 \\ SC_3 \\ SC_4 \\ SC_5 \\ \vdots \\ SC_n \end{array} \\ \text{Matriz } Y \text{ depois do pivoteamento} \end{array}$$

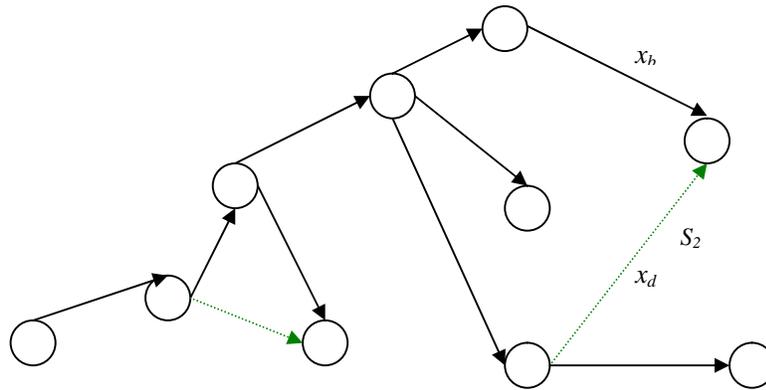


Figura III-8 – Representação da nova base para o produto K_1 após pivoteamento sobre o caso 2a (entra e sai arco de um mesmo produto). Nesta figura, o arco x_d torna-se complementar associado à folga S_2 e o arco x_c sai da base.

Um outro exemplo para o caso 2a é aquele em que o arco que sai não pertence a k -ésima árvore básica do arco que entra.

As figuras III-9 e III-10 representam a saída do arco x_h quando o arco x_d entra na base. A matriz ciclo antes e depois do pivoteamento é representada abaixo. Pode ser observado que no caso 2a as linhas ativas da matriz após pivoteamento permanecem as mesmas, apenas a coluna do arco complementar que substitui o arco que sai é modificada pelo arco que entra.

$$Y = \begin{array}{c|cccc|cccc} s_1 & & s_3 & s_2 & & & & \\ x_a & S_4 & x_g & x_b & S_5 & \cdots & S_n & \\ \times & 0 & \times & \times & 0 & \cdots & 0 & SC_1 \\ \times & 0 & \times & \times & 0 & \cdots & 0 & SC_2 \\ \times & 0 & \times & \times & 0 & \cdots & 0 & SC_3 \\ \times & 1 & \times & \times & 0 & \cdots & 0 & SC_4 \\ \times & 0 & \times & \times & 1 & \cdots & 0 & SC_5 \\ \cdots & \vdots \\ \times & 0 & \times & \times & 0 & \cdots & 1 & SC_n \end{array}$$

Matriz Y antes do pivoteamento

$$Y = \begin{array}{c|cccc|cccc} s_1 & & s_3 & s_2 & & & & \\ x_a & S_4 & x_d & x_b & S_5 & \cdots & S_n & \\ \times & 0 & \times & \times & 0 & \cdots & 0 & SC_1 \\ \times & 0 & \times & \times & 0 & \cdots & 0 & SC_2 \\ \times & 0 & \times & \times & 0 & \cdots & 0 & SC_3 \\ \times & 1 & \times & \times & 0 & \cdots & 0 & SC_4 \\ \times & 0 & \times & \times & 1 & \cdots & 0 & SC_5 \\ \cdots & \vdots \\ \times & 0 & \times & \times & 0 & \cdots & 1 & SC_n \end{array}$$

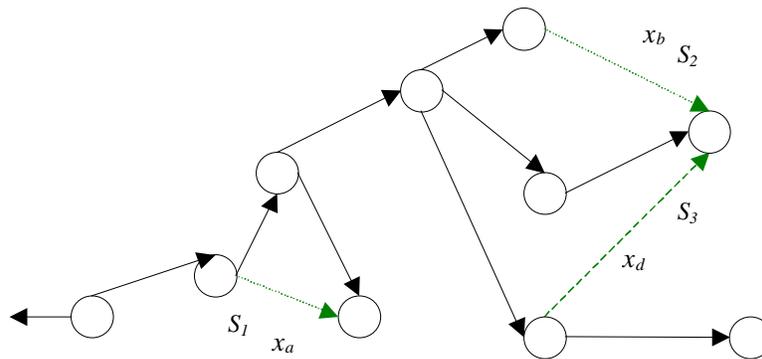
Matriz Y depois do pivoteamento

Figura III-9 – Representação da nova base para o produto K_1 após pivoteamento sobre o caso 2a (entra e sai arco de produtos diferentes). Nesta figura, o arco x_d torna-se complementar associado à folga S_3 .

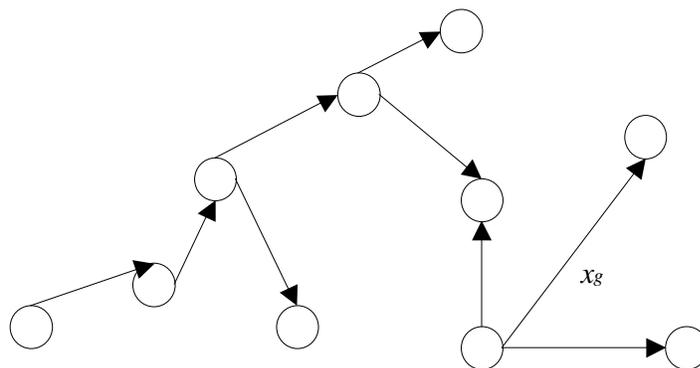


Figura III-10 – Representação da nova base para o produto K_2 após pivoteamento sobre o caso 2a (entra e sai arco de produtos diferentes). Nesta figura, o arco x_g torna-se arco da árvore e o arco x_h sai da base.

No caso 2b, nenhum caminho P_j é desconexo pela saída do arco da k -ésima árvore. Apenas o arco que entra substitui o arco que sai na k -ésima árvore. Neste caso a forma da matriz Y permanece a mesma, mas a coluna correspondente à troca de base deve ser recalculada.

Caso 3 *Entra*: arco – *Sai*: arco complementar.

A coluna da matriz ciclo Y associada ao arco que parte é substituída pelo arco que entra e o arco que entra torna-se complementar.

A figura III-11 representa a nova base para o produto K_1 após pivoteamento. A representação da matriz ciclo antes e depois do pivoteamento é mostrada abaixo.

$$Y = \begin{array}{c} \left[\begin{array}{cccc|cccc} s_1 & & s_3 & s_2 & & & & \\ x_a & S_4 & x_g & x_b & S_5 & \dots & S_n & \\ \times & 0 & \times & \times & 0 & \dots & 0 & \\ \hline \times & 0 & \times & \times & 0 & \dots & 0 & \\ \hline \times & 0 & \times & \times & 0 & \dots & 0 & \\ \times & 1 & \times & \times & 0 & \dots & 0 & \\ \times & 0 & \times & \times & 1 & \dots & 0 & \\ \dots & \\ \times & 0 & \times & \times & 0 & \dots & 1 & \end{array} \right] \begin{array}{l} sc_1 \\ sc_2 \\ sc_3 \\ sc_4 \\ sc_5 \\ \vdots \\ sc_n \end{array} \\ \text{Matriz } Y \text{ antes do pivoteamento} \end{array}$$

$$Y = \begin{array}{c} \left[\begin{array}{cccc|cccc} s_1 & & s_3 & s_2 & & & & \\ x_a & S_4 & x_g & x_d & S_5 & \dots & S_n & \\ \times & 0 & \times & \times & 0 & \dots & 0 & \\ \hline \times & 0 & \times & \times & 0 & \dots & 0 & \\ \hline \times & 0 & \times & \times & 0 & \dots & 0 & \\ \times & 1 & \times & \times & 0 & \dots & 0 & \\ \times & 0 & \times & \times & 1 & \dots & 0 & \\ \dots & \\ \times & 0 & \times & \times & 0 & \dots & 1 & \end{array} \right] \begin{array}{l} sc_1 \\ sc_2 \\ sc_3 \\ sc_4 \\ sc_5 \\ \vdots \\ sc_n \end{array} \\ \text{Matriz } Y \text{ depois do pivoteamento} \end{array}$$

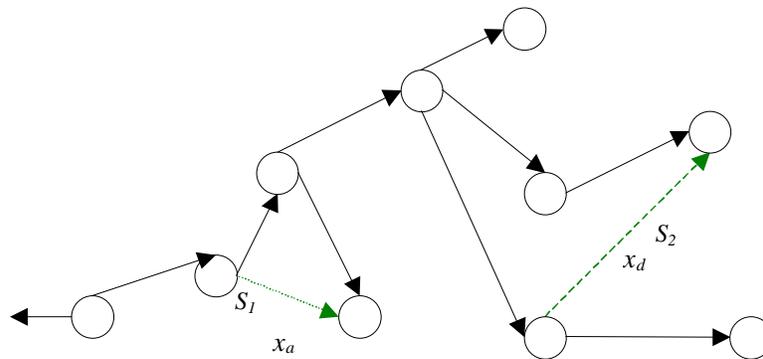


Figura III-11 – Representação da nova base para o produto K_1 após pivoteamento sobre o caso 3. Nesta figura, o arco x_d torna-se complementar associado à folga S_2 e o arco x_b sai da base.

Agora, serão considerados os casos em que a variável escolhida para entrar na base é uma variável de folga. No exemplo proposto, os casos serão mostrados sobre a folga S_2 , associada ao arco complementar x_b , candidata a entrar na base.

Caso 4 *Entra: folga - Sai: folga*

A variável de folga que parte da base substitui a folga que entra. A linha da matriz ciclo Y associada à folga que entra na base é desativada, enquanto a linha correspondente à folga que parte da base é ativada. O arco complementar após pivoteamento fica associado à folga que parte da base.

Para exemplificar, a folga S_5 relativa a restrição sc_5 foi ativada pela entrada da folga S_2 na base. A matriz ciclo antes e depois do pivoteamento é mostrada a seguir:

$$Y = \left[\begin{array}{cccc|cccc} s_1 & & s_3 & s_2 & & & & \\ x_a & S_4 & x_g & x_b & S_5 & \dots & S_n & \\ \times & 0 & \times & \times & 0 & \dots & 0 & sc_1 \\ \times & 0 & \times & \times & 0 & \dots & 0 & sc_2 \\ \times & 0 & \times & \times & 0 & \dots & 0 & sc_3 \\ \times & 1 & \times & \times & 0 & \dots & 0 & sc_4 \\ \times & 0 & \times & \times & 1 & \dots & 0 & sc_5 \\ \dots & \vdots \\ \times & 0 & \times & \times & 0 & \dots & 1 & sc_n \end{array} \right] \quad Y = \left[\begin{array}{cccc|cccc} s_1 & & s_3 & s_5 & & & & \\ x_a & S_4 & x_g & x_b & S_2 & \dots & S_n & \\ \times & 0 & \times & \times & 0 & \dots & 0 & sc_1 \\ \times & 0 & \times & \times & 1 & \dots & 0 & sc_2 \\ \times & 0 & \times & \times & 0 & \dots & 0 & sc_3 \\ \times & 1 & \times & \times & 0 & \dots & 0 & sc_4 \\ \times & 0 & \times & \times & 0 & \dots & 0 & sc_5 \\ \dots & \vdots \\ \times & 0 & \times & \times & 0 & \dots & 1 & sc_n \end{array} \right]$$

Matriz Y antes do pivoteamento Matriz Y depois do pivoteamento

Caso 5 *Entra: folga - Sai: arco complementar*

Nesse caso, a linha da matriz ciclo correspondente à folga que entra na base, é desativada, enquanto o arco complementar é removido da base, deixando de ser complementar. No entanto, duas observações devem ser feitas sobre o caso 5, em relação à folga que entra na base.

1^a) A folga que entra está associada ao arco complementar que sai.

A coluna da matriz ciclo Y associada ao arco complementar é substituída pela folga que entra. O exemplo é feito sobre a folga S_2 associada ao arco complementar x_b . A folga S_2 entra na base, enquanto o arco x_b deixa de ser complementar. A matriz ciclo antes e depois do pivoteamento, é mostrada a seguir:

$$Y = \left[\begin{array}{cccc|cccc} s_1 & & s_3 & s_2 & & & & \\ x_a & S_4 & x_g & x_b & S_5 & \cdots & S_n & \\ \times & 0 & \times & \times & 0 & \cdots & 0 & SC_1 \\ \times & 0 & \times & \times & 0 & \cdots & 0 & SC_2 \\ \times & 0 & \times & \times & 0 & \cdots & 0 & SC_3 \\ \times & 1 & \times & \times & 0 & \cdots & 0 & SC_4 \\ \times & 0 & \times & \times & 1 & \cdots & 0 & SC_5 \\ \cdots & \vdots \\ \times & 0 & \times & \times & 0 & \cdots & 1 & SC_n \end{array} \right] \quad \text{Matriz } Y \text{ antes do pivoteamento}$$

$$\left[\begin{array}{cccc|cccc} s_1 & & s_3 & & S_2 & S_5 & \cdots & S_n \\ x_a & S_4 & x_g & S_2 & S_5 & \cdots & S_n & \\ \times & 0 & \times & 0 & 0 & \cdots & 0 & SC_1 \\ \times & 0 & \times & 1 & 0 & \cdots & 0 & SC_2 \\ \times & 0 & \times & 0 & 0 & \cdots & 0 & SC_3 \\ \times & 1 & \times & 0 & 0 & \cdots & 0 & SC_4 \\ \times & 0 & \times & 0 & 1 & \cdots & 0 & SC_5 \\ \cdots & \vdots \\ \times & 0 & \times & 0 & 0 & \cdots & 1 & SC_n \end{array} \right] \quad \text{Matriz } Y \text{ depois do pivoteamento}$$

A figura III-12 representa a nova base após pivoteamento depois que o arco x_b é removido da base pela entrada da folga S_2 .

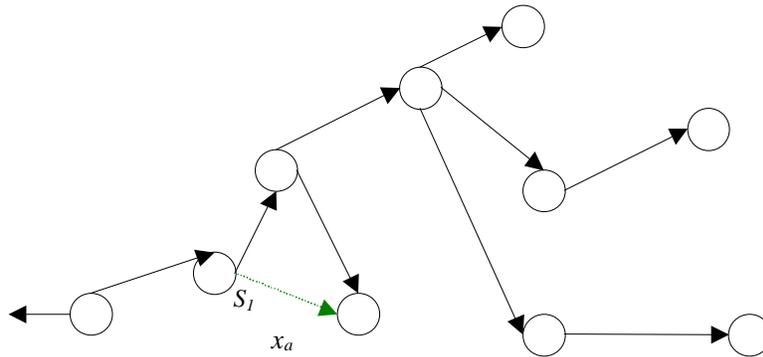


Figura III-12 – Representação da nova base para o produto K_1 após pivoteamento sobre o caso 5 (a folga que entra na base está associada ao arco complementar que sai). Nesta figura, o arco complementar x_b sai da base e a folga S_2 associada a ele entra na base.

2^a) A folga que entra na base não está associada ao arco complementar que sai da base.

A variável de folga associada ao arco complementar que parte da base associa-se ao arco complementar correspondente à folga que entra na base. A coluna da matriz ciclo associada ao arco complementar que parte da base é modificada pela folga que entra na base. A linha da matriz ciclo correspondente a folga que entra é desativada. A representação da matriz ciclo antes e depois do pivoteamento é mostrada abaixo.

$$Y = \begin{array}{c} \begin{array}{cccc|cccc} s_1 & & s_3 & s_2 & & & & \\ x_a & S_4 & x_g & x_b & S_5 & \dots & S_n & \\ \times & 0 & \times & \times & 0 & \dots & 0 & sc_1 \\ \times & 0 & \times & \times & 0 & \dots & 0 & sc_2 \\ \times & 0 & \times & \times & 0 & \dots & 0 & sc_3 \\ \times & I & \times & \times & 0 & \dots & 0 & sc_4 \\ \times & 0 & \times & \times & I & \dots & 0 & sc_5 \\ \dots & \vdots \\ \times & 0 & \times & \times & 0 & \dots & I & sc_n \end{array} \\ \text{Matriz } Y \text{ antes do pivoteamento} \end{array}$$

$$Y = \begin{array}{c} \begin{array}{cccc|cccc} s_1 & & s_3 & s_2 & & & & \\ x_a & S_4 & S_2 & x_b & S_5 & \dots & S_n & \\ \times & 0 & 0 & \times & 0 & \dots & 0 & sc_1 \\ \times & 0 & I & \times & 0 & \dots & 0 & sc_2 \\ \times & 0 & 0 & \times & 0 & \dots & 0 & sc_3 \\ \times & I & 0 & \times & 0 & \dots & 0 & sc_4 \\ \times & 0 & 0 & \times & I & \dots & 0 & sc_5 \\ \dots & \vdots \\ \times & 0 & 0 & \times & 0 & \dots & I & sc_n \end{array} \\ \text{Matriz } Y \text{ depois do pivoteamento} \end{array}$$

As figuras III-13 e III-14 representam a base para o produto K_1 tendo o arco x_b associado a folga S_3 e a base para o produto K_2 sem o arco complementar x_g que sai da base devido à folga S_2 que entra.

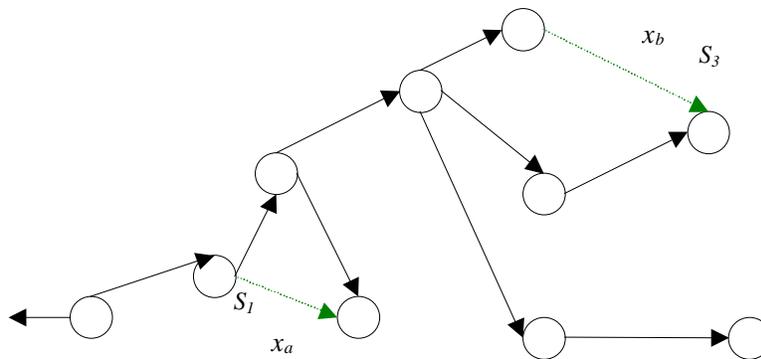


Figura III-13 – Representação da nova base para o produto K_1 após pivoteamento sobre o caso 5 (a folga que entra na base não está associada ao arco complementar que sai). Nesta figura, a folga S_2 associada ao arco x_b entra na base e o arco x_b passa a se associar à folga S_3 .

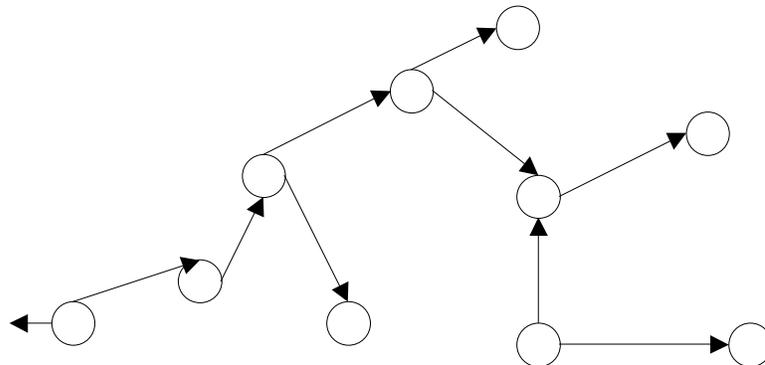


Figura III-14 – Representação da nova base para o produto K_2 após pivoteamento sobre o caso 5 (a folga que entra na base não está associada ao arco complementar que sai). Nesta figura, o arco complementar x_g associado à folga S_3 sai da base.

Caso 6) *Entra: folga – Sai: arco da k-ésima árvore.*

Neste caso, o caminho P_j é desconexo pelo arco que sai da k -ésima árvore, e o arco complementar do k -ésimo produto (o j -ésimo arco complementar) é buscado para reconectar a árvore. Este arco complementar sempre existirá (de outro modo a base se tornaria singular). A coluna da matriz Y fica associada com a folga que entra na base. A linha da matriz ciclo correspondente à folga que entra é desativada.

A figura III-15 mostra o arco complementar x_b que se torna arco da árvore para o produto K_l devido à saída do arco x_c . A matriz ciclo antes e depois do pivoteamento é mostrada abaixo.

$$Y = \left[\begin{array}{cccc|cccc} s_1 & & s_3 & s_2 & & & & \\ x_a & S_4 & x_g & x_b & S_5 & \cdots & S_n & \\ \times & 0 & \times & \times & 0 & \cdots & 0 & sc_1 \\ \hline \times & 0 & \times & \times & 0 & \cdots & 0 & sc_2 \\ \times & 0 & \times & \times & 0 & \cdots & 0 & sc_3 \\ \times & 1 & \times & \times & 0 & \cdots & 0 & sc_4 \\ \times & 0 & \times & \times & 1 & \cdots & 0 & sc_5 \\ \cdots & \vdots \\ \times & 0 & \times & \times & 0 & \cdots & 1 & sc_n \end{array} \right]$$

Matriz Y antes do pivoteamento

$$\left[\begin{array}{cccc|cccc} s_1 & & s_3 & & & & & \\ x_a & S_4 & x_g & S_2 & S_5 & \cdots & S_n & \\ \times & 0 & \times & 0 & 0 & \cdots & 0 & sc_1 \\ \times & 0 & \times & 1 & 0 & \cdots & 0 & sc_2 \\ \times & 0 & \times & 0 & 0 & \cdots & 0 & sc_3 \\ \times & 1 & \times & 0 & 0 & \cdots & 0 & sc_4 \\ \times & 0 & \times & 0 & 1 & \cdots & 0 & sc_5 \\ \cdots & \vdots \\ \times & 0 & \times & 0 & 0 & \cdots & 1 & sc_n \end{array} \right]$$

Matriz Y depois do pivoteamento

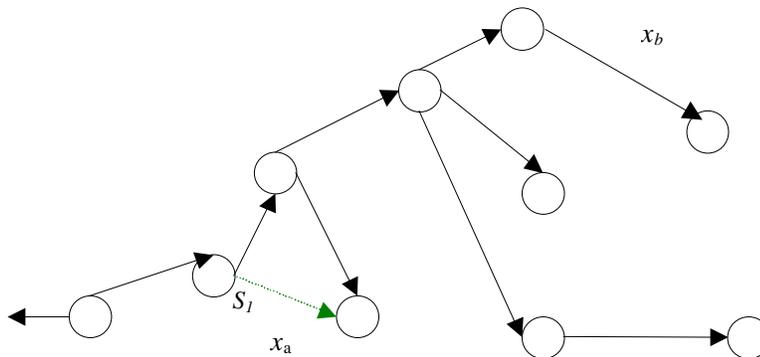


Figura III-15 – Representação da nova base para o produto K_l após pivoteamento sobre o caso 6. Nesta figura, o arco x_b torna-se um arco da árvore básica, a folga S_2 entra na base e o arco x_c sai da base.

3.9 ALGUMAS CONSIDERAÇÕES

Todas as possibilidades de pivoteamento sobre a matriz ciclo Y foram ilustradas. De certo modo, buscou-se enunciar regras para manipulação da matriz ciclo Y com as k -ésimas

árvores básicas. No entanto, a cada iteração é necessário fazer a inversão da matriz ciclo. Os conceitos e técnicas fundamentais que são relacionados com a inversão dessa matriz estão exibidos nos teoremas e definições do Anexo A.2. A técnica de inversão utilizando matrizes elementares pode ser encontrada em detalhes em [May84].

O exemplo apresentado no Apêndice I aplica todos os resultados algébricos deduzidos neste capítulo bem como ilustra as operações de pivoteamento que são realizadas a cada iteração do algoritmo.

O objetivo é mostrar passo a passo a aplicação do algoritmo primal particionado na resolução dos problemas *MP*. O método simplex primal particionado pode ser realizado diretamente no grafo manipulando com a inversa da matriz ciclo *Y*.

CAPÍTULO IV

4. MODELO DE REDE PARA O PROBLEMA DA DISTRIBUIÇÃO DE PRODUTOS COMPOSTOS

Neste capítulo será apresentado um modelo de programação linear para resolver o problema da distribuição de produtos compostos *MPC*. O processo de construção do novo modelo de rede permitirá quantificar o transporte, a produção e a quantidade de formulação dos produtos compostos. O problema *MPC* poderá ser resolvido fazendo uso da nova proposta de aplicação do algoritmo simplex com particionamento primal considerado no capítulo anterior. Diferente das propostas apresentadas por [Ken80] e [Cas96], o algoritmo especializado para estrutura de redes utilizará uma matriz de dimensão fixa, denominada matriz ciclo Y .

4.1 REFORMULAÇÃO DO MODELO DE FLUXO EM REDE PARA MÚLTIPLOS PRODUTOS

O modelo de fluxo em redes para múltiplos produtos, tradicionalmente utilizados na otimização da logística de distribuição de vários produtos, pode ser reformulado a partir de uma tradução contextual conveniente do problema *MPC*.

O problema da distribuição de produtos compostos *MPC* visa estabelecer uma solução eficiente para o problema de distribuição de certos tipos de produtos que se compõe ao longo da cadeia de distribuição. Tal composição poderá, em muitos casos, ser realizada em pontos distintos dessa cadeia.

Com o objetivo de reduzir custos, tornando o produto competitivo no mercado, é que novos elementos de análise passaram a ser considerados na logística de distribuição, tendo em vista a necessidade de se encontrar respostas para questões como:

- i) onde e quanto produzir de produtos compostos?
- ii) por onde e quanto transportar de produtos compostos?
- iii) onde realizar a mistura desses produtos?
- iv) por onde e quanto transportar de mistura até os mercados consumidores?

Tais questões remetem o problema na direção de uma nova formulação para o modelo de fluxo em rede para o problema de múltiplos produtos. O problema MPC pode ser caracterizado como uma extensão do problema MP.

4.2 FORMULAÇÃO MATEMÁTICA

4.2.1 Caracterização de Variáveis de Decisão e de Parâmetros do Modelo

O problema *MPC* é formulado como um modelo de programação linear que, em linhas gerais, consiste em minimizar a função objetivo definida pela soma dos custos logísticos envolvidos, sujeito às restrições de conservação de fluxos caracterizadas para cada nó da rede, às restrições de atendimento, à demanda associada aos consumidores, às restrições de capacidade operacional associadas aos agentes econômicos envolvidos e às restrições de proporcionalidade da mistura entre determinados tipos de produtos.

Neste problema, considere os seguintes conjuntos:

- \mathcal{P} Conjunto de super nós de produção (fábricas);
- \mathcal{D} Conjunto de super nós de transbordo (bases de distribuição e terminais);
- \mathcal{C} Conjunto de super nós de consumo (municípios e consumidores genéricos);
- \mathcal{T} Conjunto de super arcos de transporte (dutos, ferrovias, hidrovias, rotas marítimas e rodovias);
- \mathcal{K}^0 Conjunto de produtos;
- $\mathcal{K}_i^0, \mathcal{K}_{i,j}^0$ Conjunto de produtos que trafegam pelo super nó i , ou pelo super arco (i, j) ;
- \mathcal{K}^m Conjunto de produtos que participam da mistura m ;
- $\mathcal{K}_i^m, \mathcal{K}_{i,j}^m$ Conjunto de produtos que participam da mistura m , e que trafega pelo super nó i , ou pelo super arco (i, j) ;
- \mathcal{M}^0 Conjunto de misturas;

- $\mathcal{M}_i^0, \mathcal{M}_{i,j}^0$ Conjunto de misturas que trafegam pelo super nó i , ou pelo super arco (i, j) ;
- \mathcal{M}^k Conjunto de misturas que utilizam o produto k ;
- $\mathcal{M}_i^k, \mathcal{M}_{i,j}^k$ Conjunto de misturas que utilizam o produto k , e que trafegam pelo super nó i , ou pelo super arco (i, j) ;
- \mathcal{F}_i Conjunto de misturas formuladas no super nó i ;

Considere ainda, as seguintes variáveis de decisão:

- $y_i^{0,k}$ Fluxo do produto k , que trafega no super nó i ;
- $y_i^{m,k}$ Fluxo de produto k , participante na formulação da mistura m , que trafega pelo super nó i ;
- $w_i^{m,k}$ Fluxo de produto k usado na formulação da mistura m , que trafega no super nó i ;
- $z_i^{0,k}$ Déficit de produto k no atendimento da demanda do super nó i ;
- $z_i^{m,k}$ Déficit de produto k , usado na formulação da mistura m , no atendimento da demanda do super nó i ;
- $x_{i,j}^{0,k}$ Fluxo do produto k transportado do super nó i para o super nó j ;
- $x_{i,j}^{m,k}$ Fluxo de produto k , participante na formulação da mistura m , transportado do super nó i para o super nó j .

Finalmente, considere os seguintes parâmetros do modelo:

- CTB_i^k Capacidade de transbordo do produto k , associada ao super nó i ;
- CTB_i^m Capacidade de transbordo da mistura m , associada ao super nó i ;
- CFR_i^m Capacidade de formulação da mistura m , associada ao super nó i ;
- CPR_i^k Capacidade de produção do produto k , associada ao super nó i ;
- $CTR_{i,j}^k$ Capacidade de transporte do produto k , associada ao super arco (i, j) ;
- $CTR_{i,j}^m$ Capacidade de transporte da mistura m , associada ao super arco (i, j) ;
- DEM_i^k Demanda de produto k , associado ao super nó i ;
- DEM_i^m Demanda de mistura m , associado ao super nó i ;
- $\alpha^{m,k}$ Proporção do produto k na formulação do produto composto m ;

- CY_i^k Custo unitário de transbordo/produção do produto k , associado ao super nó i ;
 CY_i^m Custo unitário de transbordo/produção da mistura m , associado ao super nó i ;
 CW_i^m Custo unitário de formulação da mistura m , associado ao super nó i ;
 $CX_{i,j}^k$ Custo unitário de transporte do produto k associado ao super arco (i, j) ;
 $CX_{i,j}^m$ Custo unitário de transporte do produto composto m , associado ao super arco (i, j) ;
 L^k Penalidade unitária associada ao déficit na entrega de produto k ; e
 L^m Penalidade unitária associada ao déficit na entrega de mistura m .

Utilizando os conjuntos, as variáveis e os parâmetros acima definidos, pode-se caracterizar o MPC através da formulação matemática que será apresentada a seguir.

4.2.2 Restrições Estruturais para Super Nós de Transbordo e Mistura

Cada distribuidor ou terminal do sistema logístico poderá ser representado no grafo por um super nó composto por diversos nós elementares associados aos produtos que por eles trafegam, conforme mostra a figura IV-1. Nestas entidades poderão ser realizadas formulações de produto, a exemplo da mistura de produtos dos tipos A e B, em proporções definidas para cada tipo de mistura.

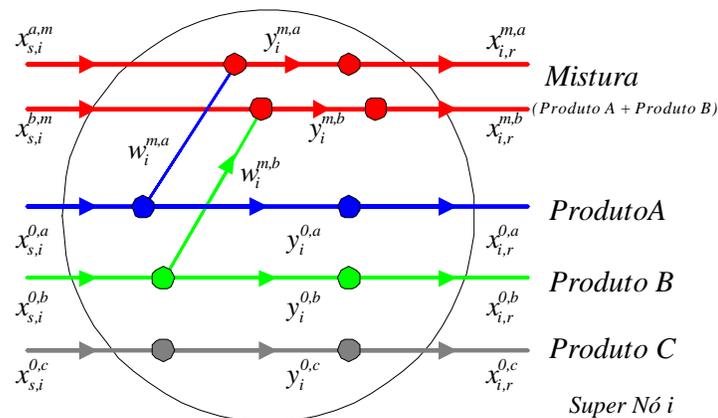


Figura IV-1 - Modelo lógico representando múltiplos produtos com mistura em distribuidores ou terminais ($i \in \mathcal{D}$). Nesta figura se observa o fluxo dos seguintes produtos: (a) Produto A – azul; (b) Produto B – verde; (m) mistura (Produto A \times Produto B) – vermelho; e (c) Produto C – cinza.

Considerando a representação lógica do super nó i , apresentado na figura IV-1, as seguintes restrições deverão ser satisfeitas:

a) Restrições de conservação de fluxo

O somatório das quantidades de fluxo do produto k transportado do super nó s para o super nó i deverá ser igual ao somatório das quantidades de fluxo de produto k , utilizados nas formulações das misturas, que trafegam no super nó i mais a quantidade de fluxo de produto k que trafega no super nó i , isto é:

$$- \sum_{(s,i) \in \mathcal{T}} x_{s,i}^{0,k} + \sum_{m \in \mathcal{F}_i \cap \mathcal{M}^k} w_i^{m,k} + y_i^{0,k} = 0 \quad \forall i \in \mathcal{D} \quad \forall k \in \mathcal{K}_i^0$$

A quantidade de fluxo do produto k que trafega no super nó i deverá ser igual ao somatório das quantidades de fluxo do produto k transportado do super nó i para o super nó r , isto é:

$$- y_i^{0,k} + \sum_{(i,r) \in \mathcal{T}} x_{i,r}^{0,k} = 0 \quad \forall i \in \mathcal{D} \quad \forall k \in \mathcal{K}_i^0$$

O somatório das quantidades de fluxo de produto k , participante na formulação da mistura, transportado do super nó s para o super nó i mais a quantidade de fluxo de produto k , utilizado na formulação da mistura, que trafega no super nó i deverá ser igual à quantidade de fluxo de produto, participante na formulação da mistura que trafega no super nó i , isto é:

$$- \sum_{(s,i) \in \mathcal{T}} x_{s,i}^{m,k} - w_i^{m,k} + y_i^{m,k} = 0 \quad \forall i \in \mathcal{D} \quad \forall k \in \mathcal{K}_i^0 \quad \forall m \in \mathcal{F}_i \cap \mathcal{M}^k$$

A quantidade de fluxo do produto k , participante na formulação da mistura e que trafega no super nó i , deverá ser igual ao somatório das quantidades de fluxo do produto k , participante na formulação da mistura, transportado do super nó i para o super nó r .

$$- y_i^{m,k} + \sum_{(i,r) \in \mathcal{T}} x_{i,r}^{m,k} = 0 \quad \forall i \in \mathcal{D} \quad \forall k \in \mathcal{K}_i^0 \quad \forall m \in \mathcal{F}_i \cap \mathcal{M}^k$$

b) Restrições de capacidade

O fluxo do produto k no super nó i é limitado superiormente e inferiormente por capacidades, isto é:

$$0 \leq y_i^{0,k} \leq CTB_i^k \quad \forall i \in \mathcal{D} \quad \forall k \in \mathcal{K}_i^0$$

O fluxo de produto k , participante na formulação da mistura, no super nó i é limitado superiormente e inferiormente por capacidades proporcionais, isto é:

$$0 \leq y_i^{m,k} \leq \alpha^{m,k} CTB_i^m \quad \forall i \in \mathcal{D} \quad \forall k \in \mathcal{K}_i^0 \quad \forall m \in \mathcal{F}_i \cap \mathcal{M}^k$$

O fluxo de produto k , utilizado na formulação da mistura, no super nó i é limitado superiormente e inferiormente por capacidades proporcionais, isto é:

$$0 \leq w_i^{m,k} \leq \alpha^{m,k} CFR_i^m \quad \forall i \in \mathcal{D} \quad \forall k \in \mathcal{K}_i^0 \quad \forall m \in \mathcal{F}_i \cap \mathcal{M}^k$$

c) Restrições de proporcionalidade das misturas

O fluxo de produto que compõe a mistura deve manter-se proporcional em relação ao fluxo dos demais componentes de mistura, isto é:

$$w_i^{m,k} - \alpha^{m,k} \sum_{q \in \mathcal{K}^m} w_i^{m,q} = 0 \quad \forall i \in \mathcal{D} \quad \forall k \in \mathcal{K}_i^0 \quad \forall m \in \mathcal{F}_i \cap \mathcal{M}^k$$

4.2.3 Restrições Estruturais para Super Nós Consumidores

Cada consumidor (município ou consumidor genérico) do sistema logístico poderá ser representado no grafo por um super nó composto por diversos nós elementares associados aos produtos consumidos, conforme mostra a figura IV-2.

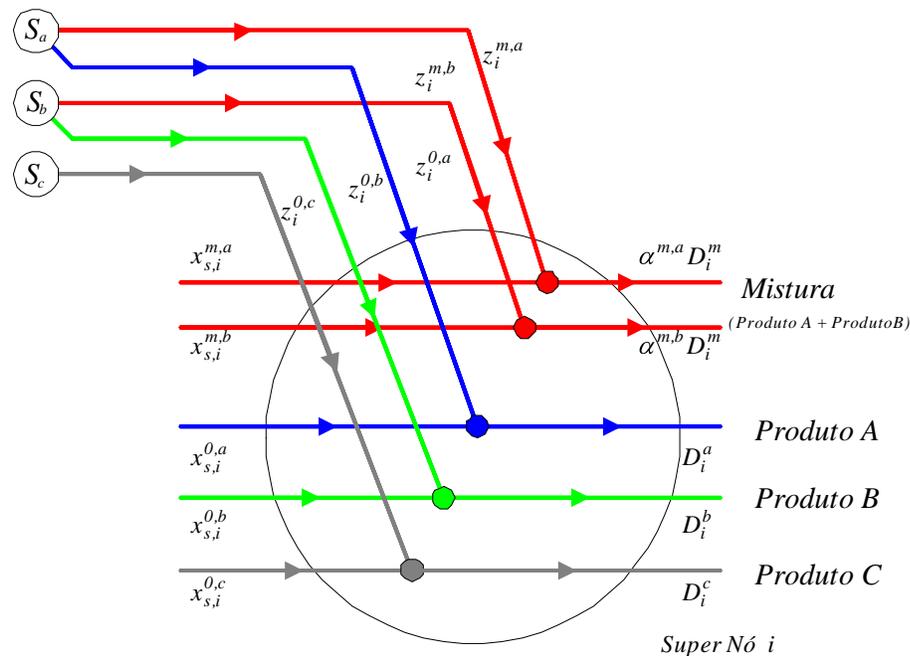


Figura IV-2 - Modelo lógico representando múltiplos produtos compostos em nós consumidores (super nó i). Nesta figura observa-se o fluxo dos seguintes produtos: (a) Produto A – azul; (b) Produto B – verde; (m) mistura (Produto A \times Produto B) – vermelho; e (c) Produto C – cinza.

Considerando a representação lógica do super nó i , apresentado na figura IV-2, as seguintes restrições deverão ser satisfeitas.

a) Restrições de conservação de fluxo

O somatório da quantidade de fluxo de produto k (ou da fração de produto k utilizado no composto m), adicionado ao déficit de produto k (ou da fração de produto k utilizado na formulação do composto m) no atendimento da demanda, deverá ser igual à quantidade de demanda do produto k (ou da fração de produto k utilizado no composto m) no super nó consumidor i , isto é:

$$-\sum_{(s,i) \in \mathcal{T}} x_{s,i}^{0,k} - z_i^{0,k} = -DEM_i^k \quad \forall i \in \mathcal{C} \quad \forall k \in \mathcal{K}_i$$

$$-\sum_{(s,i) \in \mathcal{T}} x_{s,i}^{m,k} - z_i^{m,k} = -\alpha^{m,k} DEM_i^m \quad \forall i \in \mathcal{C} \quad \forall m \in \mathcal{M}_i \quad \forall k \in \mathcal{K}^m$$

b) Restrições de capacidade

O fluxo de produto no atendimento da demanda no super nó consumidor i é limitado superiormente e inferiormente pela capacidade máxima do déficit de produto k (ou da fração de produto k utilizado na formulação do composto m), isto é:

$$0 \leq z_i^{0,k} \leq DEM_i^k \quad \forall i \in \mathcal{C} \quad \forall k \in \mathcal{K}_i$$

$$0 \leq z_i^{m,k} \leq \alpha^{m,k} DEM_i^m \quad \forall i \in \mathcal{C} \quad \forall m \in \mathcal{M}_i \quad \forall k \in \mathcal{K}^m$$

4.2.4 Restrições Estruturais para Super Nós de Produção

Cada produtor (fábricas) do sistema logístico poderá ser representado no grafo por um super nó composto por diversos nós elementares associados aos produtos produzidos, conforme mostra a figura IV-3.

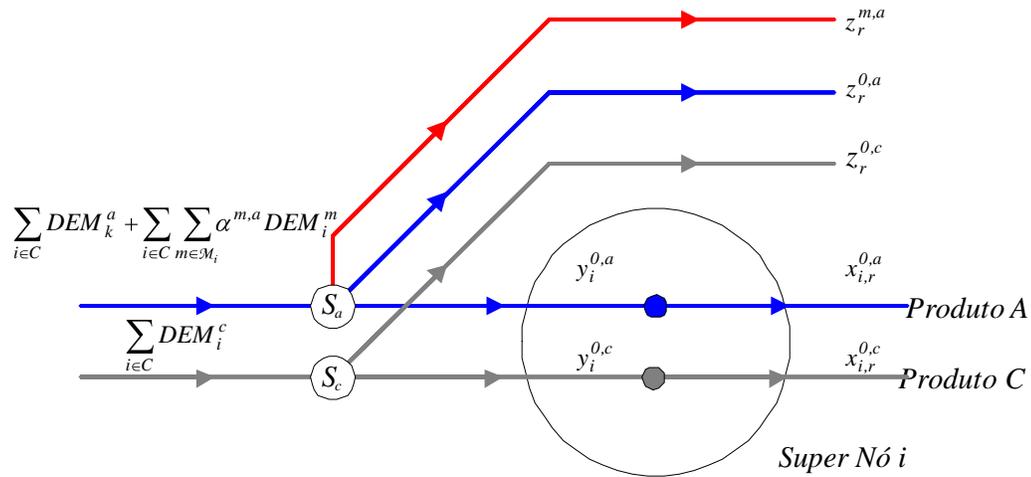


Figura IV-3 - Modelo lógico representando um produtor com múltiplos produtos (super nó i). Nesta figura observa-se o fluxo dos seguintes produtos: (a) Produto A – azul; (m) mistura (Produto A × Produto B) – vermelho; e (c) Produto C – cinza; e (b) Produto B – verde.

Considerando a representação lógica do super nó i , apresentado na figura IV-3, as seguintes restrições deverão ser satisfeitas:

a) Restrições de conservação de fluxo

A quantidade de fluxo de produto k no super nó de produção i é igual ao somatório da quantidade de fluxo de produto k do super nó i para o super nó r , isto é:

$$-y_i^{0,k} + \sum_{(i,r) \in \mathcal{T}} x_{i,r}^{0,k} = 0 \quad \forall i \in \mathcal{C} \quad \forall k \in \mathcal{K}_i$$

Essa restrição de conservação de fluxo é referida ao nó fonte S da figura IV-3, de forma a obter sempre uma rede na qual os fluxos encontram-se em equilíbrio, isto é, cuja quantidade de oferta é sempre igual à quantidade de demanda.

$$-\sum_{i \in \mathcal{P}} y_i^{0,k} - \sum_{r \in \mathcal{C}} \left(z_r^{0,k} + \sum_{m \in \mathcal{M}^k} z_r^{m,k} \right) = -\sum_{r \in \mathcal{C}} \left(DEM_i^k + \sum_{m \in \mathcal{M}^k} \alpha^{m,k} DEM_r^m \right) \quad \forall i \in \mathcal{C} \quad \forall k \in \mathcal{K}_i$$

b) Restrições de capacidade

O fluxo de produto k é ilimitado superiormente e inferiormente pela capacidade máxima de produção do produto k no super nó i .

$$0 \leq y_i^{0,k} \leq CPR_i^k$$

$$\forall i \in \mathcal{C} \quad \forall k \in \mathcal{K}_i$$

4.2.5 Restrições Estruturais para Super Arcos de Transporte

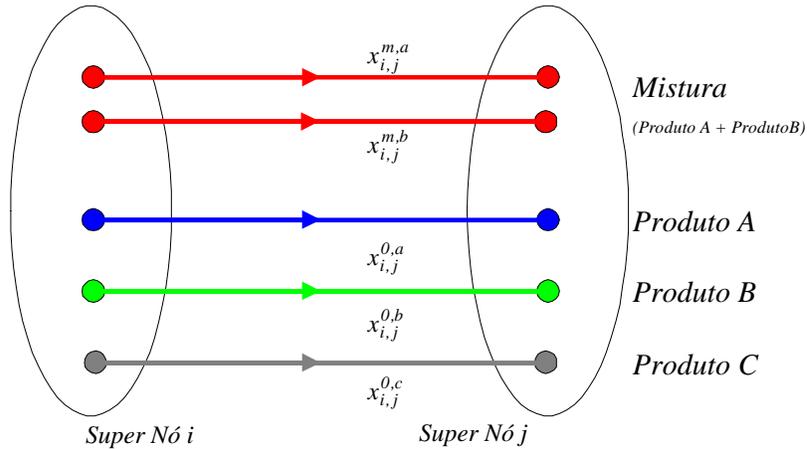


Figura IV-4 - Modelo lógico representando o transporte entre dois super nós através de um super arco (i, j) . Nesta figura observa-se o fluxo dos seguintes produtos: (a) Produto A – azul; (m) mistura (Produto A \times Produto B) – vermelho; (b) Produto B – verde; e (c) Produto C – cinza.

Considerando a representação lógica do super arco (i, j) , apresentado na figura IV-4, as seguintes restrições de capacidade deverão ser satisfeitas:

O fluxo de produto k (ou da fração de produto k participante no composto m) no super arco (i, j) é limitado superiormente e inferiormente pelas capacidades, isto é:

$$0 \leq x_{i,j}^{0,k} \leq CTR_{i,j}^k \quad \forall (i, j) \in \mathcal{T} \quad \forall k \in \mathcal{K}_{i,j}$$

$$0 \leq x_{i,j}^{m,k} \leq \alpha^{m,k} CTR_{i,j}^m \quad \forall (i, j) \in \mathcal{T} \quad \forall m \in \mathcal{M}_{i,j} \quad \forall k \in \mathcal{K}^m$$

4.2.6 Função Objetivo

O critério de otimização no *MPC* é definido pela seguinte função objetivo, que representa a soma de todos os custos logísticos envolvidos:

$$\begin{aligned} \text{Min } z = & \sum_{i \in \mathcal{P} \cup \mathcal{D}} \sum_{k \in \mathcal{K}_i} CY_i^k y_i^{0,k} + \sum_{i \in \mathcal{D}} \sum_{m \in \mathcal{M}_i} \sum_{k \in \mathcal{K}^m} (CY_i^m y_i^{m,k} + CW_i^m w_i^{m,k}) + \\ & \sum_i \sum_j \sum_k CX_{i,j}^k x_{i,j}^{0,k} + \sum_i \sum_j \sum_m \sum_{k \in \mathcal{K}^m} CX_{i,j}^m x_{i,j}^{m,k} + \sum_{i \in \mathcal{C}} \sum_{k \in \mathcal{K}_i} L^k z_i^{0,k} + \sum_{i \in \mathcal{C}} \sum_{m \in \mathcal{M}_i} \sum_{k \in \mathcal{K}^m} L^m z_i^{m,k} \end{aligned}$$

onde, as parcelas acima correspondem, respectivamente:

- i) custo associado ao fluxo do produto k em um super nó de produção ou de transbordo;
- ii) custo associado ao fluxo de produto k que participa e é utilizado na formulação do composto m em um super nó de transbordo;
- iii) custo associado ao fluxo de produto k em um super arco de transporte (i, j) ;
- iv) fluxo de produto composto m em um super arco de transporte (i, j) ;
- v) penalidade associada ao déficit na entrega de produto k no super nó de consumo;
- vi) penalidade associada ao déficit na entrega do composto m no super nó de consumo.

4.3 MODELO E ESTRUTURA DE REDE

Considerando as representações lógicas de super nós e super arcos apresentados nas figuras IV-1; IV-2; IV-3 e IV-4; para cada produto k será construída uma rede conforme mostra a figura IV-5. Nestas redes observa-se o fluxo do produto (azul), as respectivas frações na operação de formulação de produtos compostos (verde) e as respectivas frações no fluxo de produtos compostos (vermelho). Os arcos tracejados que unem o nó de origem aos nós de demanda correspondem, nesta representação, ao fluxo de produto k (ou fração de k de um produto composto m) que deixa de ser entregue para satisfazer a demanda de um determinado nó.

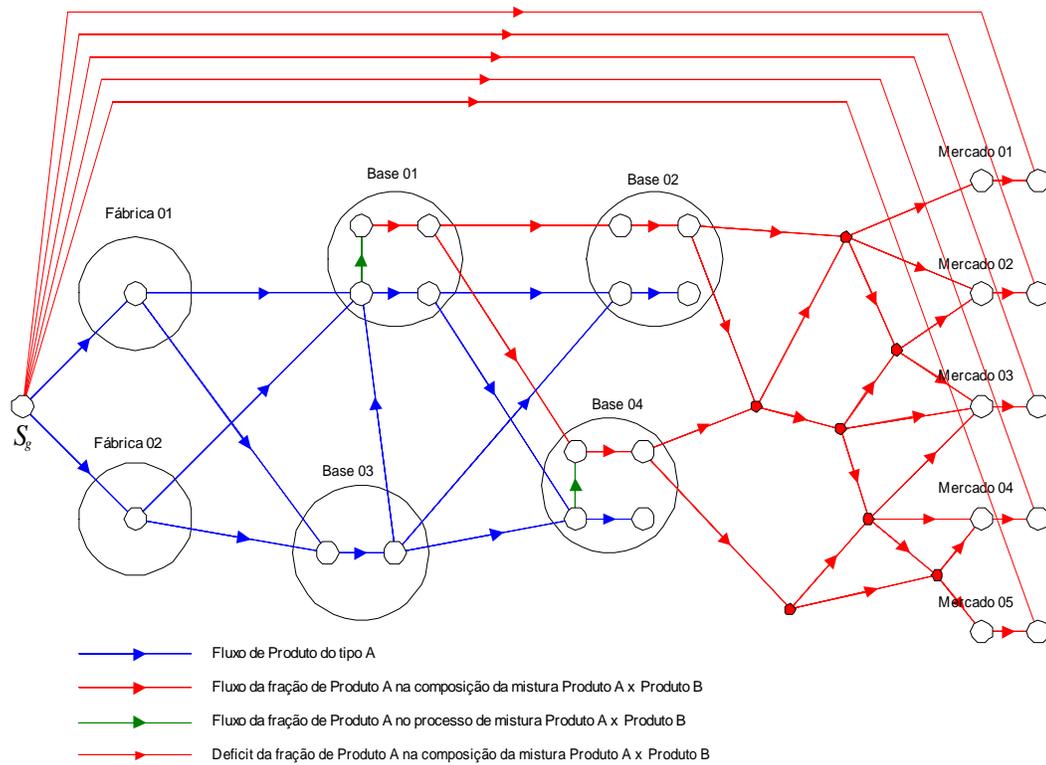


Figura IV-5 - Modelo lógico representando a rede completa para um produto (Produto A), incluído a fração correspondente do fluxo dos produtos compostos derivados (mistura Produto A x Produto B). Note-se que na Base 01 e na Base 04 é permitida a formulação do produto composto, enquanto que na Base 02 e na Base 03 somente é permitido o transbordo dos produtos.

Deve-se salientar que, a representação lógica do modelo de rede não representa o composto m , mas sim, a representação da fração dos vários produtos participantes na formulação do composto m .

Desconsiderando a existência das restrições de proporcionalidade na formulação de produtos compostos e outras restrições de agregação de fluxos, o problema poderá ser formulado e resolvido para cada produto de forma independente, isto é:

$$\begin{aligned} \text{Min } z = & \sum_{i \in P \cup D} \sum_{k \in K_i} C Y_i^k y_i^{0,k} + \sum_{i \in D} \sum_{m \in M_i} \sum_{k \in K^m} (C Y_i^m y_i^{m,k} + C W_i^m w_i^{m,k}) + \\ & \sum_i \sum_j \sum_k C X_{i,j}^k x_{i,j}^{0,p} + \sum_i \sum_j \sum_m \sum_{k \in K^m} C X_{i,j}^m x_{i,j}^{m,k} + \sum_{i \in C} \sum_{k \in K_i} L^k z_i^{0,k} + \sum_{i \in C} \sum_{m \in M_i} \sum_{k \in K^m} L^m z_i^{m,k} \end{aligned} \quad (4.1)$$

sujeito a:

$$- \sum_{(s,i) \in \mathcal{T}} x_{s,i}^{0,k} + \sum_{m \in F_i \cap M^k} w_i^{m,k} + y_i^{0,k} = 0 \quad \forall i \in \mathcal{D} \quad (4.2)$$

$$- y_i^{0,k} + \sum_{(i,r) \in \mathcal{T}} x_{i,r}^{0,k} = 0 \quad \forall i \in \mathcal{D} \quad (4.3)$$

$$- \sum_{(s,i) \in \mathcal{T}} x_{s,i}^{m,k} - w_i^{m,k} + y_i^{m,k} = 0 \quad \forall i \in \mathcal{D} \quad \forall m \in \mathcal{F}_i \cap \mathcal{M}^k \quad (4.4)$$

$$- y_i^{m,k} + \sum_{(i,r) \in \mathcal{T}} x_{i,r}^{m,k} = 0 \quad \forall i \in \mathcal{D} \quad \forall m \in \mathcal{F}_i \cap \mathcal{M}^k \quad (4.5)$$

$$- \sum_{(s,i) \in \mathcal{T}} x_{s,i}^{0,k} - z_i^{0,k} = -DEM_i^k \quad \forall i \in \mathcal{C} \quad (4.6)$$

$$- \sum_{(s,i) \in \mathcal{T}} x_{s,i}^{m,k} - z_i^{m,k} = -\alpha^{m,k} DEM_i^m \quad \forall i \in \mathcal{C} \quad \forall m \in \mathcal{M}_i \quad (4.7)$$

$$- y_i^{0,k} + \sum_{(i,r) \in \mathcal{T}} x_{i,r}^{0,k} = 0 \quad \forall i \in \mathcal{C} \quad (4.8)$$

$$- \sum_{i \in \mathcal{P}} y_i^{0,k} - \sum_{r \in \mathcal{C}} \left(z_r^{0,k} + \sum_{m \in \mathcal{M}^k} z_r^{m,k} \right) = - \sum_{r \in \mathcal{C}} \left(DEM_r^k + \sum_{m \in \mathcal{M}^k} \alpha^{m,k} DEM_r^m \right) \quad \forall i \in \mathcal{C} \quad (4.9)$$

$$0 \leq y_i^{0,k} \leq CTB_i^k \quad \forall i \in \mathcal{D} \quad (4.10)$$

$$0 \leq y_i^{m,k} \leq \alpha^{m,k} CTB_i^m \quad \forall i \in \mathcal{D} \quad \forall m \in \mathcal{F}_i \cap \mathcal{M}^k \quad (4.11)$$

$$0 \leq w_i^{m,k} \leq \alpha^{m,k} CFR_i^m \quad \forall i \in \mathcal{D} \quad \forall m \in \mathcal{F}_i \cap \mathcal{M}^k \quad (4.12)$$

$$0 \leq z_i^{0,k} \leq DEM_i^k \quad \forall i \in \mathcal{C} \quad (4.13)$$

$$0 \leq z_i^{m,k} \leq \alpha^{m,k} DEM_i^m \quad \forall i \in \mathcal{C} \quad \forall m \in \mathcal{M}_i \quad (4.14)$$

$$0 \leq y_i^{0,k} \leq CPR_i^k \quad \forall i \in \mathcal{C} \quad (4.15)$$

$$0 \leq x_{i,j}^{0,k} \leq CTR_{i,j}^k \quad \forall (i,j) \in \mathcal{T} \quad (4.16)$$

$$0 \leq x_{i,j}^{m,k} \leq \alpha^{m,k} CTR_{i,j}^m \quad \forall (i,j) \in \mathcal{T} \quad \forall m \in \mathcal{M}_{i,j} \quad (4.17)$$

$$w_i^{m,k} - \alpha^{m,k} \sum_{q \in \mathcal{K}^m} w_i^{m,q} = 0 \quad \forall i \in \mathcal{D} \quad \forall k \in \mathcal{K}_i^0 \quad \forall m \in \mathcal{F}_i \cap \mathcal{M}^k \quad (4.18)$$

O problema de rede de custo mínimo formulado acima, poderá ser facilmente descrito em termos de um problema de distribuição sobre uma rede G , representada por um conjunto de n nós elementares (nós que compõem os super-nós) e um conjunto de t arcos como mostrados na figura IV-5. Esta rede é composta de arcos cujo fluxo circulante representa unicamente um produto k .

A função objetivo z^k representa o custo total da logística associada à produção, ao transporte, ao transbordo e à distribuição de um produto k .

As variáveis de decisão $y_i^{0,k}, y_i^{m,k}, w_i^{m,k}, x_{i,j}^{0,k}, x_{i,j}^{m,k}, z_i^{0,k}$ e $z_i^{m,k}$ são denotadas unicamente por x^k representando o vetor de fluxos para todos os produtos k .

Os custos unitários denotados por $CY_i^k, CY_i^m, CW_i^m, CX_{i,j}^k, CX_{i,j}^m, L^k$ e L^m passam a ser denotados unicamente por c^k representando o vetor de custos para todos os arcos dos produtos k .

As capacidades nos arcos $CTB_i^k, CTB_i^m, CFR_i^m, CPR_i^k, CTR_{i,j}^k$ e $CTR_{i,j}^m$ são denotadas pelo vetor u^k , representando o limite superior de fluxo nos arcos da rede para cada produto k , e as capacidades mínimas pelo vetor l^k , associadas aos arcos da rede para cada produto k .

A matriz de incidência nó-arco associada à rede do produto k será representada por A^k e o vetor de ofertas e demandas associadas a cada nó da rede para o produto k será representado por r^k .

Generalizando, o problema acima poderá ser reescrito para cada produto de forma mais condensada como:

$$NP \quad \text{Min} \quad z = \sum_k c^k x^k \quad (4.19)$$

$$\text{s.a:} \quad A^k x^k = r^k \quad \forall k \quad (4.20)$$

$$l^k \leq x^k \leq u^k \quad (4.21)$$

Considerando, a existência das restrições de proporcionalidade na formulação de produtos compostos e de agregação de fluxos aos problemas NP 's formulado anteriormente tem-se um problema de programação linear de fluxo em rede para múltiplos produtos MP .

O problema Linear MP , inclui a restrição de agregação para o caso geral, sendo definido por:

$$b' \leq \sum_{k=1}^K D^k x^k \leq b \quad (4.22)$$

As componentes dos vetores b' e b representam as capacidades mínimas e máximas associadas às restrições de agregação de fluxos. Estas capacidades são divididas entre os vários produtos k . D^k serve como um fator ponderado para x^k em relação às limitações dessas capacidades, podendo ser expresso em unidades diferentes a partir da utilização de x^k . Considera-se sempre, para todo produto k , que a quantidade de oferta e demanda sejam iguais.

A restrição (4.22) poderá unir arcos de um mesmo ou de diferentes produtos. A restrição de proporcionalidade para o problema MPC pode ser considerada como um caso

particular da restrição (4.22) quando se tem $\sum_{k=1}^K D^k x^k = 0$. Matematicamente, o problema *MP* poderá ser formulado como:

$$\text{Min } z = \sum_k c^k x^k \quad (4.23)$$

$$\text{s.a: } A^k x^k = r^k \quad \forall k \quad (4.24)$$

$$\sum_{k=1}^K D^k x^k - s' = b' \quad (4.25)$$

$$\sum_{k=1}^K D^k x^k + s = b \quad (4.26)$$

$$l^k \leq x^k \leq u^k \quad (4.27)$$

e, será referido como *MPC* quando incluir as restrições de proporcionalidade.

O problema de fluxo de custo mínimo para múltiplos produtos possui $(k+1)t$ variáveis e $t + nk$ restrições (incluindo as variáveis de folga para as restrições de agregação e ignorando as restrições $l^k \leq x^k \leq u^k$). Então, mesmo em um problema de tamanho moderado, a matriz de restrições será grande. Por exemplo, um problema contendo 100 nós, 250 arcos e 10 produtos, terá 2750 variáveis e 1250 restrições.

O problema acima poderá ser resolvido de forma eficiente fazendo uso do algoritmo simplex com particionamento primal, especializado para estrutura de redes.

A matriz correspondente às restrições para o problema *MPC* possui a estrutura da matriz (3.21) apresentada no capítulo 3, assim representada:

	Produto 1	Produto 2	Produto k	Folgas
Restrições de Conservação de Fluxo para o Produto 1	$A^1 e^1$			
⋮		\ddots		
Restrições de Conservação de Fluxo para o Produto k			$A^k e^k$	
Restrições de capacidade e Proporcionalidade	$D^1 0$	$D^2 0$	$D^k 0$	I

As restrições de conservação de fluxo para cada produto k são expressas através de (4.2)-(4.9), as restrições de capacidade através de (4.10)-(4.17) e as restrições de proporcionalidade através da expressão (4.18).

O exemplo a seguir, ilustra uma rede que permite a distribuição de produtos compostos. A representação da matriz básica aplicada ao problema MPC é apresentada na figura IV-9.

O exemplo considera um problema de mistura (MPC) de dois tipos de adubos A e B , cuja rede é ilustrada na figura IV-6. A representação dos nós de oferta para os adubos tipo A e B são indicadas respectivamente por (R_1, R_2) e (U_1) . A formulação do produto composto (adubo tipo A x adubo tipo B) é realizada nos super nós B_1 e B_2 , sendo transportada até os nós de demanda representados por $(C_1, C_2$ e $C_3)$.

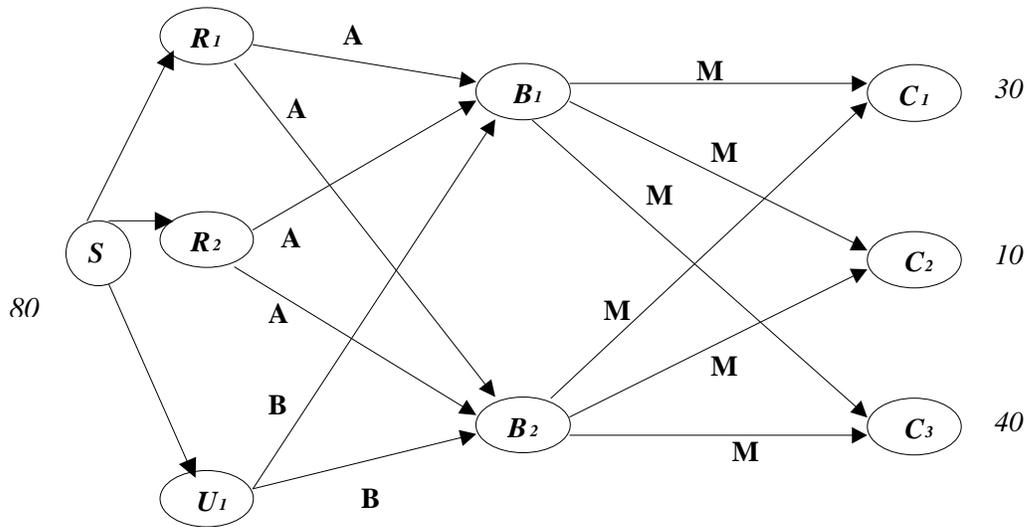


Figura IV-6 - Rede completa para os produtos dos tipos A, B e produtos compostos.

Considerando a representação lógica dos super nós (B_1 e B_2), para cada produto do tipo A e B é construída uma rede conforme mostra as figura IV-7 e IV-8. A mistura (Adubo tipo A x Adubo tipo B) é composta de 75% do produto tipo A e 25% do produto tipo B, sendo que a capacidade de formulação do produto composto (mistura) nos super nós B_1 e B_2 são 30 e 50 unidades respectivamente. As especificações dos arcos são apresentadas nas tabelas IV-1 e IV-2. As restrições que unem os arcos $A_1M_1(a)$ e $B_1M_1(b)$ são mostradas a seguir:

$$w_{B_1}^{m,A} + w_{B_1}^{m,B} + S_1 = 30 \text{ e } \frac{1}{0,75} w_{B_1}^{m,A} = \frac{1}{0,25} w_{B_1}^{m,B},$$

e as restrições que unem $A_2M_2(a)$ e $B_2M_2(b)$ são:

$$w_{B_2}^{m,A} + w_{B_2}^{m,B} + S_2 = 50 \text{ e } \frac{1}{0,75} w_{B_2}^{m,A} = \frac{1}{0,25} w_{B_2}^{m,B}.$$

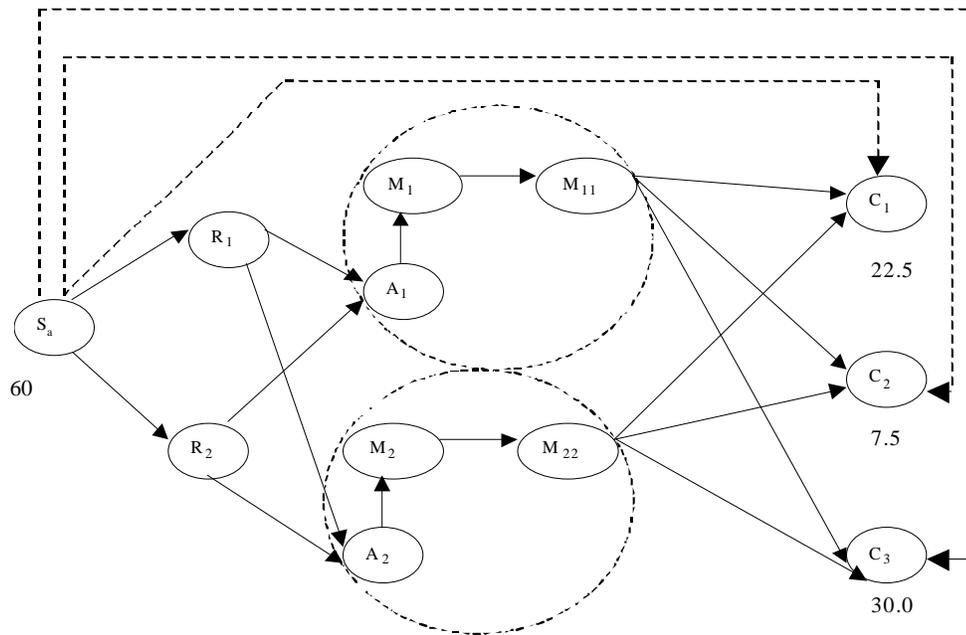


Figura IV-7 - Rede completa em equilíbrio representando o produto do tipo A, incluindo a fração do fluxo do produto composto derivado (mistura adubo A x adubo B).

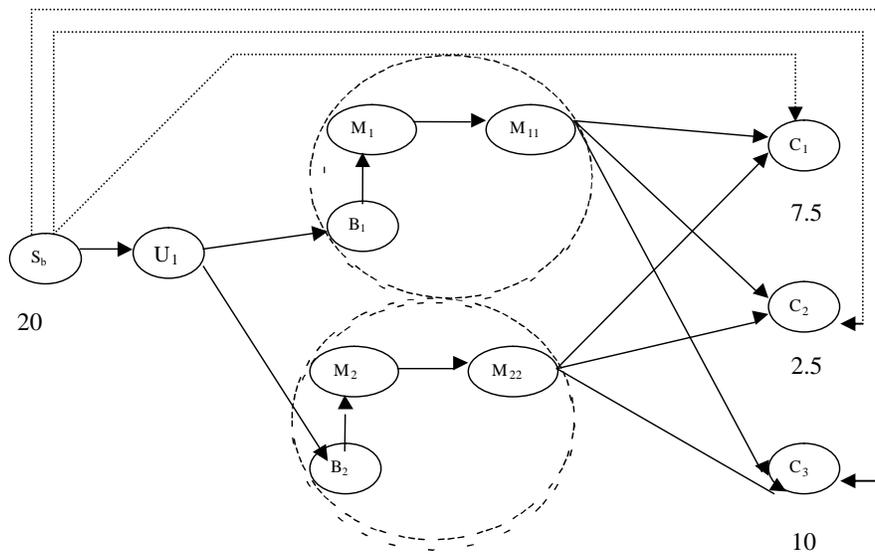


Figura IV-8 - Rede completa em equilíbrio representando o produto do tipo B, incluindo a fração do fluxo do produto composto derivado (mistura adubo A x adubo B).

Tabela IV-1 - Especificação dos Arcos (Adubo do tipo A)

ARCOS	CUSTO UNITÁRIO	CAPACIDADE	FLUXO INICIAL	FLUXO ÓTIMO
S_aC_1	100	∞	22.5	0.0
S_aC_2	100	∞	7.5	0.0
S_aC_3	100	∞	30	0.0
S_aR_1	0.0	∞	0.0	22.5
S_aR_2	0.0	∞	0.0	27.5
R_1A_1	3.0	∞	0.0	22.5
R_1A_2	5.0	∞	0.0	0.0
R_2A_1	4.0	∞	0.0	0.0
R_2A_2	2.0	∞	0.0	37.5
A_1M_1	0.0	∞	0.0	22.5
A_2M_2	0.0	∞	0.0	37.5
$M_1M_{11}(a)$	0.0	∞	0.0	22.5
$M_2M_{22}(a)$	0.0	∞	0.0	37.5
$M_{11}C_1(a)$	4.5	∞	0.0	0.0
$M_{11}C_2(a)$	5.0	∞	0.0	0.0
$M_{11}C_3(a)$	4.0	∞	0.0	22.5
$M_{22}C_1(a)$	2.5	∞	0.0	22.5
$M_{22}C_2(a)$	3.5	∞	0.0	7.5
$M_{22}C_3(a)$	4.0	∞	0.0	7.5

Tabela IV-2 - Especificação dos Arcos (Adubo do tipo B)

ARCOS	CUSTO UNITÁRIO	CAPACIDADE	FLUXO INICIAL	FLUXO ÓTIMO
S_bC_1	100	∞	7.5	0.0
S_bC_2	100	∞	2.5	0.0
S_bC_3	100	∞	10	0.0
S_aU_1	0.0	∞	0.0	20.0
U_1B_1	3.0	∞	0.0	7.5
U_1B_2	4.0	∞	0.0	12.5
B_1M_1	0.0	∞	0.0	7.5
B_2M_2	0.0	∞	0.0	12.5
$M_1M_{11}(b)$	0.0	∞	0.0	7.5
$M_2M_{22}(b)$	0.0	∞	0.0	12.5
$M_{11}C_1(b)$	4.5	∞	0.0	0.0
$M_{11}C_2(b)$	5.0	∞	0.0	0.0
$M_{11}C_3(b)$	4.0	∞	0.0	7.5
$M_{22}C_1(b)$	2.5	∞	0.0	7.5
$M_{22}C_2(b)$	3.5	∞	0.0	2.5
$M_{22}C_3(b)$	4.0	∞	0.0	2.5

O problema acima consiste em minimizar a função objetivo definida pela minimização dos custos nos arcos, sujeito às restrições de conservação de fluxo dos produtos, às restrições de proporcionalidade e às restrições de capacidade das mistura desses produtos.

A quantidade de oferta e demanda para ambos os produtos são iguais, respeitando as frações de fluxo dos produtos que compõem a mistura. A capacidade máxima individual u^k dos arcos para cada produto é infinita e a capacidade mínima l^k é igual a zero.

Os arcos que compõem a árvore básica para o produto do tipo *A* são $S_aR_1, S_aR_2, R_1A_1, R_2A_2, A_1M_1, A_2M_2, M_1M_{11}(a), M_2M_{22}(a), S_aC_1, S_aC_2, S_aC_3$.

Os arcos que compõem a árvore básica para o produto do tipo *B* são $S_bU_1, U_1B_1, U_1B_2, B_1M_1, B_2M_2, M_1M_{11}(b), M_2M_{22}(b), S_bC_1, S_bC_2, S_bC_3$.

As restrições de agregação são escritas sob a forma:

$$\begin{cases} A_1M_1 + B_1M_1 + S_1 = 30 \\ \frac{A_1M_1}{0,75} - \frac{B_1M_1}{0,25} + S_2 = 0 \\ A_2M_2 + B_2M_2 + S_3 = 50 \\ \frac{A_2M_2}{0,75} - \frac{B_2M_2}{0,25} + S_4 = 0 \end{cases}$$

O quadro apresentado na figura IV-9 representa a matriz básica \bar{B} dada em (3.22) com partições R^k vazias.

A estrutura do modelo permite que o arco por onde circula o composto m (mistura) seja dividido em tantos arcos quanto for o número de produtos k participantes na formulação do composto m . Cada produto k participante na formulação do produto composto m possui o mesmo modelo de rede em relação aos arcos nos quais circulam as frações do composto m . A solução básica inicial é obtida através do *Método M Grande*. A quantidade de demanda do composto m deverá ser igual à soma das frações de fluxo dos produtos k que compõem o composto m , portanto, a quantidade fracionada de produto k através de arcos artificiais satisfaz as condições de proporcionalidade. Toda quantidade de demanda é alocada sobre os arcos artificiais.

PÁGINA 90 IMPRESSA EM OUTRO ARQUIVO

Figura IV-9 - Representação da matriz Básica

No Apêndice II, são realizadas três iterações completas ao problema de mistura considerado acima.

4.4 OPERACIONALIZAÇÃO DA BUSCA DE UMA BASE INICIAL VIÁVEL

Na obtenção da base inicial viável do exemplo anterior, que ilustrou a aplicação do algoritmo com particionamento primal em função de uma matriz de dimensão fixa, foi utilizado o *Método do M Grande*. Algumas desvantagens em relação ao tempo de processamento do algoritmo, na busca da solução ótima em relação aos problemas de múltiplos produtos com restrições de acoplamento, podem ser encontradas na aplicação do *Método M Grande*. Além disso, a solução viável para problemas de múltiplos produtos geralmente é obtida utilizando um algoritmo que busca uma base viável para, posteriormente buscar o ótimo do problema em si.

A solução inicial para o problema *MP* descrita por Castro e Nabona [Cas96] provou ser muito eficiente. O processo utilizado para encontrar o ponto inicial viável é dividido em dois estágios. No primeiro estágio são resolvidos problemas de monoprodutos e no segundo estágio após inserção das restrições de acoplamento, é obtida a solução viável para o problema com particionamento primal, através de variáveis artificiais.

Com base nessa estratégia de obter uma solução viável a partir da solução ótima dos problemas monoprodutos surge o desenvolvimento de um algoritmo com particionamento dual da base que também utiliza as propriedades da matriz ciclo de dimensão fixa. No próximo capítulo será mostrado o desenvolvimento do método simplex primal-dual especializado para os problemas de fluxo em rede.

CAPÍTULO V

5. MÉTODO SIMPLEX PRIMAL-DUAL ESPECIALIZADO PARA O PROBLEMA DE FLUXO EM REDES COM MÚLTIPLOS PRODUTOS

Este capítulo apresenta o método simplex dual para solucionar um problema particionado primal cuja base inicial não é disponível. A estratégia adotada na resolução dos problemas com múltiplos produtos é dividida em dois estágios. O método dual é aplicado somente no segundo estágio, a partir do momento em que as restrições de agregação são inseridas ao problema. A forma como as iterações duais são realizadas sobre o problema particionado é apresentada.

O algoritmo simplex dual surge da necessidade de buscar a solução ótima para o problema de múltiplos produtos a partir da solução ótima de cada um dos problemas mono-produto. Em muitas situações práticas é importante ser capaz de encontrar a nova solução ótima sem o esforço computacional de resolver o problema a partir do ponto inicial em decorrência de modificações nas restrições de agregação. Além do mais, dividir o processo para encontrar o ponto inicial viável em dois estágios tem provado ser muito eficiente em um número de iterações com relação ao método que inicia a partir de um ponto dado considerando o total de inviabilidades para todas as restrições.

5.1 ESTRATÉGIA DE SOLUÇÃO

A implementação do método que soluciona o problema de múltiplos produtos *MP* ou *MPC* segue dois estágios. O primeiro estágio resolve problemas de monoproduto com arcos capacitados sem considerar as restrições de agregação de fluxos. No segundo estágio, após a inclusão das restrições de agregação de fluxos, aplica-se o método dual. Para esclarecer essas idéias, cada estágio será descrito a seguir.

Inicialmente, procura-se encontrar um ponto inicial viável onde o algoritmo considera somente as restrições de rede e restrições de limites para as variáveis do problema. Nesse estágio não são consideradas as restrições de capacidade de formulação dos produtos compostos nos super nós de produção e as restrições que agregam fluxos de um mesmo produto ou de diferentes produtos. Tenta-se obter para cada produto $k, k=1, \dots, K$ um ponto inicial viável para o problema linear de rede.

$$\text{Min } z = [c_B^1 \quad c_B^2 \quad \dots \quad c_B^K] \begin{bmatrix} x_B^1 \\ x_B^2 \\ \vdots \\ x_B^K \end{bmatrix} + [c_N^1 \quad c_N^2 \quad \dots \quad c_N^K] \begin{bmatrix} x_N^1 \\ x_N^2 \\ \vdots \\ x_N^K \end{bmatrix}$$

$$\text{s.a: } \begin{bmatrix} B^1 & 0 & \dots & 0 \\ 0 & B^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B^K \end{bmatrix} \begin{bmatrix} x_B^1 \\ x_B^2 \\ \vdots \\ x_B^K \end{bmatrix} + \begin{bmatrix} N^1 & 0 & \dots & 0 \\ 0 & N^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & N^K \end{bmatrix} \begin{bmatrix} x_N^1 \\ x_N^2 \\ \vdots \\ x_N^K \end{bmatrix} = \begin{bmatrix} r^1 \\ r^2 \\ \vdots \\ r^K \end{bmatrix}$$

$$l_B^k \leq x_B^k \leq u_B^k \quad \forall k = 1, \dots, K$$

$$l_N^k \leq x_N^k \leq u_N^k \quad \forall k = 1, \dots, K.$$

onde os indexadores B e N utilizados acima identificam, respectivamente, os elementos básicos e não básicos. Redefinindo os termos do sistema acima, tem-se⁴.

$$\text{Min } z = c^B x^B + c^N x^N$$

$$Bx^B + Nx^N = r$$

$$l^B \leq x^B \leq u^B$$

$$l^N \leq x^N \leq u^N$$

onde

$$x^B \leftarrow \begin{bmatrix} x_B^1 \\ x_B^2 \\ \vdots \\ x_B^K \end{bmatrix}, \quad u^B \leftarrow \begin{bmatrix} u_B^1 \\ u_B^2 \\ \vdots \\ u_B^K \end{bmatrix}, \quad c^B \leftarrow \begin{bmatrix} c_B^1 \\ c_B^2 \\ \vdots \\ c_B^K \end{bmatrix}, \quad l^B \leftarrow \begin{bmatrix} l_B^1 \\ l_B^2 \\ \vdots \\ l_B^K \end{bmatrix},$$

⁴ Note-se que este problema pode ser desacoplado em problemas isolados de dimensão menor.

$$x^N \leftarrow \begin{bmatrix} x_N^1 \\ x_N^2 \\ \vdots \\ x_N^K \end{bmatrix}, \quad u^N \leftarrow \begin{bmatrix} u_N^1 \\ u_N^2 \\ \vdots \\ u_N^K \end{bmatrix}, \quad c^N \leftarrow \begin{bmatrix} c_N^1 \\ c_N^2 \\ \vdots \\ c_N^K \end{bmatrix}, \quad l^N \leftarrow \begin{bmatrix} l_N^1 \\ l_N^2 \\ \vdots \\ l_N^K \end{bmatrix},$$

$$B \leftarrow \begin{bmatrix} B^1 & 0 & \dots & 0 \\ 0 & B^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B^K \end{bmatrix}, \quad N \leftarrow \begin{bmatrix} N^1 & 0 & \dots & 0 \\ 0 & N^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & N^K \end{bmatrix}, \quad r \leftarrow \begin{bmatrix} r^1 \\ r^2 \\ \vdots \\ r^K \end{bmatrix}$$

5.1.1 Estágio I

A obtenção da solução inicial é feita com o auxílio de um algoritmo que, determina o caminho mínimo a partir do nó v_0 , para cada vértice v da rede. Para cada um dos problemas monoproduto a alocação da demanda é feita através dos caminhos mínimos, sem considerar as restrições de capacidade de cada arco.

A partir da solução inicial o problema pode então ser resolvido pela aplicação de um algoritmo simplex especializado para redes, descritos em [Ken80] e [Gri86].

A implementação desenvolvida segue a idéia do algoritmo primal particionado sem restrições de capacidade agregada equivalente a ter uma matriz ciclo nula já que, nesse estágio, somente são resolvidos problemas de monoproduto sem restrições de agregação.

O procedimento utilizado no primeiro estágio na obtenção da solução ótima para os problemas monoproduto é descrito a seguir.

Procedimento 5.1 Obtenção da solução ótima para os problemas monoproduto, sem restrições de capacidades agregadas.

P1. A solução inicial é obtida alocando-se a demanda na árvore geradora mínima. Faz-se necessária a criação de arcos artificiais paralelos aos arcos que excederam sua capacidade, de modo que, o valor excedente seja colocado sobre o arco artificial.

P2. O algoritmo primal particionado (sem restrições de capacidade agregada) é executado para obter a solução ótima para o problema com múltiplos produtos *MP* ou *MPC*.

P3. Os arcos artificiais são removidos.

Cabe salientar que, no problema *MPC*, o procedimento utilizado acima garante a proporcionalidade na obtenção do composto *m*. Se o custo associado aos arcos do composto *m* é o mesmo aplicado a cada um de seus componentes, e a quantidade de demanda alocada obedece à fração do composto *m* em relação ao produto *k* então a proporcionalidade é mantida. Dessa forma, o novo modelo de rede descrito no capítulo IV, permite que os arcos das frações do composto *m* tenham a mesma representação física na árvore geradora mínima dos *k* produtos que o compõem.

5.1.2 Estágio II

Ao resolver de forma desacoplada cada um dos problemas monoproduto, define-se na solução ótima de cada rede uma partição dos arcos em conformidade com as variáveis básicas e não básicas. Caso esta solução satisfaça as restrições de agregação de fluxo, onde poderão estar incluídas as restrições de proporcionalidade associadas aos produtos compostos, ter-se-á obtida a solução ótima do problema, ou seja, a solução ótima do problema original satisfaz as restrições de acoplamento com folgas positivas. Então esta é a solução ótima para o problema com restrições de acoplamento. Entretanto, em geral, a substituição das variáveis estruturais básicas e não básicas nas restrições de acoplamento não satisfazem a condição para as variáveis de folga $s_i \geq 0$. Existem valores de folgas negativos e os custos reduzidos, depois de inseridas restrições de agregação, satisfazem a regra de parada para o simplex com particionamento primal. A iteração dual é realizada para solucionar, via método dual, um problema particionado primal cuja base inicial viável não é disponível.

Após inseridas as restrições de agregação, tem-se o sistema:

$$\text{Min } z = \begin{bmatrix} c_B^1 & c_B^2 & \cdots & c_B^K & | & 0 \end{bmatrix} \begin{bmatrix} x_B^1 \\ x_B^2 \\ \vdots \\ x_B^K \\ \dots \\ s_B \end{bmatrix} + \begin{bmatrix} c_N^1 & c_N^2 & \cdots & c_N^K & | & 0 \end{bmatrix} \begin{bmatrix} x_N^1 \\ x_N^2 \\ \vdots \\ x_N^K \\ \dots \\ s_N \end{bmatrix}$$

$$s.a: \begin{bmatrix} B^1 & 0 & \cdots & 0 & 0 \\ 0 & B^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & B^K & 0 \\ \hline D_B^1 & D_B^2 & \cdots & D_B^K & I \end{bmatrix} \begin{bmatrix} x_B^1 \\ x_B^2 \\ \vdots \\ x_B^K \\ s_B \end{bmatrix} + \begin{bmatrix} N^1 & 0 & \cdots & 0 \\ 0 & N^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N^K \\ \hline D_N^1 & D_N^2 & \cdots & D_N^K \end{bmatrix} \begin{bmatrix} x_N^1 \\ x_N^2 \\ \vdots \\ x_N^K \\ s_N \end{bmatrix} = \begin{bmatrix} r^1 \\ r^2 \\ \vdots \\ r^K \\ b \end{bmatrix}$$

onde

$D^B \leftarrow [D_B^1 \ D_B^2 \ \cdots \ D_B^K]$, $D^N \leftarrow [D_N^1 \ D_N^2 \ \cdots \ D_N^K]$ indicam as restrições de proporcionalidade e de capacidade agregada.

Suponha que a base B é a base ótima antes de inserir a restrição de agregação de índice i , $D_i x \leq b_i$, onde se faz corresponder as seguintes partições:

$$D_i = [D_i^B \mid D_i^N], \text{ para alguma restrição de índice } i.$$

$$x = [x^B \mid x^N]$$

O sistema correspondente é

$$\begin{aligned} z + (c^B B^{-1} N - c^N) x^N &= c^B B^{-1} r \\ x^B + B^{-1} N x^N &= B^{-1} r. \end{aligned} \quad (5.1)$$

A restrição de agregação $D_i x \leq b_i$ pode ser reescrita como

$$D_i^B x^B + D_i^N x^N + s_i = b_i \quad (5.2)$$

onde, D_i foi decomposto em $D_i = [D_i^B \mid D_i^N]$ com s_i uma variável de folga positiva correspondente à restrição i .

Multiplicando a equação (5.1) por D_i^B tem-se

$$D_i^B x^B + D_i^B B^{-1} N x^N = D_i^B B^{-1} r. \quad (5.3)$$

Subtraindo (5.3) de (5.2) tem-se o seguinte sistema

$$(D_i^N - D_i^B B^{-1} N) x^N + s_i = b_i - D_i^B B^{-1} r. \quad (5.4)$$

A equação (5.4) fornece a solução básica para o novo sistema. A única possibilidade de violação da otimalidade para o problema com restrições de acoplamento é o sinal de $b_i - D_i^B B^{-1} r$. Se, $b_i - D_i^B B^{-1} r \geq 0$ então a solução corrente é ótima. Caso contrário, se $b_i - D_i^B B^{-1} r < 0$, então o método simplex particionado dual é utilizado para restaurar a viabilidade.

5.2 MÉTODO SIMPLEX COM PARTICIONAMENTO DUAL

Descreve-se, a seguir, iteração dual que conduz à solução ótima, para o problema particionado. Resultados algébricos são mostrados para obter as r -ésimas linhas da matriz particionada $(\bar{B})^{-1}$ depois de inseridas as restrições de capacidade agregada que em geral conduzem a uma solução inviável do ponto de vista primal.

A matriz \bar{B}^{-1} é expressa em função da matriz inversa ciclo Y^{-1} apresentada em (3.34), cujo particionamento é o mesmo da proposição 12.

Devido à grande esparsidade da matriz inversa ciclo e a maneira eficiente de obtê-la é que a determinação de $\mathbf{e}_r(\bar{B})^{-1}\bar{\mathbf{a}}_j$ torna-se possível, de modo a permitir a restauração da viabilidade através de iterações duais. Além do mais, computar $\mathbf{e}_r(\bar{B})^{-1}$ (r -ésima linha⁵ de $(\bar{B})^{-1}$) é tarefa não muito fácil sob o ponto de vista computacional, no entanto computar $\mathbf{e}_r(\bar{B})^{-1}\bar{\mathbf{a}}_j$ é realizado de modo convenientemente simples, através de inspeção no grafo.

As colunas das variáveis não básicas candidatas a entrar na base serão denotadas por:

$$\bar{\mathbf{a}}_j = \begin{bmatrix} \alpha^k \\ \mu^k \end{bmatrix}$$

sendo a partição:

$$\mu^k = \begin{bmatrix} \beta^k \\ \gamma^k \end{bmatrix}, \text{ quando a variável que entra na base corresponder a um arco e;}$$

$$\mu^k = \begin{bmatrix} \beta^k \\ 0 \end{bmatrix}, \text{ quando a variável que entra na base corresponder a uma folga pertencente a uma}$$

restrição ativa de agregação.

Os valores $\bar{y}_r = \mathbf{e}_r(\bar{B})^{-1}\bar{\mathbf{a}}_j$ a serem determinados são particionados da forma

$$\bar{y}_r = \begin{bmatrix} \bar{y}^{1k} \\ \bar{y}^2 \end{bmatrix}$$

sendo a partição:

$$\bar{y}^2 = \begin{bmatrix} \bar{y}^{2k} \\ \bar{y}^3 \end{bmatrix}$$

⁵ A notação \mathbf{e}_r é utilizada para referir-se ao vetor linha canônico.

\bar{y}^{1k} é o valor obtido identificando a r -ésima linha correspondente ao arco do k -ésimo produto que sai da base;

\bar{y}^{2k} é o valor obtido identificando a r -ésima linha correspondente ao arco complementar do k -ésimo produto que sai da base. Essa linha corresponde a uma restrição ativa de agregação.

\bar{y}^3 é o valor obtido identificando a r -ésima linha correspondente a uma folga que sai da base. Essa linha corresponde a uma restrição não ativa de agregação.

Considera-se inicialmente a determinação de \bar{y}^3 . Ao inserir as restrições de agregação depois de obtidas as soluções ótimas dos problemas monoproduto, a variável de folga associada a alguma restrição de capacidade agregada pode ser negativa, violando a condição para $s_i \geq 0$. Dessa forma, o valor b_i mais negativo (correspondente à variável de folga que sai da base) é identificado e o valor \bar{y}^3 é determinado.

A nova base \bar{B} depois de inseridas as restrições de acoplamento passa a ser constituída pelas variáveis básicas que compõem as árvores geradoras mínimas relativas aos k produtos e pelas variáveis de folga nas restrições de agregação. Algumas variáveis não básicas encontram-se no limite inferior, enquanto que outras se encontram no limite superior. Salienta-se que a base obtida, depois de inseridas às restrições de acoplamento, não possuem variáveis super-básicas (arcos complementares) e, portanto, ainda não existem restrições ativas.

Como nessa etapa não existem folgas fora da base, o arco escolhido para entrar na base torna-se um arco complementar associado à folga que sai da base. A partir dessa iteração tem-se um problema particionado com restrições ativas e não ativas de agregação.

A seguir são determinados os valores \bar{y}_r para todos os arcos e folgas candidatos a entrar na base.

O produto da r -ésima linha no segundo bloco da matriz \bar{B}^{-1} por algum arco candidato a entrar na base é dada por

$$\begin{bmatrix} \bar{y}^{2k} \\ \bar{y}^3 \end{bmatrix} = \mathbf{e}_r \left[-Y^{-1} D^k (B^k)^{-1} \ ; \ Y^{-1} \begin{bmatrix} \alpha^k \\ \mu^k \end{bmatrix} \right]. \quad (5.5)$$

Os valores \bar{y}^2 são determinados buscando a r -ésima linha (correspondente a um arco complementar ou folga inviável) da expressão (3.39), dada por:

$$\bar{y}^2 = \mathbf{e}_r Y^{-1} [\mu^k - D^k (B^k)^{-1} \alpha^k], \forall k \quad (5.6)$$

O produto $\mathbf{e}_r Y^{-1}$ é a linha inversa da matriz Y^{-1} correspondente à folga ou arco complementar inviável.

Como a matriz Y^{-1} é bastante esparsa o cálculo de \bar{y}^2 é determinado pela diferença

$$\bar{y}^2 = \mathbf{e}_r Y^{-1} \mu^k - \mathbf{e}_r Y^{-1} D^k (B^k)^{-1} \alpha^k \quad (5.7)$$

O produto $\mathbf{e}_r Y^{-1} D^k (B^k)^{-1} \alpha^k$ tem a forma apropriada. Portanto buscam-se somente as restrições agregadas em correspondência com os elementos não nulos $\mathbf{e}_r Y^{-1}$ (correspondente à variável que sai da base). É na garantia da obtenção de \bar{y}^{2k} ou \bar{y}^3 , através dos elementos não nulos de $\mathbf{e}_r Y^{-1}$ e a correspondência destes com as restrições de agregação, que reside toda eficiência computacional nas iterações duais, já que, \bar{y}_r é determinado para todos os possíveis arcos dos k -ésimos produtos e todas as possíveis folgas candidatas a entrar na base.

O produto da r -ésima linha no segundo bloco da matriz \bar{B}^{-1} por uma coluna não básica correspondente a uma variável de folga s_i candidata a entrar na base é dada por:

$$y^3 = \mathbf{e}_r \begin{bmatrix} -Y^{-1} D^k (B^k)^{-1} \vdots Y^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ \dots \\ \mu_i \end{bmatrix}$$

$$y^3 = \mathbf{e}_r Y^{-1} [\mu_i] = e_r Y^{-1} \quad (5.8)$$

onde $\mu_i = 1$ é o coeficiente para a folga s_i .

Portanto, é fácil observar que a equação (5.8) que determina \bar{y}_r para folgas s_i , é um caso particular da equação (5.6) quando $(B^k)^{-1} \alpha^k = 0$.

Procedimento 5.2 *Iteração dual realizada sobre uma variável de folga ou arco complementar inviável*

P1. Identifica-se a r -ésima linha da matriz Y^{-1} correspondente à variável inviável (folga ou arco complementar).

P2. Se a variável (arco ou folga) possui sua capacidade mínima violada, então passo P3. Senão, passo P4.

P3. Calcula-se \bar{y}_r através de (5.6) para todos os arcos e folgas que se encontram fora da base. Se não existe $\sigma(\bar{y}_r) = -\delta$, então pare. O dual é ilimitado e o primal é inviável. Em caso contrário, passo P5.

P4. Calcula-se \bar{y}_r através de (5.6) para todos os arcos e folgas que se encontram fora da base.

Se não existe $\sigma(\bar{y}_r) = \delta$, então pare. O dual é ilimitado e o primal é inviável. Em caso contrário, passo P5.

P5. Escolhe-se a variável que entra na base realizando o seguinte teste de razão:

$$\text{Calcule } \min \alpha_N = \min_i \left\{ \frac{\Delta z_i}{\bar{y}_r} \right\}, \text{ determinando quem entra na base.}$$

Suponha agora que, em alguma iteração é necessário obter a r -ésima linha correspondente a um arco inviável para k -ésimo produto. Para cada partição R^k não vazia, existe uma correspondência entre um arco complementar e uma folga pertencente a uma restrição ativa de agregação. Para obter a r -ésima linha correspondente a um arco inviável é necessário inicialmente obter as r -ésimas linhas inversas da matriz ciclo Y associadas às restrições ativas de agregação para o k -ésimo produto. Então, para cada produto k , que contenha arcos complementares, a equação (5.6) passa ter a seguinte notação com relação a \bar{y}^{2k} .

$$\begin{aligned} \bar{y}^{21} &= \mathbf{e}_{R_1} Y^{-1} [\mu^1 - D^1 (B^1)^{-1} \alpha^1] \\ \bar{y}^{22} &= \mathbf{e}_{R_2} Y^{-1} [\mu^2 - D^2 (B^2)^{-1} \alpha^2] \\ &\vdots \\ \bar{y}^{2k} &= \mathbf{e}_{R_k} Y^{-1} [\mu^k - D^k (B^k)^{-1} \alpha^k] \end{aligned} \quad (5.9)$$

onde \mathbf{e}_{R_k} é a matriz cujas linhas correspondem ao vetor canônico com entrada não nula correspondente às restrições ativas de agregação associadas aos arcos complementares da partição R^k .

A forma eficiente de trabalhar com a matriz ciclo Y^l de dimensão fixa é que os elementos não nulos de $\mathbf{e}_{R_k} Y^{-l}$ estão em correspondência com as restrições ativas de agregação necessárias na determinação de \bar{y}^{lk} .

Denota-se por

e_{j_s} arco complementar correspondente a s -ésima coluna de R^k ;

$\mathbf{e}_{R_k} Y^{-l}$ r -ésimas linhas ativas da matriz Y^l associadas aos arcos complementares para o k -ésimo produto;

$\mathbf{e}_r (B^k)^{-l} \alpha^k$ orientação do arco que sai no ciclo formado com o arco que entra;

$\mathbf{e}_r(B^k)^{-1}R^k$ orientação do arco que sai nos ciclos formados com os arcos complementares e_{j_s} .

Seja a r -ésima linha de $(\bar{B})^{-1}$ correspondente um arco inviável do k -ésimo produto no primeiro bloco da partição

$$\mathbf{e}_r \bar{B}^{-1} = \mathbf{e}_r \left[(B^k)^{-1} R^k \mathbf{e}_{R_k} Y^{-1} D^l (B^l)^{-1} \mid \dots \mid (B^k)^{-1} + (B^k)^{-1} R^k \mathbf{e}_{R_k} Y^{-1} D^k (B^k)^{-1} \mid - (B^k)^{-1} R^k \mathbf{e}_{R_k} Y^{-1} \right] \quad (5.10)$$

e os possíveis arcos e_k candidatos a entrar na base, cujas colunas não básicas é representada por:

$$\bar{\mathbf{a}}_k^1 = \begin{bmatrix} \alpha^1 \\ \vdots \\ 0 \\ \mu^1 \end{bmatrix}, \quad \bar{\mathbf{a}}_k^2 = \begin{bmatrix} 0 \\ \alpha^2 \\ \vdots \\ \mu^2 \end{bmatrix}, \quad \dots, \quad \bar{\mathbf{a}}_k^K = \begin{bmatrix} 0 \\ \vdots \\ \alpha^k \\ \mu^k \end{bmatrix}$$

com

$$\bar{y}^{2k} = \mathbf{e}_{R_k} Y^{-1} [\mu^k - D^k (B^k)^{-1} \alpha^k], \text{ correspondente aos arcos complementares } e_j^k.$$

Considere os valores

$$\begin{aligned} \bar{y}^{11} &= \mathbf{e}_r (\bar{B})^{-1} \bar{\mathbf{a}}_k^1 = \mathbf{e}_r \left[(B^k)^{-1} R^k \mathbf{e}_{R_k} Y^{-1} D^l (B^l)^{-1} \alpha^1 - (B^k)^{-1} R^k \mathbf{e}_{R_k} Y^{-1} \mu^1 \right] \\ &= \mathbf{e}_r \left[- (B^k)^{-1} R^k \mathbf{e}_{R_k} Y^{-1} (\mu^1 - D^l (B^l)^{-1} \alpha^1) \right] = \mathbf{e}_r \left[- (B^k)^{-1} R^k \bar{y}^{21} \right] \end{aligned} \quad (5.11)$$

$$\begin{aligned} \bar{y}^{12} &= \mathbf{e}_r (\bar{B})^{-1} \bar{\mathbf{a}}_k^2 = \mathbf{e}_r \left[- (B^k)^{-1} R^k \mathbf{e}_{R_k} Y^{-1} (\mu^2 - D^2 (B^2)^{-1} \alpha^2) \right] \\ &= \mathbf{e}_r \left[- (B^k)^{-1} R^k \bar{y}^{22} \right] \end{aligned} \quad (5.12)$$

\vdots

$$\begin{aligned} \bar{y}^{1k} &= \mathbf{e}_r (\bar{B})^{-1} \bar{\mathbf{a}}_k^K = \mathbf{e}_r \left[(B^k)^{-1} \alpha^k + (B^k)^{-1} R^k \mathbf{e}_{R_k} Y^{-1} D^k (B^k)^{-1} \alpha^k - (B^k)^{-1} R^k \mathbf{e}_{R_k} Y^{-1} \mu^k \right] \\ &= \mathbf{e}_r \left[(B^k)^{-1} \alpha^k - (B^k)^{-1} R^k \mathbf{e}_{R_k} Y^{-1} [\mu^k - D^k (B^k)^{-1} \alpha^k] \right] \\ &= \mathbf{e}_r \left[(B^k)^{-1} \alpha^k - (B^k)^{-1} R^k \bar{y}^{2k} \right]. \end{aligned} \quad (5.13)$$

A equação (5.13) é a diferença da orientação do arco que sai no ciclo formado com o arco e_k (candidato a entrar na base) pelo somatório do número real \bar{y}^{2k} que multiplica a orientação do arco que sai no ciclo formado com os arcos complementares correspondentes ao k -ésimo produto do arco que sai.

As equações (5.11) e (5.12) são casos particulares da equação (5.13), nesse caso o arco que sai, correspondente ao k -ésimo produto, não forma ciclo com o arco e_k que entra na base, isto é, tem-se $\mathbf{e}_r (B^k)^{-1} \alpha^k = 0$.

Folgas associadas a arcos complementares também podem ser candidatas a entrar na

base. Então o produto da partição (5.10) pela coluna não básica $\bar{\mathbf{a}}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ \mu_i \end{bmatrix}$ correspondente a

uma folga s_i é dada por:

$$\begin{aligned} \mathbf{e}_r(\bar{B})^{-1}\bar{\mathbf{a}}_i &= \mathbf{e}_r[-(B^k)^{-1}R^k e_r Y^{-1}\mu^i] \\ &= \mathbf{e}_r[-(B^k)^{-1}R^k \bar{y}^{2k}]. \end{aligned} \quad (5.14)$$

A equação (5.14) é também um caso particular da equação (5.13). Neste caso $\mathbf{e}_r(B_k^{-1})\alpha_k = 0$, com $\bar{y}^{2k} = \mathbf{e}_{R^k} Y^{-1}[\mu^i]$.

Na figura V-1, tem-se uma base composta por dois arcos complementares, um arco inviável que sai, um arco e_k (do mesmo produto do arco que sai) candidato a entrar na base.

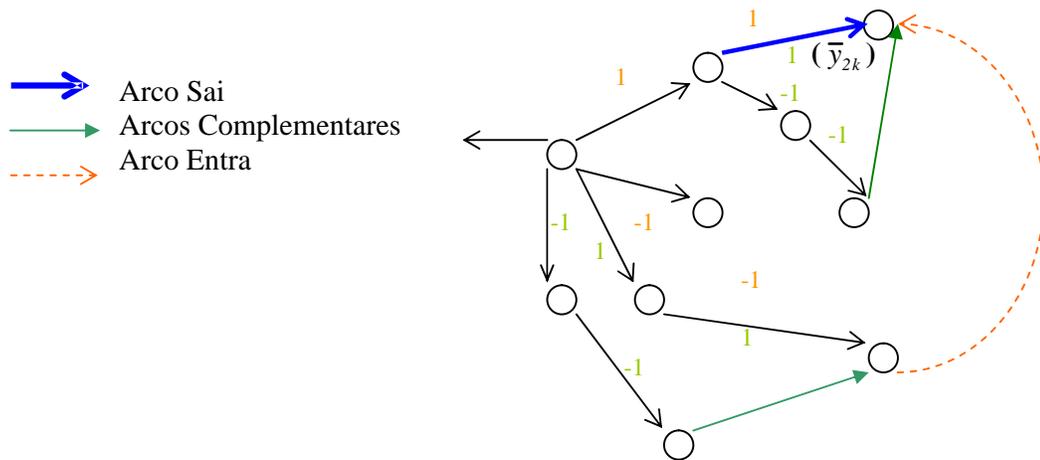


Figura V-1 - Orientação dos arcos básicos nos ciclos formados pelos arcos complementares e pelo arco que entra na base.

Deve ser observado que o arco que entra na base determina uma quantidade \bar{y}^{2k} (fator fluxo) que deve ser multiplicada pela orientação do arco que sai no ciclo do arco complementar que o contém. Nesse exemplo, somente um fator \bar{y}^{2k} é considerado visto que o arco que sai não forma ciclo com o outro arco complementar. Ainda, como o arco que entra pertence ao mesmo produto do arco que sai, então o fator fluxo que multiplica a orientação do arco que sai, nos ciclos formados pelos arcos complementares, é diminuído pela orientação do arco que sai no ciclo do arco que entra. Cada fator \bar{y}^{2k} , produzido por uma variável não

básica, deve ser multiplicado pela orientação do arco que sai, no ciclo formado com o arco complementar, determinando assim y^{lk} .

Identificada a variável que entra na base, a forma como são realizadas as operações de pivoteamento durante as iterações duais são as mesmas apresentadas no capítulo III. Na figura V-1, tem-se o caso 2 para operação de pivoteamento, nesse caso, o arco que entra substitui o arco que sai na árvore básica.

Procedimento 5.3 *Iteração dual realizada sobre um arco inviável*

- P1. Identifica-se o k -ésimo produto correspondente ao arco inviável que deve sair da base.
- P2. Se o arco possuir sua capacidade mínima violada, então passo P3. Senão, passo P4.
- P3. Calcula-se \bar{y}^{lk} através de (5.13) para arcos e \bar{y}^{lk} através de (5.14) para folgas. Se não existir $\sigma(\bar{y}_r) = -\delta$, então pare. O dual é ilimitado e o primal é inviável. Em caso contrário, passo P5.
- P4. Calcula-se \bar{y}^{lk} através de (5.13) para arcos e \bar{y}^{lk} através de (5.14) para folgas. Se não existir $\sigma(\bar{y}_r) = \delta$, então pare. O dual é ilimitado e o primal é inviável. Em caso contrário, passo P5.
- P5. Escolhe-se a variável candidata a entrar na base realizando o seguinte teste de razão:

$$\text{Calcule } \min \alpha_N = \min_i \left\{ \frac{Az_i}{\bar{y}^{lk}} \right\}, \text{ determinando quem entra na base.}$$

A iteração dual permite forçar um pivoteamento sobre uma linha inviável, o que não é uma operação válida para o algoritmo primal particionado. Apesar desse fato singular ocorrer no problema primal particionado, no correspondente dual as regras do método simplex estão sendo perfeitamente atendidas. Nesse caso, ao prosseguir aplicando o algoritmo dual chegar-se-á a solução ótima do problema dual, sugerindo que, o processo adaptado ao problema primal será também capaz de convergir para uma solução viável ótima. Em outras palavras, o problema primal particionado com restrições agregadas não possuía solução viável de partida, mas o dual particionado sim. Ao utilizar o algoritmo simplex dual para solucionar um problema primal cuja base inicial não era disponível, um novo algoritmo pode ser desenhado.

ALGORITMO 5.1 Algoritmo Particionado Dual (Problema de Minimização)

- P1. Encontre uma base particionada \bar{B} cuja solução é ótima sob o ponto vista primal.
- P2. Se $\bar{B}^{-1}r > 0 \quad \forall j$, pare. A solução corrente é ótima. Senão, selecione a r -ésima linha mais inviável determinando qual variável deverá sair da base.
- P3. Execute o procedimento iteração dual.
- P4. Faça a operação de pivoteamento e retorne ao passo 1.

O exemplo que ilustra o algoritmo acima é mostrado no próximo capítulo. A solução inicial para o problema particionado dual é obtida a partir da solução ótima dos problemas monoproduto. O exemplo considerado é o mesmo exemplo que ilustra o algoritmo particionado primal (ver Apêndice I) que utiliza a matriz inversa ciclo Y^l .

5.3 CONSIDERAÇÕES SOBRE A REALIZAÇÃO DAS ITERAÇÕES DUAS

O aspecto fundamental abordado durante o processo de desenvolvimento dos procedimentos para realização das iterações duais foi o aspecto computacional. Como e quais os instrumentos numéricos que seriam utilizados para tornar o sistema computacional mais eficiente, como os fundamentos matemáticos do método dual simplex seriam aplicados. A implantação e avaliação do algoritmo de programação linear, capaz de resolver problemas de grande porte, utilizando o particionamento dual é viável a partir das estruturas de armazenamento de dados, das estruturas dos subprogramas minimizando a transmissão de dados entre a memória principal e memória auxiliar dos computadores atuais.

Uma preocupação constante tida ao longo do processo de desenvolvimento das iterações duais foi, a otimização do processo computacional, no sentido de economizar em memória e tempo de CPU.

Considerando o grau de esparsidade da matriz inversa ciclo Y e o elevado número de arcos a serem pesquisados para efetuar o teste de razão para o problema dual foram desenvolvidos procedimentos com maior eficiência computacional.

O valor da diferença $[\mu_i^k - D_i^k (B^k)^{-1} \alpha^k]$ obtido para cada restrição de índice i através da inspeção no ciclo formado com o arco não básico, possui grande peso

computacional quando a inspeção é realizada para todas as restrições. Dessa forma, (5.6) passa ser computada através de (5.7), a matriz inversa ciclo é armazenada em forma de listas através da memorização dos elementos não nulos que estão em correspondência com as restrições de agregação de índice i . O tempo computacional é reduzido pelo fato que o espaço das restrições a serem pesquisadas torna-se menor.

A outra possibilidade de redução de tempo computacional nas iterações duais é com relação aos arcos não básicos que possuem o valor $\bar{y}_r = 0$ e portanto não fazem parte do teste de razão. Existe uma diferença significativa de tempo de processamento entre obter o valor $\bar{y}_r = 0$ do que afirmar através da inspeção no grafo que o valor \bar{y}_r é nulo. A seguir é ilustrado quando o cálculo de \bar{y}_r se faz necessário.

A figura V-2 representa uma árvore básica para o produto A com x_1 e x_2 representando dois arcos não básicos, x_3 um arco básico pertencente à restrição ativa de agregação. A figura V-3 representa uma árvore básica para o produto B contendo um arco complementar x_4 associado à restrição ativa de agregação e um arco inviável x_5 .

O arco x_2 poderá produzir um valor $\bar{y}_r \neq 0$ forçando o pivoteamento sobre o arco inviável x_5 . Ao incidir uma unidade a mais de fluxo no ciclo formado pelo arco x_2 faz com que o fluxo no arco x_3 pertencente à restrição seja modificado, em consequência disto, os fluxos nos arcos pertencentes ao ciclo formado pelo arco complementar são modificados, o arco inviável sai da base, o arco complementar torna-se arco da árvore e o arco x_2 torna-se arco complementar.

A árvore básica para o produto A, a partir do arco pertencente a restrição é dividida em duas sub-árvores T_A e T_B como mostra figura V-3. Se a origem e destino do arco x_2 pertencerem a sub-árvores diferentes, então o valor \bar{y}_r correspondente a x_2 deve ser calculado, o arco x_2 será incluído no teste de razão.

Por outro lado o valor \bar{y}_r correspondente a x_1 não precisa ser determinado, não existe possibilidade do arco x_1 fazer parte do teste de razão, o arco x_1 não pode forçar o pivoteamento sobre o arco inviável x_5 visto que os arcos que compõem o ciclo não pertencem a restrição ativa de agregação. A origem e destino do arco x_1 pertencem à mesma sub-árvore.

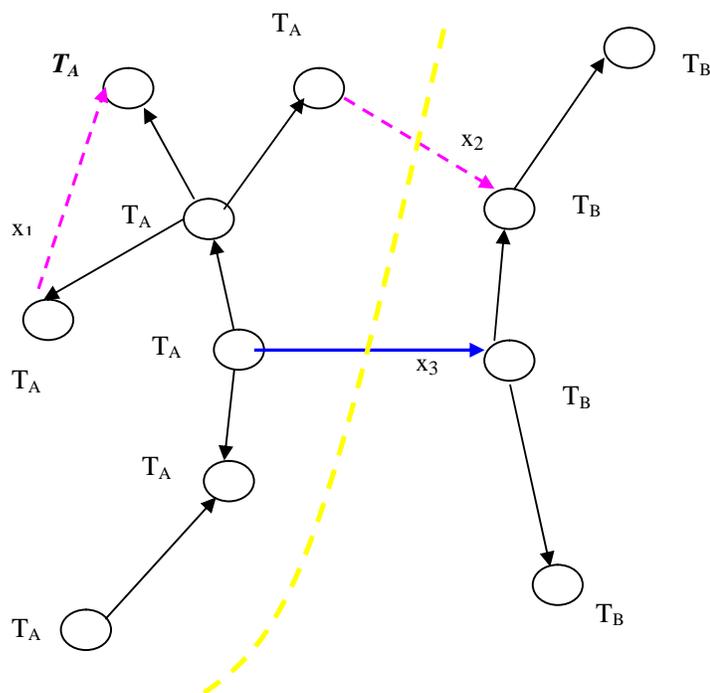


Figura V-2 - Árvore básica para o produto A dividida em duas sub-árvores

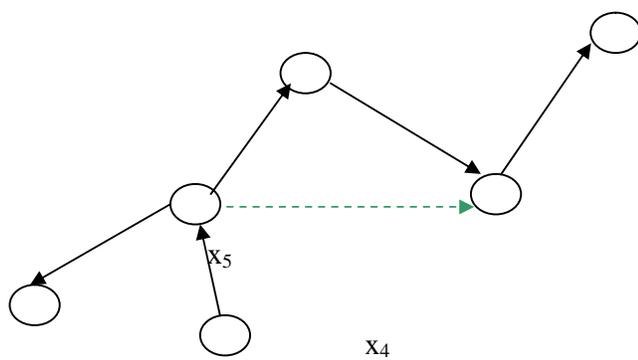


Figura V-3 - Árvore básica para o produto B

No próximo capítulo é apresentado um exemplo numérico onde os resultados matemáticos obtidos são aplicados na realização das iterações duais. O capítulo também é dedicado aos resultados computacionais.

CAPÍTULO VI

6. RESULTADOS NUMÉRICOS E COMPUTACIONAIS

Neste capítulo são apresentados resultados numéricos e computacionais para os problemas lineares de múltiplos produtos ao quais se pode incluir o transporte de produtos compostos. Um exemplo numérico mostra como as iterações duais são realizadas após inserção das restrições de acoplamento. Os problemas escolhidos para avaliar as potencialidades do algoritmo, com particionamento primal-dual, contém em torno de 200.000 restrições e 300.000 variáveis.

6.1 REALIZAÇÃO DAS ITERAÇÕES DUAIS SOBRE UMA REDE COM SUPER NÓS DE TRANSBORDO.

Como já mencionado, em muitas situações práticas é importante determinar a solução ótima para os problemas de múltiplos produtos com restrições de capacidade agregada a partir da solução ótima de cada um dos problemas monoproduto. Usualmente, é importante encontrar uma solução viável sob o ponto de vista dual com uma relação favorável entre qualidade e o tempo necessário para sua obtenção, isto é, buscar uma solução viável e satisfatória, em algum sentido, no menor tempo possível.

Através do exemplo apresentado no Apêndice I mostraram-se as iterações primais realizadas a partir de uma solução básica viável para o problema com particionamento primal. O mesmo exemplo é agora utilizado para mostrar as iterações duais que são realizadas após a inserção das restrições de acoplamento. É importante observar que no estágio I são resolvidos problemas de monoproduto. Dividir o processo para encontrar uma solução viável para os problemas (*MP* ou *MPC*) dentro de dois estágios provou ser muito eficiente em número de

iterações com relação ao método que inicia a partir de uma solução básica considerando o total das inviabilidades para todas as restrições.

As unidades de fluxo inicial nos arcos apresentados nas tabelas VI-1 e VI-2 representam o fluxo ótimo obtido no estágio I que não considera as restrições de capacidades agregadas.

Tabela VI-1 - Especificação dos Arcos para o Produto 1, cujo fluxo inicial é a solução ótima obtida no problema sem restrição de capacidade agregada.

ARCO	NÓ INICIAL	NÓ FINAL	CUSTO	FLUXO INICIAL	CAPACIDADE
x_1	R_1				
x_{11}	R_1	B_1	5.0	0	∞
x_{12}	R_1	B_2	6.0	60	∞
x_{13}	B_1	B_1	0.0	0	∞
x_{14}	B_2	B_2	0.0	60	∞
x_{15}	B_1	C_1	10.0	0	∞
x_{16}	B_1	C_2	8.0	0	∞
x_{17}	B_1	C_3	9.0	0	∞
x_{18}	B_2	C_1	4.0	20	∞
x_{19}	B_2	C_2	6.0	20	∞
x_{110}	B_2	C_3	7.0	20	∞

Tabela VI-2 - Especificação dos Arcos para o Produto 2, cujo fluxo inicial é a solução ótima obtida para o problema sem restrição de capacidade agregada.

ARCO	NÓ INICIAL	NÓ FINAL	CUSTO	FLUXO INICIAL	CAPACIDADE
x_2	U_1				
x_{21}	U_1	B_1	3.0	10	∞
x_{22}	U_1	B_2	4.0	5	∞
x_{23}	B_1	B_1	0.0	10	∞
x_{24}	B_2	B_2	0.0	5	∞
x_{25}	B_1	C_1	7.0	0	∞
x_{26}	B_1	C_2	8.0	10	∞
x_{27}	B_1	C_3	9.0	0	∞
x_{28}	B_2	C_1	6.0	5	∞
x_{29}	B_2	C_2	8.0	0	∞
x_{210}	B_2	C_3	10.0	0	∞

O custo ótimo obtido para o exemplo apresentado em E.I antes de inserir as restrições de capacidade agregada é 860. Após, inseridas as restrições $x_{13} + x_{23} + S_1 = 50$ e $x_{14} + x_{24} + S_2 = 40$, verifica-se que a folga S_2 é negativa e, portanto a iteração dual deve ser realizada até que todos os fluxos nos arcos e folgas sejam positivos para que a solução ótima do problema de minimização de custos seja encontrada.

As figuras VI-1 e VI-2 representam a árvore básica e os valores duais para ambos os produtos.

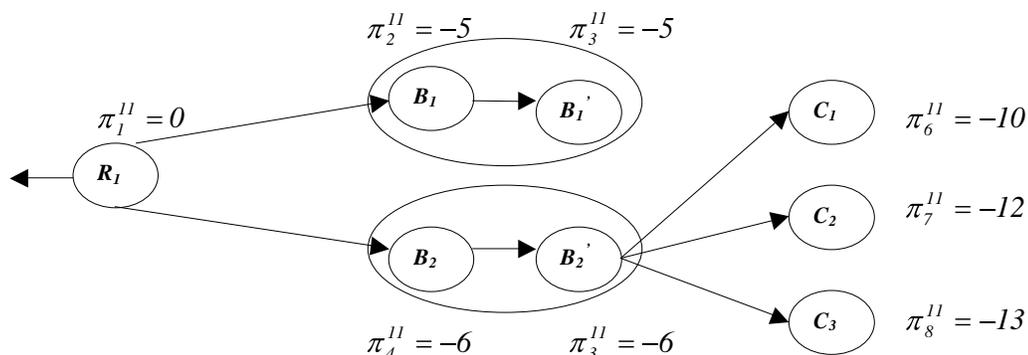


Figura VI-1 - Valores duais dos nós para o produto 1 sem restrição de agregação

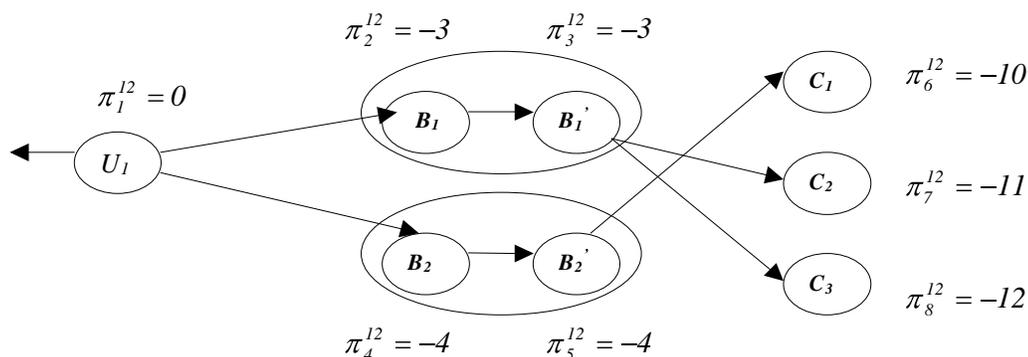


Figura VI-2 - Valores duais dos nós para o produto 2 sem restrição de agregação

Após, inseridas as restrições, sob o ponto de vista primal, não há mais candidatos a entrar na base. Os valores Δz para os arcos que se encontram no limite inferior (x_{15} , x_{16} , x_{17} , x_{25} , x_{29} e x_{210}) são respectivamente $(-5, -1, -1, 0, -1, -2)$ e, portanto tem-se a condição de parada para o problema primal. Contudo, esta solução não é viável sob o ponto de vista do primal, e iterações duais poderão ser realizadas para obtenção de uma solução viável.

Para realizar a iteração dual escolhe-se a folga $S_2 = -25$ para sair da base. Na primeira iteração não há folgas candidatas a entrar na base, e os arcos candidatas a entrar na base deverão ter $\bar{y}_r < 0$, pois todas as variáveis não básicas encontram-se no limite inferior ($x_N = 0$). A quantidade de fluxo Δ é razão entre o fluxo da folga negativa que sai por \bar{y}_{lk} , onde \bar{y}_{lk} é correspondente ao arco que entra.

Determina-se \bar{y}_r através de (5.6), onde $\mathbf{e}_r Y^{-1}$ é a r -ésima linha da matriz inversa ciclo correspondente a folga negativa S_2 . Então

$$\bar{y}_r x_{15} = \mathbf{e}_r Y^{-1} [\mu^k - D^k (B^k)^{-1} \alpha^k] = [0 \quad I] \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -I \\ I \end{bmatrix} \right\} = -I, \text{ e, portanto, candidato a entrar}$$

na base para o produto 1. Os valores obtidos para $\bar{y}_r x_{16}$ e $\bar{y}_r x_{17}$ com respeito ao produto 1 são iguais a -1 . Em relação ao produto 2, tem-se

$$\bar{y}_r x_{25} = \mathbf{e}_r Y^{-1} [\mu^k - D^k (B^k)^{-1} \alpha^k] = [0 \quad I] \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -I \\ I \end{bmatrix} \right\} = -I. \text{ Para os arcos } x_{29} \text{ e } x_{210}, \text{ tem-se}$$

se $\bar{y}_r = 1$ e portanto não fazem parte do teste de razão.

O teste de razão sugere que o arco x_{25} entra na base, pois $\alpha_k = \min_j \alpha_j = \min\{5, 1, 1, 0\} = 0$.

Portanto o arco x_{25} entra na base e se torna um arco complementar associado à folga S_2 que sai da base.

A quantidade de fluxo Δ é igual a 25 e a atualização dos fluxos para ambos os produtos é feito sobre o arco que entra x_{25} .

$$\text{Determina-se, } y^2 = Y^{-1} [\mu^k - D^k (B^k)^{-1} \alpha^k] = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -I \\ I \end{bmatrix} \right\} = \begin{bmatrix} I \\ -I \end{bmatrix} \leftarrow S_1, \text{ para}$$

atualização, respectivamente, das folgas S_1 e S_2 .

Tem-se, $y^{11} = 0$ não havendo atualização de fluxos no produto 1. E, obtém-se

$$y^{12} = (B^2)^{-1} \alpha^2 = \begin{bmatrix} x_2 & x_{21} & x_{22} & x_{23} & x_{24} & x_{26} & x_{27} & x_{28} \\ 0 & -I & I & -I & I & 0 & 0 & I \end{bmatrix} \text{ para atualização dos fluxos no}$$

produto 2.

Portanto, tem-se a seguinte atualização dos fluxos:

$$x_{25} \leftarrow 25$$

$$x_{21} = 10 - I(-1)25 = 35$$

$$x_{22} = 5 - I(1)25 = -20$$

$$x_{23} = 10 - I(-1)25 = 35$$

$$x_{24} = 5 - I(1)25 = -20$$

$$x_{28} = 5 - I(1)25 = -20$$

$$S_1 = 40 - I(1)25 = 15$$

$$S_2 = -25 - I(-1)25 = 0.$$

A árvore básica do produto 1 permanece a mesma. A árvore básica para o produto 2 contém o arco complementar x_{25} .

Na próxima iteração, deve ser calculado o valor dual associado à folga S_2 que se encontra fora da base e associada ao arco complementar x_{25} . Tem-se a matriz ciclo $Y = \begin{bmatrix} I & I \\ 0 & -I \end{bmatrix} = Y^I$. A primeira e segunda coluna desta matriz correspondem respectivamente à folga S_1 e ao arco complementar x_{25} (associado à folga S_2).

O valor dual $\pi^2 = 0$, pois $c^{22} - c^{12}(B^2)^{-1}\alpha^2 = 7 - (-3 + 4 + 6) = 0$, não havendo, portanto, correção dual.

Os valores duais não são corrigidos permanecendo os mesmos já mostrados na figura VI-2. A figura VI-3 reinterpreta esses valores duais, além do arco complementar associado à folga S_2 .

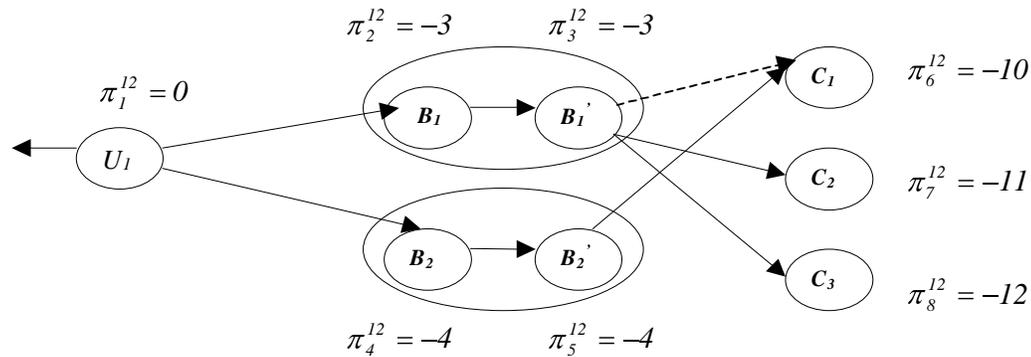


Figura VI-3 - Valores duais dos nós para o produto 2 na segunda iteração dual.

Dado que a solução primal ainda não é viável, realiza-se uma nova iteração dual onde um arco escolhido para sair da base apresenta fluxo negativo. Faz-se então, a escolha sobre o arco x_{28} .

Não há arcos candidatos a entrar na base em relação ao produto 2, pois os arcos x_{29} e x_{210} não formam ciclo com o arco negativo que sai. A folga S_2 associada ao arco complementar forma ciclo com o arco negativo que sai, portanto é candidata a entrar na base. No entanto, determinando-se $\bar{y}_r S_2$, através de (5.9), tem-se:

$$\bar{y}_r S_2 = -\mathbf{e}_r (B^2)^{-1} R^2 \mathbf{e}_{R_2} Y^{-1} \mu^2, \text{ onde}$$

$\mathbf{e}_r (B^2)^{-1} R^2$ → orientação do arco que sai x_{28} no ciclo formado com o arco complementar x_{25} do produto 2.

$\mathbf{e}_{R_2} Y^{-1}$ → r -ésima linha ativa correspondente ao arco complementar x_{25} do produto 2.

μ^2 → coeficiente da folga S_2 na restrição de agregação ativa.

$$\bar{y}_r S_2 = - (1) \begin{bmatrix} 0 & -I \\ 0 & I \end{bmatrix} = 1. \text{ Logo, } S_2 \text{ não pode entrar na base.}$$

Os arcos candidatos a entrar na base em relação ao produto 1 são os arcos x_{15} , x_{16} e x_{17} . Determinando, através de (5.9) tem-se

$$\bar{y}^{11} x_{15} = - \mathbf{e}_{R_2} (B^2)^{-1} R^2 \bar{y}^{21}, \text{ onde}$$

$\mathbf{e}_{R_2} (B^2)^{-1} R^2 = 1$, a orientação do arco negativo x_{28} no ciclo formado com o arco complementar x_{25} .

$\mathbf{e}_{R_k} Y^{-1} = [0 \quad -I]$, a r -ésima linha ativa correspondente ao arco complementar x_{25} para o produto 2.

$\mu^l \rightarrow$ coeficiente para o arco que entra na restrição de agregação.

$D^l (B^l)^{-1} \alpha^l \rightarrow$ somatório do produto dos coeficientes dos arcos básicos do produto 1 nas restrições de agregação pela sua orientação no ciclo formado com arco que entra x_{15} .

Assim,

$$\bar{y}^{21} = [0 \quad -I] \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -I \\ I \end{bmatrix} \right\} = [0 \quad -I] \begin{bmatrix} I \\ -I \end{bmatrix} = 1, \text{ com } \bar{y}^{11} x_{15} = -1.$$

Portanto, o arco x_{15} é candidato a entrar na árvore básica. Analogamente, são determinados os \bar{y}_r para os arcos x_{16} e x_{17} , obtendo $\bar{y}^{11} x_{16} = \bar{y}^{11} x_{17} = -1$.

O teste de razão sugere que o arco x_{16} entre na base, pois, $\alpha_k = \min_j \alpha_j = \min\{5,1,1\} = 1$. Com a saída do arco x_{28} , o arco x_{25} é buscado para substituir o arco que parte da base, deixando de ser arco complementar. O arco x_{16} que entra torna-se complementar associado à folga S_2 .

A quantidade de fluxo Δ é igual a 20 e a atualização dos fluxos para ambos os produtos é feita sobre o arco que entra x_{16} .

$$\text{Determina-se } y^2 = Y^{-1} [\mu^k - D^k (B^k)^{-1} \alpha^k] = \begin{bmatrix} I & I \\ 0 & -I \end{bmatrix} \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -I \\ I \end{bmatrix} \right\} = \begin{bmatrix} 0 \\ I \end{bmatrix} \leftarrow S_1, \text{ onde } y^{22} = 1$$

e $y^{21} = 0$.

$$\text{Portanto, } y^{11} = (B^1)^{-1} \alpha^1 = \begin{bmatrix} x_1 & x_{11} & x_{12} & x_{13} & x_{14} & x_{18} & x_{19} & x_{110} \\ 0 & -I & I & -I & I & 0 & I & 0 \end{bmatrix},$$

$$\text{e } y^{12} = -(B^2)^{-1} R^2 y^{22} = (B^2)^{-1} R^2 = \begin{bmatrix} x_2 & x_{21} & x_{22} & x_{23} & x_{24} & x_{26} & x_{27} & x_{28} \\ 0 & I & -I & I & -I & 0 & 0 & -I \end{bmatrix}.$$

Atualizando os fluxos, tem-se:

$$\begin{aligned}
 x_{16} &\leftarrow 20 & x_{21} &= 35 - I(1)20 = 15 \\
 x_{11} &= 0 - I(-1)20 = 20 & x_{22} &= -20 - I(-1)20 = 0 \\
 x_{12} &= 60 - I(1)20 = 40 & x_{23} &= 35 - I(1)20 = 15 \\
 x_{13} &= 0 - I(-1)20 = 20 & x_{24} &= -20 - I(-1)20 = 0 \\
 x_{24} &= 60 - I(1)20 = 40 & x_{28} &= 20 - I(-1)20 = 0 \\
 x_{19} &= 20 - I(1)20 = 0 & &
 \end{aligned}$$

As folgas S_1 e S_2 respectivamente iguais a 15 e 0.

Tem-se a solução ótima, pois todos os fluxos que circulam nos arcos e folgas são positivos, com custo ótimo igual a 880.

O exemplo apresentado ilustrou passo a passo a realização das iterações duais durante o estágio II.

6.2 TESTES NUMÉRICOS

O algoritmo descrito no capítulo 5 foi implementado em Delphi 7.0. Todos os testes foram realizados em um computador com processador AMD Athlon XP 2000 contendo 758 Mbytes de memória principal.

A fim de avaliar as potencialidades do algoritmo são apresentados na tabela VI.1 os resultados de alguns problemas utilizados para testes. Foram elaborados e resolvidos com êxito problemas contendo até 370.000 variáveis e 200.000 restrições. Na referida tabela consta tempo de CPU no primeiro e segundo estágio do algoritmo. Cabe salientar que no tempo de processamento da primeira fase do algoritmo está incluído o tempo de leitura e montagem da rede e na segunda fase o tempo de inserção das restrições de agregação.

Tabela VI.3 - Tempo de CPU para os dois estágios do Algoritmo

Nós	Arcos	Total de Variáveis	Total de Restrições	Tempo de processamento (1ª fase)	Total de Iterações (1ª fase)	Tempo de processamento (2ª fase)	Total de Iterações (2ª fase)	Total de Iterações
746	1.417	7.085	4.401	00:00:00.063	65	00:00:00.531	37	102
10.870	25.780	128.900	69.260	00:01:32	845	00:02:57	439	1.284
12.469	28.976	144.880	78.852	00:01:48	2.081	00:07:58	1.264	3.335
31.184	74.448	372.240	199.184	00:02:30	684	00:03:01	80	764

Como pode ser observado, através da tabela acima, o número de iterações na segunda fase do algoritmo é menor quando comparado com a primeira fase. No entanto, o tempo de processamento se torna maior na segunda fase do algoritmo, devido à inclusão das

restrições de agregação. O peso computacional exigido pela correção das inviabilidades através das iterações duais, bem como as operações de pivoteamento que se fazem necessárias na resolução de problemas de múltiplos produtos na segunda fase do algoritmo torna-se maior quando comparado com a primeira fase.

É importante salientar que as restrições de capacidade agregada atribuída aos problemas testes na avaliação do algoritmo são baseadas em sistemas de distribuição nos quais, em geral, as capacidades da maioria dos arcos são irrelevantes. Nestes casos, apenas uma parcela muito pequena em relação ao número total de arcos é considerada capacitada.

Os dois primeiros resultados correspondem a problemas com distribuição de produtos compostos, sendo que em 16 super nós de transbordo são permitidas formulações de produtos compostos. Foram atribuídas restrições de capacidade máxima de formulação do composto em dois super nós de transbordo, considerando como infinita a capacidade de formulação do composto nos 14 super nós restantes. O número de restrições de proporcionalidade é relativo à quantidade de componentes participantes na formulação do produto composto.

Com relação ao primeiro resultado foram considerados dois componentes para formulação do composto e no segundo resultado foram considerados quatro componentes para obtenção da mistura.

Os dois últimos resultados não consideram formulações de produtos compostos e, portanto não existe a presença de restrições de proporcionalidade; apenas restrições de capacidade agregada.

O fator mais importante que afeta o tempo computacional do método simplex é o número de restrições funcionais. Para este método, uma regra empírica comum para estimar o número de iterações é determinar o dobro do número de restrições funcionais, ver [Fre91]. Então, 8.802, 138.520, 157.704 e 398.368, nessa ordem, seria o número estimado de iterações para cada problema teste da tabela VI-1. Comparando o número estimado de iterações com o número total de iterações apresentadas na tabela VI-1, observa-se que o número de iterações obtidas nos problemas testes tende a ser significativamente menor.

Dessa forma, a avaliação do algoritmo, quanto à eficiência computacional, a partir de um desenvolvimento teórico baseado num método exato que utiliza um particionamento dual mostrou ser eficiente quando aplicado a problemas de médio e grande porte.

6.3 CONSIDERAÇÕES FINAIS

Os testes indicaram que o algoritmo desenvolvido é adequado. Pode-se dizer que sua eficiência computacional fica vinculada: à formulação do modelo de fluxo em rede ao problema da distribuição de produtos compostos; à abordagem modificada de implantação do método tradicional com particionamento primal a partir de uma matriz ciclo de dimensão não variável; a utilização de um método simplex primal-dual, especializado para problemas de fluxo em rede.

Contudo, uma avaliação mais criteriosa deveria ser realizada se houvessem condições de comparar os resultados obtidos com implementações realizadas por outros autores, que também desenvolveram algoritmos para problemas de fluxo em rede. Esta comparação apresenta algumas dificuldades de realização tendo em vista os seguintes aspectos:

- i) Não estão disponíveis publicamente os dados utilizados por outros autores, que apenas informam as dimensões do problema, os tempos computacionais e o número de iterações; para as diversas instâncias do problema.
- ii) Algoritmos implementados por autores distintos não apresentam, necessariamente, estratégias iguais, sendo as mesmas adequadas às estruturas dos modelos abordados por cada autor. Assim, o algoritmo desenvolvido por um autor não apresentará, necessariamente, bom desempenho em todos os modelos abordados pelos diferentes autores.

CAPÍTULO VII

7. COMENTÁRIOS, CONCLUSÕES E RECOMENDAÇÕES

7.1 COMENTÁRIOS

Algumas observações e dificuldades encontradas durante a fase de implementação do método com particionamento dual, devem ser ressaltadas:

- i) No cálculo da matriz ciclo Y , idealizou-se uma estrutura de armazenamento sem a preocupação de manter as colunas das partições R^k e folgas ordenadas. No entanto, encontraram-se algumas dificuldades quanto à recuperação das informações com relação à operação de pré-multiplicação ou pós-multiplicação de um vetor pela inversa da matriz ciclo armazenada na forma produto.
- ii) A aplicação do algoritmo por meio da matriz ciclo de dimensão fixa exigiu uma atualização desta matriz a cada iteração, pois são determinados os valores y 's no teste de razão para arcos complementares e folgas simultaneamente. Contudo, quando realizada adequadamente esta operação da forma produto da inversa, a re-inversão da matriz ciclo Y refere-se ao acréscimo de uma matriz elementar na seqüência de matrizes elementares cujo produto forma Y^l .
- iii) Os problemas com a estabilidade numérica foram resolvidos através da avaliação dos erros de precisão.
- iv) A ordem lexicográfica foi utilizada para não ocorrer problemas de ciclagem.

- v) Houve a preocupação constante no sentido de economizar memória e tempo de CPU. Inicialmente implementou-se um algoritmo com particionamento primal-dual, apenas em um único estágio. Após rápida análise, concluiu-se, como Castro e Nabona [Cas96], que esse não era o caminho mais razoável devido ao peso computacional causado pelo método que já inicia considerando o total das inviabilidades para todas as restrições.
- vi) Era de grande importância que todos os casos de pivoteamento fossem testados. Nesse sentido, problemas testes foram elaborados e os procedimentos baseados na inspeção no grafo com respeito à entrada e saída de variáveis de um mesmo produto ou de diferentes produtos, foram todos observados.

7.2 CONCLUSÕES

- i) Uma tradução contextual conveniente do problema em questão foi capaz de identificar os elementos fundamentais do *MPC*, caracterizando-o como uma extensão do *MP*. Assim, foi possível a utilização de estratégias baseadas em técnicas de solução para este problema.
- ii) Considerando que, o número estimado de iterações do método simplex tende a ser o dobro do número de restrições funcionais conforme citado por diversos autores, então, sem dúvida nenhuma o algoritmo proposto é eficiente. O número de iterações necessárias na resolução dos problemas *MPC* por meio da estratégia de solução proposta tende a ser uma fração relativamente pequena desta estimativa.
- iii) O problema *MPC* fica caracterizado por incluir restrições de proporcionalidade nas restrições de agregação de fluxos. A reformulação do modelo na busca de uma solução consiste basicamente da representação lógica da rede que não representa o produto composto, mas sim a fração do produto composto. Pode-se dizer que, nesse fato residiu a possibilidade da aplicação da técnica de particionamento na resolução dos problemas com distribuição de produtos compostos, mantendo baixo o número de restrições de acoplamento.

- iv) A recuperação das informações necessárias no cálculo do teste de razão no algoritmo com particionamento dual têm um custo computacional muito alto. No entanto, a forma utilizada neste trabalho com relação à seleção dos arcos que fazem parte no teste de razão bem como o cálculo de \bar{y}_r , têm um papel importante e possibilitam a realização das iterações duais com um bom desempenho computacional.
- v) A obtenção da solução inicial através da heurística proposta, neste trabalho, fez com que o número de iterações fosse sensivelmente reduzido, contribuindo para que o objetivo geral fosse alcançado. Neste sentido, deve-se lembrar ainda, que a divisão da estratégia de solução em etapas também foi importante, pois deixa para tratar o problema com seus vários produtos e restrições de acoplamento apenas na última etapa, onde poucas iterações são realizadas envolvendo todos os produtos simultaneamente.
- vi) Deve-se ressaltar, entretanto, que a estratégia de solução adotada é apropriada para os problemas aqui testados, que foram construídos a partir de situações reais encontradas na distribuição de produtos compostos de uma indústria do setor petroquímico. Eventualmente, o desempenho aqui observado poderá não ocorrer para o caso de problemas em que as restrições de proporcionalidade das frações que compõem o produto composto devam ser explicitadas em todos os arcos da rede, bem como o caso em que o número relativo de arcos capacitados é significativamente maior.
- vii) Finalmente, considerando a natureza não polinomial do problema que em última instância é um caso particular de programação linear, deve-se lembrar que o tempo computacional gasto na solução dos mesmos não é deterministicamente calculado em função das dimensões do problema. Fatores como, custos, demandas e capacidades associadas ao problema, também contribuem para o tempo total de solução. Em outras palavras, dependendo da distribuição da demanda sobre a rede e da capacidade dos arcos, os tempos computacionais poderão sofrer alterações significativas.

7.3 RECOMENDAÇÕES

Este trabalho não esgota os estudos sobre o método de particionamento aplicado à distribuição de múltiplos produtos. Os resultados obtidos até aqui são encorajadores, sendo necessário recomendar estudos posteriores através das seguintes investigações:

- i) Comparação do método proposto com uma abordagem de métodos de pontos interiores.
- ii) Um aspecto muito importante é a avaliação do comportamento do algoritmo com relação à carga tributária imposta. Tributos dos mais variados tipos e formas de incidência podem fazer com que a estratégia logística seja totalmente modificada.
- iii) Em novas situações, relacionadas com a distribuição de produtos compostos, verificar as características da rede, a fim de se constatar adequação do uso da estratégia de solução proposta no trabalho.
- iv) Uma investigação maior, sobre o tempo de processamento das rotinas de inspeção no grafo, deve ser realizada no futuro.

BIBLIOGRAFIA

- [Ali80] ALI, A.; HELGASON, R.; KENNINGTON, J. e LALL, H. (1980). Computational Comparison among Three Multicommodity Network Flow Algorithms, *Operations Research*, **28**, 23-45.
- [Ass78] ASSAD, A.A. (1978). Multicommodity Network Flows a Survey, *Networks*, **8**, 37-91.
- [Bar86] BARR, R.S.; FARHAAGIAN, K. e KENNINGTON (1986). Networks with side constraints: An LU factorisation update, *The Annals of the Society of Logistics Engineering*, **1**, 66-85.
- [Bar97] BARNHART, C.; HANE, C.A. e VANCE, P. Integer multicommodity flow problems. *Lecture Notes in Economics and Mathematical System*, **450**, 17-31.
- [Bas88] BAASE, S. (1988). *Computer Algorithms*, 2.ed. Addison-Wesley.
- [Baz77] BAZARAA, M.S.; JARVIS, J.J. e SHERALI, H.D (1977). *Linear Programming and Network Flows*, John Wiley & Sons, New York.
- [Bel71] BELLMORE, M.; BENNINGTON, G. e LUBORE, S. (1971). A Multivehicle Tanker Scheduling Problem, *Trans. Science*, **5**, 36-47.
- [Ber85] BERTSEKAS, D.P. (1985). Unified Framework for Primal-Dual Methods in Minimum Cost Network Flow Problems, *Mathematical Programming*, **32**, 125-145.
- [Bie98] BIENSTOCK, D.; CHOPRA, S.; GUNLUK, O. e TSAI, C.Y. (1998). Minimum Cost Capacity Installation for Commodity Network Flows, *Mathematical Programming*, **81**, 177-199.
- [Bir65] BIRKHOFF, G. (1965). *A Survey of Modern Algebra*, 3.ed., Macmillan.
- [Bra77] BRADLEY, G.H.; BROWN, G.G. e GRAVES, G.W. (1977). Design Implementation of Large Scale Primal Transshipment Algorithms, *Management Science*, **24**, 1-34.
- [Cap03] CAPPANERA, P. e FRANGIONI, A. (2003). Symetric and Asymetric Parallelization of a Cost-Decomposition Algorithm for Multicommodity Flow Problems, *Informes Journal on Computing*, **15**, 369-384.

- [Cas00] CASTRO, J. (2000). A Specialized Interior-Point Algorithm for Multicommodity Network Flow, *Journal on Optimization*, **10**, 852-877.
- [Cas01] CASTRO, J. e FRANGIONI, A. (2001). A Parallel Implementation of an Interior-Point Algorithms for Multicommodity Network Flow, *Lecture Notes in Computer Science* **1981**, 301-315.
- [Cas03] CASTRO, J. (2003). Solving Difficult Multicommodity Problems with a Specialized Interior-Point Algorithm, *Annals of Operations Research*, **124**, 35-48.
- [Cas95] CASTRO, J. e NABONA, N. (1995). Primal-Dual Interior Point Method for Multicommodity Network Flows with Side Constraints and Comparison with Alternative Methods, *System Modelling and Optimization*, 451-458.
- [Cas96] CASTRO, J. e NABONA, N. (1996). An Implementation of Linear and Nonlinear Multicommodity Network Flows, *European Journal of Operational Research*, **92**, 37-53.
- [Cha02] CHARDAIRE, R. e LISSER, A. (2002). Simplex and Interior Point Specialized Algorithms for Solving Nonoriented Multicommodity Flow Problems, *Operations Research*, **50**, 260-276.
- [Cha61] CHARNES, A. e COOPER, W.W. (1961). *Management Models and Industrial Applications of Linear Programming*, John Wiley and Sons, New York.
- [Che74] CHEN, H. e DEWALD, C. G. (1974). A Generalized Chain Labelling Algorithm for Solving Multicommodity Flow Problems, *Computers and Operations Research*, **1**, 437-465.
- [Che92] CHEN, Y. e CHIN, Y. (1992). Multicommodity network flows with safety considerations. *Operations Research*, **40**, 48-55.
- [Chr75] CHRISTOFIDES, N. (1975). *Graph Theory: An Algorithmic Approach*, Academic Press, London.
- [Coe83] COELHO, A. (1983). Uma Opção de Análise de Pós Otimalidade para o Algoritmo Project, Dissertação de Mestrado, UFSC.
- [Con04] CONDERT, D.; RIVANO, H. e ROCHE, X. (2004). A Combinatorial Approximation Algorithm for the Multicommodity Flow Problem, *Lecture Notes in Computer Science* **2909**, 256-259.
- [Dan63] DANTZIG, G. B. (1963). *Linear Programming and Extensions*, Princeton University Press, New Jersey.
- [Det02] DETLEFSEN, N.K. e WALLACE, S.W. (2002). The simplex algorithm for multicommodity networks, *Networks*, **39**, 15-28.
- [Far93] FARVOLDEN, J.M.; POWELL, W.B. e LUSTIG, I.J. (1993). A Primal Partitioning Solution for Multicommodity Network Flow Problems, *Operations Research*, **41**, 669-693.
- [For58] FORD, L.R. e FULKERSON, D.R. (1958). A Suggested Computation for Maximal Multi-Commodity Network Flow, *Management Science*, **5**, 822- 844.

- [For92] FORREST, J.J. e GOLDFARB, D. (1992). Steepest-Edge Simplex Algorithm for Linear Programming, *Mathematical Programming*, **3**, 157-177.
- [For72] FORREST, J.J. e TONLIN, J.A. (1972). Update Triangular Factors of the Basis to Maintain Sparsity in the Product Form Simplex Method, *Math Programming*, **2**, 263-278.
- [Fra99] FRANGIONI, A. e GALLO, G. (1999). A bundle Type Dual-Ascent Approach to Linear Multicommodity Min-Cost Flow Problems, *Inform Journal on Computing*, **11**, 370-393.
- [Fre96] FREUND, R.M. e MIZUNO, S. (1996). *Interior Point Methods: Current Status and Future Direction*.
- [Ful71] FULKERSON, D.R. (1971). An Out-of-Kilter method for Minimal Cost Flow Problems, *Journal of the Society of Industrial and Applied Mathematics*, **9**, 18-27.
- [Gab03] GABREL, V.; KNIPPEL, A. e MINOUX, M. (2003). A comparison of Heuristics for the Discrete Cost Multicommodity Network Optimization Problems, *Journal of Heuristics*, **9**, 429-445.
- [Gab99] GABREL, V.; KNIPPEL, A. e MINOUX, M. (1999). Exact Solution of Multicommodity Network Optimization Problems with General Step Cost Functions, *Operations Research Letters*, **25**, 15-23.
- [Geo74] GEOFFRION, A.M. e GRAVES, G.M. (1974). Multicommodity Distribution System Design By Benders Decomposition. *Management Science*, **20**, 822-844.
- [Glo74] GLOVER, F.; KARNEY, D. e KLINGMAN, D. (1974). A Computational Study on Start Procedures, Basis Change Criteria, and Solution Algorithms for Transportation Problems, *Management Science*, **20**, 793-813.
- [Gol00] GOLDBARG, M.C. e LUNA, H.P. (2000) *Otimização Combinatória e Programação Linear*, Campus, Rio de Janeiro.
- [Gon93] GONDZIO, J. (1993). Implementing Choleski Factorization for Interior Point Methods of Linear Programming, *Optimization and Applications*, **27**, 121-140.
- [Gon97] GONDZIO, J.; SARKISSIAN, R. e VIAL, P. (1997). Using an Interior Point Method for the Master Problem in a Decomposition Approach, *European Journal of Operation Research*, **101**, 577-587.
- [Gon92] GONZAGA, C.C. (1992). Path Following Methods for Linear Programming. *SIAM, Review* **34**, **2**, 167-227.
- [Gra76] GRAVES, G.W. e McBRIDE, R.D. (1976). The Factorization Approach to Large-Scale Linear Programming, *Mathematical Programming*, **10**, 91-110.
- [Gri72] GRIGORIADIS, M.D. e WHITE, W.W. (1972). A Partitioning Algorithm for the Multicommodity Network Flows Problem, *Mathematical Programming*, **3**, 157-177.
- [Gri86] GRIGORIADIS, M.D. (1986). An Efficient Implementation of the Network Simplex Method, *Mathematical Programming Study*, **26**, 83-111.

- [Had00] HADJIAT, M.; MAURRAS, J.F. e VAXES, Y. A Primal Partitioning approach for single and non-simultaneous Multicommodity Flow Problems, *European Journal of Operational Research*, **123**, 382-393.
- [Har72] HARTMAN, J.K. e LASDON, L.S. (1972). A Generalized Upper Bounding Algorithm for Multicommodity Network Flow Problems, *Networks*, **1**, 333-354.
- [Hel77] HELGASON, R.V. e KENNINGTON, J.L. (1977). A Product Form Representation of the Inverse of a Multicommodity Cycle Matrix, *Networks*, **7**, 297-322.
- [Hil91] HILLIER, F.S. e LIEBERMAN, G.J. (1991). *Introducción a la Investigación de Operaciones*, Mc Graw Hill, México.
- [Hof61] HOFFMAN, K. e KUNZE, R. (1961). *Linear Álgebra*, Prentice-Hall.
- [Hol03] HOLMBERG, K. e YUAN, D. (2003). A Multicommodity Network Flow Problem with Side Constraints on Paths Solved by Column Generation, *Inform Journal on Computing*, **15**, 42-57.
- [Joh65] JOHNSON, E.L. (1965). Programming in Networks and Graphs, *Technical Report ORC 65-1*, Operational Research Center.
- [Ken75] KENNINGTON, J. (1975). Solving Multicommodity Transportation Problems Using a Primal Partitioning Simplex Technique, *Technical Report 75013*, Department of Computer Science and Operations Research.
- [Ken77] KENNINGTON, J.L. (1977). Solving Multicommodity Transportation Problems Using Primal Partitioning Simplex Technique, *Naval Research Logistics Quarterly*, **24**, 309-325.
- [Ken78] KENNINGTON, J.L. (1978). A Survey of Linear Cost Multicommodity Network Flows, *Operational Research*, **26**, 209-236.
- [Ken80] KENNINGTON, J.L. e HELGASON, R.V. (1980). *Algorithm for Network Programming*. New York:, John Willey & Sons.
- [Ken77] KENNINGTON, J.L. e SHALABY, M. (1977). An Effective Subgradiante Procedure for Minimal Cost Multicommodity Flow Problems, *Management Science*, **9**, 994-1004.
- [Kol93] KOLMAN, B. (1993). *Introductory Linear Algebra with Applications*, 5.ed., New York: McMillan.
- [Lar93] LARI, A.R. e NAG, B. (1993). Model Management Solution System for Commodity Network Flows, *European Journal of Operational Research*, **71**, 398-418.
- [Lar04] LARSSON, T. e DI, YA. (2004). An Augmented Lagrangian Algorithm for Large Scale Multicommodity Routing, *Computational Optimization and Applications*, **27**, 187-215.
- [Lar97] LARSSON, T. e LIU, Z. (1997). A Lagrangean relaxation scheme for structured linear programs with application to multicommodity network flows, *Operational Research*, **3**, 247-284.

- [Len91] LENSTRA, J.K.; KAN, A.R. e SCHIRIVER, A. (1991). *History of Mathematical Programming*. North – Holland.
- [Lin97] LIN, S. e LIN C. (1997). A computationally efficient method for non linear multicommodity network flow problems. *Networks*, **29**, 225-244.
- [Liu97] LIU, C.M. (1997). Network Dual Steepest-Edge Methods for Solving Capacitated Multicommodity Network Problems, *Computers Industrial Engineering*, **33**, 697-700.
- [Lue84] LUENBERGER, D.G. (1984). *Linear and Non Linear Programming*, 2.ed., Addison-Wesley.
- [Mam00] MAMER, J.W. e McBRIDE, R.D. (2000). A decomposition-based pricing procedure for large-scale linear programs: An application to the linear multicommodity flow problem. *Management Science*, **46**, 693-709.
- [Mar57] MARKOVITZ, H.M. (1957). The Elimination Form of Inverse and its Applications to Linear Programming, *Management Science*, **3**, 255-269.
- [May84] MAYERLE, S.F. (1984). Programação Linear com Trocas Múltiplas de Base (Implementação de um Sistema Computacional usando FPI), Dissertação de Mestrado, UFSC.
- [Mcb98] McBRIDE, R.D. (1998) *Advances in solving the multicommodity flow problem*. Interfaces, 28, 32-41.
- [Mes99] MESQUITA, M. e PAIXÃO, J. (1999). Exact algorithms for the multi-depot vehicle scheduling problem based on multicommodity network flow type formulations. *Lecture Notes in Economics and Mathematical System*, **471**, 221-243.
- [Mcb01] McBRIDE, R.D. e MAMER, J.W. (2001). Solving the Undirected Multicommodity Flow Problem using a Shortest Path-Based Pricing Algorithm, *Networks*, **38**, 181-188.
- [Mcb98] McBRIDE, R.D. (1998). Progress made in Solving the Multicommodity Flow Problem, *Siam Journal on Optimization*, **8**, 947-955.
- [Moz04] MOZ, M. e PATO, M.V. (2004). Solving the Problem of Rostering Nurse Schedules with hard Constraints: New Multicommodity Flow Modelo. *Annals of Operation Research*, **128**, 179-197.
- [Mur78] MURTAGH, B.A. e SAUNDERS, M.A. (1972). Large-Scale Linearly Constrained Optimization. *Mathematical Programming*, **14**, 41-72.
- [Nan69] NANTWADA, M. (1969). Multicommodity Flows in a Communication Network, *Eletronics and Communications in Japan*, **52**, 34-41.
- [Orc68] ORCHARD-HAYS, W. (1968). *Advanced Linear-Programming Computing Techniques*, McGraw-Hill, New York.
- [Ouo00] OUOROU, A.; MAHEY, P. e VIAL, J. (2000). Survey of Algorithms for Convex Multicommodity Flow Problems, *Management Science*, **46**, 126-147.
- [Par02] PARDALOS, P.M. e RESENDE, M.G.C. (2002). *Handbook of Applied Optimization*, Oxford University Press, New York.

- [Pot72] POTTS, R.B. e OLIVER, R.M. (1972). *Flow in Transportation Networks*, Academic Press, New York.
- [Röd80] RÖDDER, W. e BLAUTH, M. (1980). Project – An alternative LP-code. *Boletim de Produção e Sistemas – UFSC*, **2**, 24 -35.
- [Röd82a] RÖDDER, W. (1982). Finiteness of Project under Perturbation, Artigo de circulação interna do departamento de Engenharia de Produção e Sistemas, UFSC, Florianópolis.
- [Röd82b] RÖDDER, W.; COELHO, S. e MAYERLE, S. (1982). A Forma Mista da Inversa para Troca Múltiplas de Base em um LP-Algoritmo, Artigo apresentado no XV SBPO/ I CLAPO, Rio de Janeiro, 621-638.
- [Ros93] ROSS, K. e LYNN (1993). The Impact of Formulation Decomposition, *Operations Research*, **54**.
- [Sai66] SAIGAL, R. (1966). Multicommodity Flows in Directed Networks, *Operation Research Center Report*, ORC Report 66-24.
- [Sch98] SCHNEUR, R.R. e ORLIN, J.B. (1998). A scaling algorithm for multicommodity flow problems. *Operation Research*, **46**, 231-246.
- [Sou00] SOUZA, R.S. (2000). Estudos em Otimização Linear. Dissertação apresentada ao Instituto de Ciências Matemática e de Computação – ICMS - USP, São Carlos.
- [Vis04] VISWANATH, K. e PEETA, S. (2004). Multicommodity Maximal Covering Network Design Problem for Planning Critical Routes for Earthquake Response, *Transportation Research Record*, 1-10.
- [Wei72] WEIGEL, H.S. e CREMEANS, J.E. (1972). The Multicommodity Network Flow Model Revised to Include Vehicle Per Time Period and Node Constraints, *Naval Research Logistics*, **19**, 77-89.
- [Whi72] WHITE, W.W. (1972). Mathematical Programming, Multicommodity Flows, and Communication Nets, *Proceedings of the Symposion on Computer-Communications Networks and Teletraffic*, 325-334.
- [Wol72] WOLLMER, R.D. (1972). Multicommodity with Resource Constraint: The Generalized Multicommodity Flow Problem, *Networks*, **1**, 245-263.
- [Yam96] YAMAKAWA, E.; MATSUBARA, Y. e FUKUSHIMA, M. (1996). A parallel primal-dual interior point method for multicommodity flow problems quadratic cost, *Journal of the Operations Research Society of Japan*, **39**, 566-591.

GLOSSÁRIO

G	grafo conexo;
\hat{G}	subgrafo de G ;
G^k	grafo associado ao produto k ;
\mathcal{N}	conjunto de nós para o grafo G ;
$\hat{\mathcal{N}}$	subconjunto de nós;
\mathcal{R}	conjunto de arcos para o grafo G ;
$\hat{\mathcal{R}}$	subconjunto de arcos;
$F(j)$	nó de origem do arco j ;
$T(j)$	nó de destino do arco j ;
τ	árvore geradora para G ;
τ^k	árvore geradora para G^k ;
\mathbf{l}	nó raiz para o grafo G ou G^k ;
z	função objetivo;
A	matriz de incidência nó-arco;
A_{ij}	elemento da matriz de incidência A ;
A_j	j -ésima coluna da matriz de incidência A ;
A_{j_i}	j -ésima coluna da matriz de incidência A correspondente a e_{j_i} ;
$A(j)$	j -ésima coluna da matriz de incidência A associada ao arco e_j ;
$A(k)$	k -ésima coluna da matriz de incidência A associada ao arco e_k ;
\hat{A}	matriz de restrições para o problema MP ;
\bar{A}^k	matriz de incidência nó-arco associada a um grafo G^k com nó raiz \mathbf{l} .
A^k	matriz de incidência nó-arco associada ao k -ésimo produto;
\bar{B}	matriz básica particionada para a matriz \hat{A} ;

- B matriz de incidência nó-arco que forma uma base para a matriz A ;
 B^k matriz de incidência nó-arco que forma uma base para \bar{A}^k associada ao produto k ;
 P seqüência finita representando um caminho no grafo G ;
 C seqüência finita representando um ciclo no grafo G ;
 O_iP orientação do arco j na seqüência do caminho P no grafo;
 O_iC orientação do arco j na seqüência do ciclo C no grafo;
 s_i índice do nó na seqüência do caminho ou ciclo no grafo G ;
 e_j arco correspondente a j -ésima coluna da matriz A ;
 e_{j_i} índice i do arco j arco na seqüência do caminho ou ciclo no grafo G ;
 e_k k -ésimo arco candidato a entrar na base;
 e_{j_s} arco correspondente a s -ésima coluna de R^k ;
 s_i variáveis de folga;
 \mathbf{e}^{s_i} vetor coluna canônico com entrada 1 na i -ésima linha;
 \mathbf{e}^l vetor coluna canônico com entrada 1 na l -ésima linha (correspondente ao nó raiz);
 \mathbf{e}_r vetor linha canônico com entrada não nula correspondente a r -ésima linha para o arco complementar ou folga inviável;
 \mathbf{e}_{R_k} matriz cujas linhas representam o vetor canônico com entrada não nula correspondente às restrições ativas de agregação associadas aos arcos complementares da partição R^k ;
 N matriz associadas as variáveis não básicas
 $N(i)$ i -ésima coluna da matriz N correspondente à variável não básica candidata a entrar na base
 x vetor de fluxos
 x^B vetor de fluxos associados as variáveis básicas
 x^N vetor de fluxos associados as variáveis não básicas
 x^k vetor de fluxos para o k -ésimo produto
 r vetor de ofertas e demandas
 r^k vetor de ofertas e demandas associados ao produto k ;
 b vetor das capacidades associadas às restrições de agregação de fluxo
 c vetor de custos unitários
 c_{j_i} índice i do custo unitário do arco j na seqüência do caminho ou ciclo no grafo G
 c^B vetor de custos unitários associados as variáveis básicas
 c^N vetor de custos unitários associados as variáveis não básicas
 l capacidade mínima nos arcos

- l^k capacidade mínima de fluxo nos arcos associadas ao produto k ;
 u capacidade máxima nos arcos
 u^k capacidade máxima de fluxo nos arcos associadas ao produto k ;
 D^k matriz de coeficientes das restrições de agregação de fluxo para o k -ésimo produto
 R^k matriz de incidência nó-arco associada aos arcos superbásicos definidas para o produto k ;
 P^k matriz de coeficientes associados aos arcos básicos do produto k , nas restrições ativas de agregação de fluxo;
 S^k matriz de coeficientes associados aos arcos básicos do produto k , nas restrições não ativas de agregação de fluxo;
 T^k matriz de coeficientes associados aos arcos superbásicos (complementares) do produto k , nas restrições ativas de agregação de fluxo;
 U^k matriz de coeficientes associados aos arcos superbásicos (complementares) do produto k , nas restrições não ativas de agregação de fluxo;
 I matriz identidade associada às folgas nas restrições não ativas de agregação;
 Q matriz ciclo Q ;
 Y matriz ciclo Y ;
 π^{1k} valores duais associados aos nós para a árvore τ^k do k -ésimo produto;
 π^{2k} valores duais para as folgas associadas aos arcos complementares do k -ésimo produto nas restrições ativas de agregação;
 π^3 valores duais associados as folgas básicas pertencentes às restrições não ativas de agregação;
 $\bar{\mathbf{a}}_j$ coluna particionada correspondente a um arco ou uma folga não básica;
 y^{1k} valores de y 's do teste de razão para todos os arcos básicos do k -ésimo produto;
 y^{2k} valores de y 's do teste de razão para todos os arcos complementares do k -ésimo produto;
 y^3 valores de y 's do teste de razão para todas as variáveis de folga básicas;
 \bar{y}^{1k} valor de y no teste de razão para o arco básico do k -ésimo produto;
 \bar{y}^{2k} valor de y no teste de razão para o arco complementar do k -ésimo produto;
 \bar{y}^3 valor de y no teste de razão para a variável de folga básica.

APÊNDICES

- I. REDE COM SUPER NÓ DE TRANSBORDO PARA ILUSTRAR PASSO A PASSO O ALGORITMO PARTICIONADO PRIMAL QUE UTILIZA A MATRIZ CICLO Y .

- II. REDE COM DISTRIBUIÇÃO DE PRODUTOS COMPOSTOS PARA ILUSTRAR PASSO A PASSO TRÊS ITERAÇÕES DO ALGORITMO PARTICIONADO PRIMAL QUE UTILIZA A MATRIZ CICLO Y .

I. REDE COM SUPER NÓ DE TRANSBORDO PARA ILUSTRAR PASSO A PASSO O ALGORITMO PARTICIONADO PRIMAL QUE UTILIZA A MATRIZ CICLO Y.

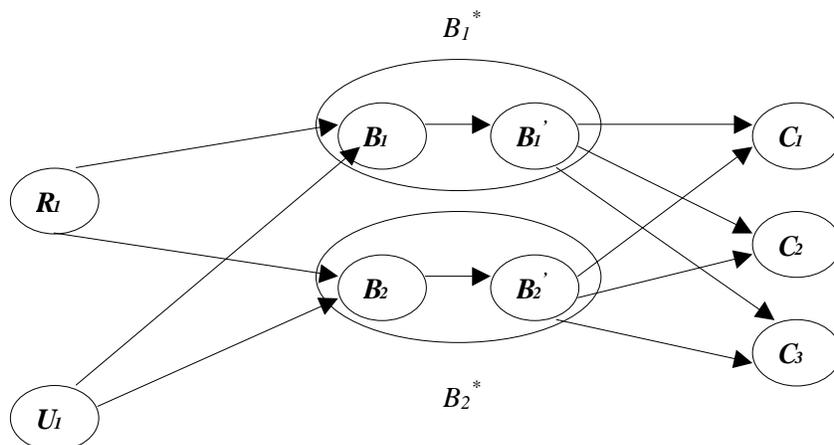


Figura I-1 - Transporte de dois produtos em um modelo de rede contendo dois super nós.

O exemplo abaixo, considera o transporte de dois produtos. Os nós de oferta para o produto 1 e 2 são respectivamente os nós R_1 e U_1 , os nós de demanda para ambos os produtos são os nós C_1 , C_2 e C_3 . A figura I-1 mostra a representação lógica dos super nós B_1^* e B_2^* (nós de transbordo) onde circulam os dois produtos sujeitos a uma restrição de capacidade de 50 e 40 unidades respectivamente.

As tabelas I-1 e I-2 representam as quantidades em unidades de oferta e demanda para cada produto e as tabelas I-3 e I-4 representam a especificação dos arcos para ambos os produtos.

Tabela I-1 - Oferta e demanda para o Produto 1

Código do Nó	PRODUTO 1	
	Oferta	Demanda
R_1	60	
B_1 , B_1'		
B_2 , B_2'		
C_1		20
C_2		20
C_3		20

Tabela I-2 - Oferta e demanda para o Produto 2

Código do Nó	PRODUTO 2	
	Oferta	Demanda
U_1	15	
B_1 , B_1'		
B_2 , B_2'		
C_1		5
C_2		10
C_3		0

Tabela I-3 - Especificação dos Arcos para o Produto 1

CÓDIGO DO ARCO	NÓ INICIAL	NÓ FINAL	CUSTO UNITÁRIO DO ARCO	FLUXO INICIAL NOS ARCOS	CAPACIDADE NOS ARCOS
x_1	R_1				
x_{11}	R_1	B_1	5.0	50	∞
x_{12}	R_1	B_2	6.0	10	∞
x_{13}	B_1	B_1	0.0	50	∞
x_{14}	B_2	B_2	0.0	10	∞
x_{15}	B_1	C_1	10.0	10	∞
x_{16}	B_1	C_2	8.0	20	∞
x_{17}	B_1	C_3	9.0	20	∞
x_{18}	B_2	C_1	4.0	10	∞
x_{19}	B_2	C_2	6.0	0	∞
x_{110}	B_2	C_3	7.0	0	∞

Tabela I-4 - Especificação dos Arcos para o Produto 2

CÓDIGO DO ARCO	NÓ INICIAL	NÓ FINAL	CUSTO UNITÁRIO DO ARCO	FLUXO NOS ARCOS	CAPACIDADE NOS ARCOS
x_2	U_1				
x_{21}	U_1	B_1	3.0	0	∞
x_{22}	U_1	B_2	4.0	15	∞
x_{23}	B_1	B_1	0.0	0	∞
x_{24}	B_2	B_2	0.0	15	∞
x_{25}	B_1	C_1	7.0	0	∞
x_{26}	B_1	C_2	8.0	0	∞
x_{27}	B_1	C_3	9.0	0	∞
x_{28}	B_2	C_1	6.0	5	∞
x_{29}	B_2	C_2	8.0	10	∞
x_{210}	B_2	C_3	10.0	0	∞

Pretende-se, através desse exemplo, fazer uma interpretação física do problema, procurando interpretar as expressões que solucionam o problema por meio do método simplex primal particionado. Todos os resultados algébricos deduzidos no capítulo três são interpretados de uma forma mais intuitiva. A idéia é mostrar que o algoritmo simplex com particionamento primal pode ser realizado diretamente no grafo eliminando assim, a necessidade de operações matriciais.

Temos um problema de fluxo a custo mínimo, onde corresponde à necessidade de circular fluxos de dois produtos sujeitos a restrições de rede e restrições de capacidade agregada nos super-nós (máximo e mínimo de fluxo admissível nos arcos, bem como um valor associado ao trânsito em cada arco por unidade de fluxo (c_{ij})).

Parte-se de uma solução básica viável, onde existe fluxo do produto 1 que circula através de um arco superbásico associado a uma restrição em que a soma dos fluxos dos dois

produtos através do arco B_1B_1' alcança a capacidade máxima igual a 50 unidades, ou seja, o arco B_1B_1' se encontra saturado com folga nula.

Computar as variáveis duais nos problemas de múltiplos produtos sem restrições de capacidade agregada significa obter a cada iteração uma base sem arcos superbásicos para os k produtos. Resolver o sistema $\pi \bar{B} = C_B$, é determinar o potencial π_i do nó que fornece o equilíbrio ou estado estável com respeito ao fluxo sustentado pela árvore básica. A equação $\pi_i - \pi_j = c_{ij}$ significa estar em equilíbrio quando se produz uma unidade a mais no nó i , transporta essa unidade ao longo do arco (i,j) e a vende no nó j .

Portanto, no exemplo apresentado sem o arco superbásico, as variáveis duais podem ser computadas diretamente nas árvores, como mostrado nas figuras abaixo.

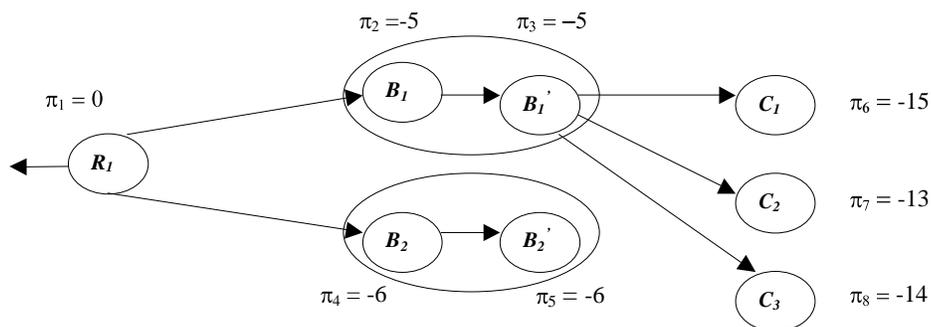


Figura I-2 - Valor dual dos nós para o produto 1 sem considerar as restrições de capacidade agregada

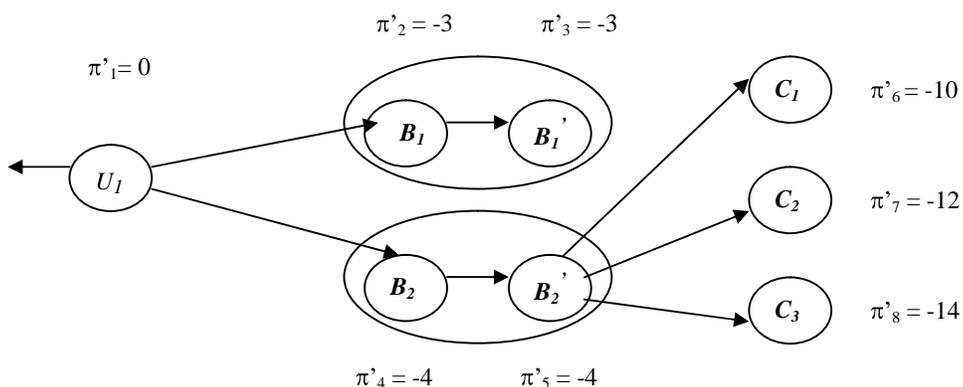


Figura I-3 - Valor dual dos nós para o produto 2 sem considerar as restrições de capacidade agregada.

Entretanto a existência de um arco complementar (ou superbásico) associado a uma restrição de capacidade agregada ativa com folga nula, significa a existência de um fluxo sustentado pela árvore básica e também pelo arco complementar de um determinado produto k . Então, é natural pensar que as equações $(\pi_i - \pi_j = c_{ij})$ e $(z_{ij} - c_{ij} = 0)$ sejam também verdadeiras para os arco complementares, já que o fluxo é sustentado pela árvore básica e também pelos arcos complementares.

Logo, o arco x_{18} quando unido a sua árvore básica deverá ter $(\pi_i - \pi_j = c_{ij})$ e $\sum_{i=1}^n c_i O_i(C) A(j_i) = 0$ (a soma dos custos no ciclo formado com o arco x_{18} seja nula). O potencial π^2 é então obtido para restabelecer esse equilíbrio. Com a inclusão de arcos complementares obtém-se uma nova base que contém ciclos possuindo arcos saturados pertencentes a restrições ativas de agregação. Pode ser observado que na cadeia formada pelo arco complementar x_{18} , existe o arco x_{13} que se encontra saturado na restrição ativa de agregação com folga S_1 nula. O valor dual π^2 é o valor dual associado à folga nula correspondente à restrição ativa de agregação o qual está associado a um arco complementar. Portanto o preço π^2 pode ser interpretado como um preço de ajuste para fornecer o equilíbrio ou estado estável com respeito ao fluxo sustentado pela árvore básica e pelo arco complementar, observando que esse preço deve ser repassado aos arcos do caminho na árvore básica para os k -ésimos produtos que pertencem à restrição ativa de agregação.

Assim,

$$\pi^2 Q = c^{12} - c^{11} (B^1)^{-1} R^1, \text{ onde}$$

No ciclo formado com o arco complementar, tem-se

$$Q = 0x_{18} - (0x_{14} + 0x_{12} + 0x_{11} + 1(O_i P_j) + 0x_{15}) = 0 - 1(+1) = -1, \text{ para a restrição } x_{13} + x_{23} + S_1 = 50$$

$$\sum_{i=1}^n c_i O_i(C) A(j_i) = 4 - (0 - 6 + 5 + 0 + 10) = -5$$

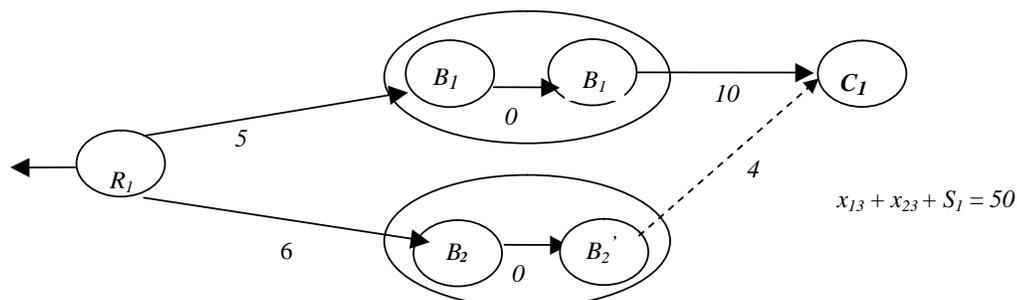


Figura I-4 - Valor dos custos nos Arcos da árvore básica para o produto 1 no ciclo formado com o arco complementar.

$$\sum_{i=1}^n c_i O_i(C) A(j_i) = 4 - (0 - 6 + 5 + 0 + 10) = -5$$

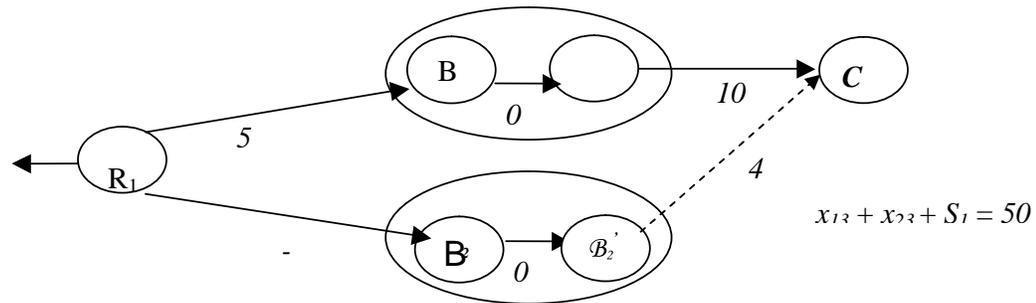


Figura I-4 - Valor dos custos nos Arcos da árvore básica para o produto 1 no ciclo formado com o arco complementar.

Então

$$\pi^2 = [c^{12} - c^{11} (B^1)^{-1} R^1] Q^{-1} = [4 - [5 - 6 + 10]] (-1) = 5.$$

O valor dual obtido π^2 deve ser repassado a toda rede, ou seja, esse valor é adicionado ao valor dual do nó B_1' e a todos os nós a partir deste nas árvores básicas para ambos os produtos. Essa adição deve ser feita no valor dual dos nós que foram computados diretamente nas árvores básicas como se não houvesse arcos complementares. O valor dual acrescido é mostrado nas figuras I-5 e I-6.

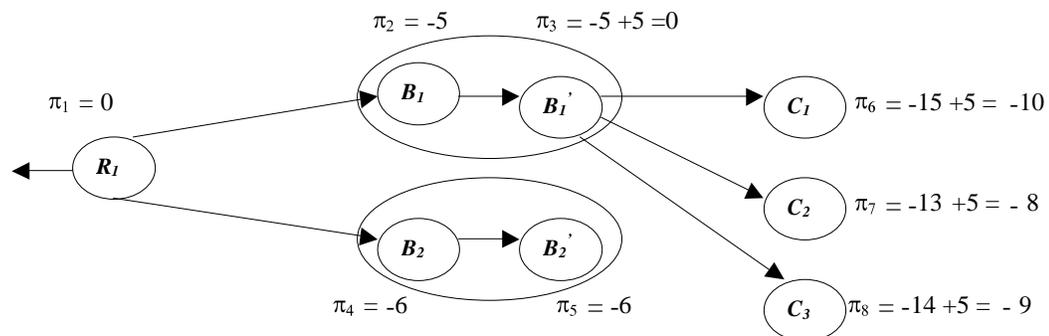


Figura I-5 - Correção do valor dual dos nós para o produto 1 considerando a restrição de capacidade agregada associada ao arco complementar.

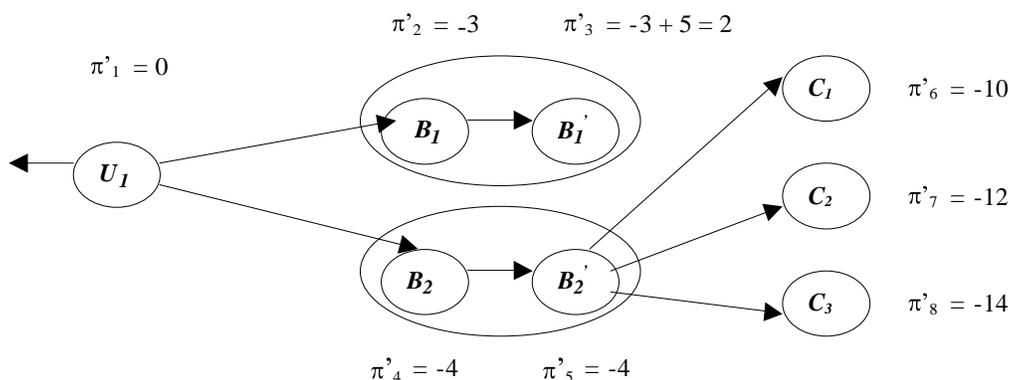


Figura I-6 - Correção do valor dual dos nós para o produto 2 considerando a restrição de capacidade agregada associada ao arco complementar.

Equivalentemente, de uma outra maneira o ajuste pode ser feito nos arcos do caminho P_j (arcos que pertencem à restrição de agregação e que estão na árvore básica para os k produtos) de forma que estes passam a ter seus custos corrigidos em função da adição de arcos complementares. A partir desta correção são computadas as variáveis duais diretamente na árvore básica. A diferença $(c^{1k} - \pi^2 P^k)$ que aparece na equação (3.28) que determina a variável dual π^{1k} é a correção nos custos dos arcos que se encontram na restrição ativa de agregação no caminho P_j da árvore básica.

No exemplo, a correção é feita no custo do arco $B_1 B_1'$ (arco de agregação para ambos os produtos), o novo custo é então -5 para ambos os produtos.

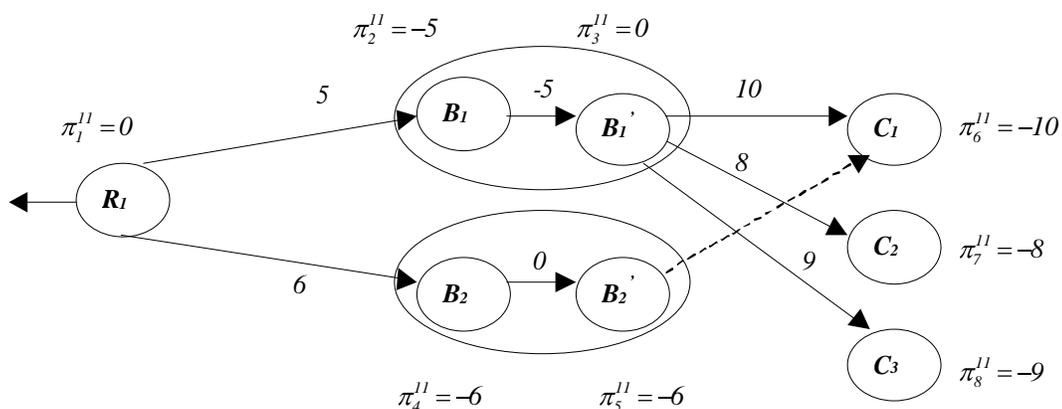


Figura I-7 - Correção no custo dos arcos para o produto 1 considerando a restrição de capacidade agregada associada ao arco complementar

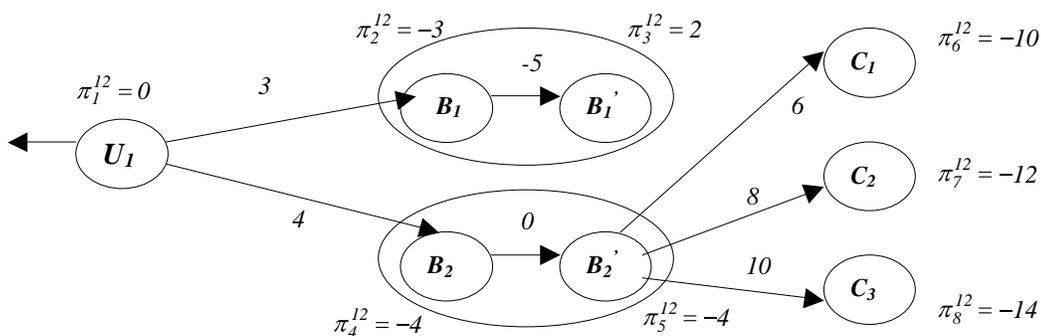


Figura I-8 - Correção no custo dos arcos para o produto 2 considerando a restrição de capacidade agregada associada ao arco complementar

Tabela I-5 - Representação da matriz de incidência A_k e matriz dos coeficientes D_k nas restrições de agregação de fluxos.

	x_1	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_2	x_{21}	x_{22}	x_{23}	x_{24}	x_{28}	x_{29}	x_{210}	x_{18}	S_2
R_1	1	1	1															
B_1		-1		1														
B_1'				-1		1	1	1										
B_2			-1		1													
B_2'					-1												1	
C_1						-1											-1	
C_2							-1											
C_3								-1										
U_1									1	1	1							
B_1										-1								
B_1'												-1						
B_2											-1		1	1	1	1		
B_2'													-1	1	1	1		
C_1														-1				
C_2															-1			
C_3																-1		
				1								1						
					1								1					
																		1

Agora, as equações $(\pi_i - \pi_j = c_{ij})$ e $(\sum_{i=1}^n c_i O_i(C) A(j_i)) = 0$ são verdadeiras para todos

os arcos que compõem a nova base, o arco não básico candidato a entrar na base pode ser então escolhido.

O quadro abaixo representa as árvores básicas mostradas nas figuras I-7 e I-8 obtidas a partir da solução inicial viável para a primeira iteração. As árvores básicas são representadas

na forma matricial, identificando as restrições de conservação de fluxo e capacidades agregadas.

As duas últimas linhas do quadro anterior correspondem à restrição ativa de agregação ($x_{13} + x_{23} + S_1 = 50$) e não ativa de agregação ($x_{14} + x_{24} + S_2 = 40$) para o problema proposto - cujas matrizes correspondentes são identificadas como:

$$T^1 = [0], \quad P^1 = P^1 = [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0],$$

$$S^1 = S^2 = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0], \quad U^1 = [0], \quad \text{com } Q = [1], \quad \text{tal que } \dim Q = 1.$$

Iteração 1)

Nessa primeira iteração são mostrados a determinação das variáveis duais e os fluxos atualizados a partir de Q^{-1} e Y^1 . A diferença entre a manipulação de uma ou outra matriz consiste basicamente em como armazenar as linhas ativas e colunas associadas a arcos complementares de forma que a cada pivoteamento a matriz ciclo tenha dimensão fixa dada pelo número n de restrições de agregação.

As variáveis duais determinadas anteriormente para cada nó de ambos os produtos são

$$\pi^{11} = \begin{bmatrix} R_1 & B_1 & B_1' & B_2 & B_2' & C_1 & C_2 & C_3 \\ 0 & -5 & 0 & -6 & -6 & -10 & -8 & -9 \end{bmatrix} \text{ e}$$

$$\pi^{12} = \begin{bmatrix} U_1 & B_1 & B_1' & B_2 & B_2' & C_1 & C_2 & C_3 \\ 0 & -3 & 2 & -4 & -4 & -10 & -12 & -14 \end{bmatrix}.$$

Através do cálculo das variáveis duais em ambas as redes, os candidatos a entrar na base são:

Para o produto 1, somente a folga S_1 associada ao arco complementar x_{18} é candidata a entrar na base pois $S_1 = 0$ e possui $\pi^2 = 5 > 0$. Para o produto 2, os candidatos são os arcos x_{25} , x_{26} e x_{27} .

A folga S_1 (Produto 1) é a variável escolhida para entrar na base, $\delta \leftarrow 1$.

Teste de razão

Considerando-se a existência da união de uma árvore básica a um arco complementar para o produto 1, o envio de uma quantidade Δ de fluxo através de S_1 fará por ajustar as variáveis básicas e o arco complementar de modo a manter a viabilidade, determinando assim a variável candidata a sair da base.

Adicionar uma quantidade Δ de fluxo a variável de folga S_1 reduzirá uma quantidade Δ de fluxo no arco x_{13} de modo a manter a viabilidade com respeito à capacidade máxima de

fluxo agregado no arco B_1B_1' , ou seja, a passagem de fluxo para variável de folga S_1 faz com que o arco x_{13} (que se encontrava saturado) seja reduzido por uma quantidade Δ de fluxo. Como o processo de ajuste do envio de fluxo se dá ao redor da cadeia formada pelo arco complementar, todos os arcos dessa cadeia serão ajustados de modo a manter o equilíbrio. A folga S_2 que se encontra na base pertencente a uma restrição de capacidade agregada não ativa que une os arcos x_{14} e x_{24} deverá ser ajustada em função do arco x_{14} que é modificado por pertencer à cadeia formada pelo arco complementar.

Para o exemplo apresentado, o aumento da quantidade Δ na cadeia formada com o arco complementar faz com que o arco x_{15} atinja o limite inferior e, portanto, seja candidato a sair da base. A rede para o produto 2 permanece a mesma, e isso pode ser observado através do teste de razão abaixo.

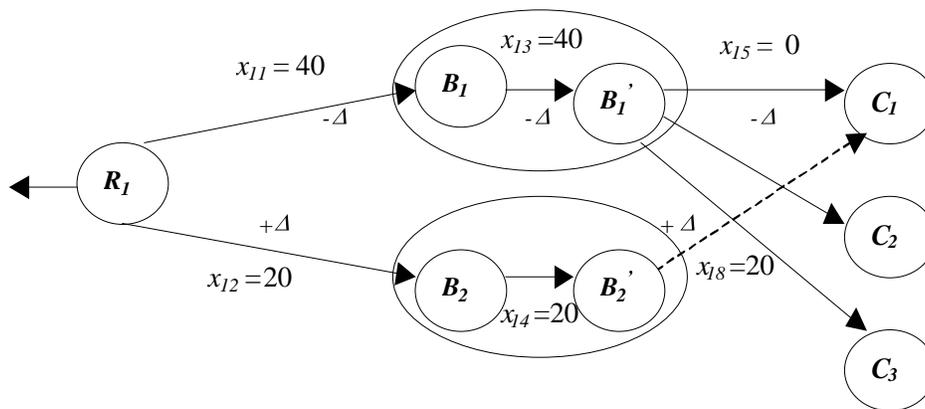


Figura I-9 - Atualização dos Fluxos nos arcos do produto 1 na iteração 1.

Como entra a folga S_1 , tem-se $(B^1)^{-1}\alpha^1 = 0$, portanto

$$y^2 = y^{2l} = Q^{-1}[\beta^1 - P^1(B^1)^{-1}\alpha^1] = (-I)[1 - 0] = \begin{bmatrix} x_{18} \\ -1 \end{bmatrix},$$

$$(y^{1l})^T = (B^1)^{-1}\alpha^1 - (B^1)^{-1}R^1y^{2l} = \begin{bmatrix} x_1 & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & x_{17} \\ 0 & 1 & -1 & 1 & -1 & 1 & 0 & 0 \end{bmatrix},$$

$$y^3 = \gamma - S^1 y^{1l} = \begin{bmatrix} S_2 \\ 1 \end{bmatrix},$$

$$\Delta_1 = \min\{x_{11}, x_{13}, x_{15}, S_2\} = \min\{50, 50, 10, 15\} = 10,$$

$$\Delta_2 = \infty,$$

$$\Delta = 10.$$

Atualização dos Fluxos

$$\begin{aligned}
S_1 &\leftarrow 10 \\
x_{11} &= 50 - I(1)10 = 40 \\
x_{12} &= 10 - I(-1)10 = 20 \\
x_{13} &= 50 - I(1)10 = 40 \\
x_{14} &= 10 - I(-1)10 = 20 \\
x_{15} &= 10 - I(1)10 = 0 \\
x_{18} &= 10 - I(-1)10 = 20 \\
S_2 &= 15 - I(1)10 = 5.
\end{aligned}$$

Observa-se que o fluxo no produto 2 é mantido, pois $y^{22}=0$.

Tendo em vista obter uma matriz ciclo de tamanho não variável é que o cálculo das variáveis duais π^2 e π^3 é determinado juntamente quando multiplicadas pela matriz inversa ciclo, assim como a obtenção dos valores de y^2 e y^3 no teste de razão.

A matriz ciclo Y obtida de acordo com (3.23) é $Y = \left[\begin{array}{c|c} -I & 0 \\ \hline I & I \end{array} \right]$, tendo como matriz inversa ciclo a matriz $Y^{-1} = \left[\begin{array}{c|c} -I & 0 \\ \hline I & I \end{array} \right]$. A primeira linha da matriz ciclo se encontra ativa assim como a primeira coluna está associada ao arco complementar x_{18} .

No cálculo das variáveis duais π^2 e π^3 tem-se:

$$\left[\pi^2 \mid \pi^3 \right] = \left[c^{21} - c^{11}(B^1)^{-1}R^1 \mid 0 \right] Y^{-1} = \left[4 - [5 - 6 + 10] \mid 0 \right] \left[\begin{array}{c|c} -I & 0 \\ \hline I & I \end{array} \right] = \left[5 \mid 0 \right].$$

O cálculo de y^2 e y^3 é obtido da seguinte forma:

$$\left[\begin{array}{c} y^2 \\ \vdots \\ y^3 \end{array} \right] = \left[\begin{array}{c|c} -I & 0 \\ \hline I & I \end{array} \right] \left\{ \left[\begin{array}{c} I \\ 0 \end{array} \right] - \left[\begin{array}{c} 0 \\ 0 \end{array} \right] \right\} = \left[\begin{array}{c} -I \\ I \end{array} \right] \left\{ \begin{array}{l} \leftarrow x_{18} \\ \leftarrow S_2 \end{array} \right.$$

Pivoteamento

Para o problema de múltiplos produtos, a variável que parte da base quando uma folga entra pode estar referida a um arco ou então a uma folga. Quando a variável que parte da base é um arco, existem duas possibilidades: a saída de um arco da árvore básica ou então a saída do próprio arco complementar ao qual a folga está associada. Se a variável que parte da base é um arco complementar, então não há mais arco complementar associado à folga, ou seja, a restrição de agregação que é ativa passa a ser não ativa. E se a variável que parte da base é um arco da árvore, então o arco complementar associado à folga que entra na base torna-se um arco da árvore básica. Dessa forma o arco complementar re-conecta a árvore

básica a fim de não torná-la singular. No entanto, se a variável que parte da base é uma folga, então o arco complementar fica associado a essa nova folga que partiu da base dando lugar à folga que entra, observando que, a restrição associada à folga que entra na base é desativada para ativar a restrição associada à folga que sai da base.

Nessa iteração tem-se o caso (Entra: Folga e Sai: Arco da k -ésima árvore). Com a saída do arco x_{15} a árvore correspondente ao produto 1 fica desconexa, então um arco complementar é buscado afim de re-conectar a árvore, senão a base se tornaria singular. Logo, o arco complementar x_{18} irá substituir o arco x_{15} que parte da árvore básica no produto 1. As restrições são desativadas, não existindo na próxima iteração arco complementar. A coluna da matriz Y correspondente ao arco complementar x_{18} é substituída pela folga S_1 que entra na base. Na próxima iteração a $\dim Q = 0$, ou seja, a matriz ciclo Y não possui linha ativa e coluna associada a arco complementar.

Iteração 2)

Na iteração 2, não há arcos complementares, logo as variáveis duais são obtidas diretamente nas árvores básicas não havendo, portanto, o ajuste do valor dual π^2 . As variáveis candidatas a entrar na base são x_{19} , x_{110} e x_{26} , x_{27} para os produtos 1 e 2, respectivamente. A variável escolhida para entrar na base é o arco x_{26} (produto2), $\delta \leftarrow -1$.

No teste de razão não existe y^2 , pois não há arcos complementares e a matriz Y é a matriz identidade, já que as colunas de Y estão somente associadas a folgas das restrições não ativas de agregação. Portanto, uma quantidade Δ de fluxo é enviada ao redor do ciclo formado com o arco não básico x_{26} fazendo com que o arco x_{29} parta da base. As folgas S_1 e S_2 das restrições não ativas são ajustadas para manter a viabilidade com respeito à capacidade de agregação dos fluxos. O fluxo nos arcos do produto 1 não é modificado e o teste de razão é mostrado abaixo:

$$y^{11} = \begin{bmatrix} x_1 & x_{11} & x_{12} & x_{13} & x_{14} & x_{18} & x_{16} & x_{17} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$y^{12} = \begin{bmatrix} x_2 & x_{21} & x_{22} & x_{23} & x_{24} & x_{28} & x_{29} & x_{210} \\ 0 & -1 & 1 & -1 & 1 & 0 & 1 & 0 \end{bmatrix},$$

$$y^2 = Y^{-1}[\mu^2 - D^2(B^2)^{-1}\alpha^2] = I \left[\begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right] = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \leftarrow \begin{matrix} S_1 \\ S_2 \end{matrix},$$

$$\Delta_1 = \min\{x_{22}, x_{24}, x_{29}, S_1\} = \min\{15, 15, 10, 10\} = 10,$$

$$\Delta_2 = \infty,$$

$$\Delta = 10.$$

Atualização dos Fluxos

$$x_{26} \leftarrow 10$$

$$x_{21} = 0 - 1(-1)10 = 10$$

$$x_{22} = 15 - 1(1)10 = 5$$

$$x_{23} = 0 - 1(-1)10 = 10$$

$$x_{24} = 15 - 1(1)10 = 5$$

$$x_{29} = 10 - 1(1)10 = 0$$

$$S_1 = 10 - 1(1)10 = 0$$

$$S_2 = 5 - 1(-1)10 = 15.$$

Observa-se que o fluxo no produto 1 é mantido.

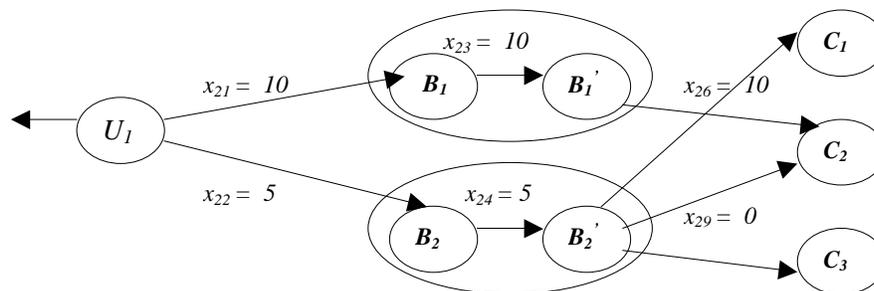


Figura I-10 - Atualização dos Fluxos nos arcos do produto 2 na iteração 2.

Pivoteamento

Entra: Arco Sai: Arco da k -ésima árvore

O arco x_{26} que entra substitui o arco x_{29} que parte da árvore básica no produto 2, podendo ser observado que o arco x_{26} faz a conexão da árvore após a saída do arco x_{29} , pois os dois arcos pertencem ao mesmo ciclo, e nessa iteração não há arco complementar. As restrições continuam desativadas, não existindo na próxima iteração arco complementar.

Iteração 3)

Na iteração 3, as variáveis duais são computadas para os dois produtos como na iteração anterior, observando que não existe π^2 . As variáveis candidatas são x_{19} e x_{25} , x_{27} para os produtos 1 e 2. A variável escolhida para entrar na base é o arco x_{19} (produto 1), $\delta \leftarrow 1$.

Uma quantidade Δ de fluxo é enviada ao redor do ciclo formado com o arco x_{19} , no entanto essa quantidade Δ não é suficiente para que algum arco desse ciclo saia da base. Logo o arco que entra x_{19} se torna um arco complementar ativando a restrição $x_{14} + x_{24} + S_2 = 40$, que passa a ter folga nula, pois os arcos x_{14} e x_{24} encontram-se saturados para a capacidade agregada. O fluxo no produto 2 é mantido devido ao teste de razão como mostrado abaixo.

$$y^{11} = (B^1)^{-1} \alpha^1 = \begin{bmatrix} x_1 & x_{11} & x_{12} & x_{13} & x_{14} & x_{18} & x_{16} & x_{17} \\ 0 & 1 & -1 & 1 & -1 & 0 & 1 & 0 \end{bmatrix},$$

$$y^{12} = \begin{bmatrix} x_2 & x_{21} & x_{22} & x_{23} & x_{24} & x_{28} & x_{26} & x_{210} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$y^2 = Y^{-1} [\mu^2 - D^2 (B^2)^{-1} \alpha^2] = I \left[\begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right] = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \leftarrow S_1, \right.$$

$$\Delta_1 = \min\{x_{11}, x_{13}, x_{16}, S_2\} = \min\{40, 40, 20, 15\} = 15,$$

$$\Delta_2 = \infty,$$

$$\Delta = 15.$$

Atualização dos Fluxos

$$x_{19} \leftarrow 15$$

$$x_{11} = 40 - 1(1)15 = 25$$

$$x_{12} = 20 - 1(-1)15 = 35$$

$$x_{13} = 40 - 1(1)15 = 25$$

$$x_{14} = 20 - 1(-1)15 = 35$$

$$x_{16} = 20 - 1(1)15 = 5$$

$$S_1 = 0 - 1(-1)15 = 15$$

$$S_2 = 15 - 1(1)15 = 0.$$

Observa-se que o fluxo no produto 2 é mantido.

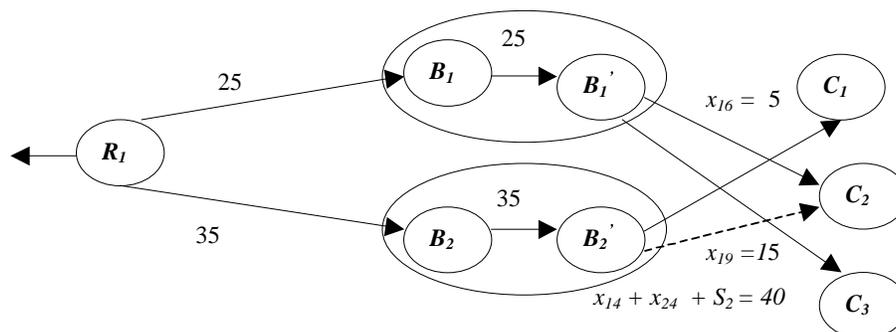


Figura I-11 - Atualização dos Fluxos nos arcos do produto 1 na iteração 3.

Pivoteamento

Entra: Arco Sai: Folga

Entrou o arco x_{19} e partiu a folga S_2 da base. A restrição $x_{14} + x_{24} + S_2 = 40$ é ativada e o arco x_{19} torna-se arco complementar.

Iteração 4)

Na iteração 4, é necessário determinar o valor dual π^2 e o valor de y^2 no teste de razão, pois a matriz ciclo Y possui uma coluna com o arco complementar x_{19} associado à folga S_2 .

Tem-se, $Y = \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}$, onde $Y^{-1} = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix}$. A segunda linha da matriz ciclo se encontra ativa assim como a segunda coluna está associada ao arco complementar x_{19} .

Para as variáveis duais tem-se $[\pi^3 \mid \pi^2] = [0 \mid -1]$ e a correção do valor dual foi no arco $B_2 B_2'$, obtendo-se as seguintes variáveis duais:

$$\pi^{11} = \begin{bmatrix} R_1 & B_1 & B_1' & B_2 & B_2' & C_1 & C_2 & C_3 \\ 0 & -5 & -5 & -6 & -7 & -11 & -13 & -14 \end{bmatrix} \text{ e}$$

$$\pi^{12} = \begin{bmatrix} U_1 & B_1 & B_1' & B_2 & B_2' & C_1 & C_2 & C_3 \\ 0 & -3 & -3 & -4 & -5 & -11 & -11 & -15 \end{bmatrix}.$$

As variáveis candidatas a entrar na base são x_{25} e x_{27} para o produto 2. O arco x_{25} é escolhido para entrar na base, $\delta \leftarrow I$.

Teste de Razão

Quando uma quantidade Δ de fluxo é enviada ao redor do ciclo formado com o arco não básico x_{25} , o arco x_{24} tem seu fluxo diminuído por essa quantidade Δ . Como o arco x_{24} pertence à restrição ativa para o arco complementar x_{19} (produto 1), essa quantidade Δ deve ser acrescida no arco x_{14} de modo a manter o equilíbrio exigido para as restrições. Nessa iteração deve ser observado que o fluxo nos dois produtos é alterado pela entrada do arco x_{25} diferente do que acontecia nas iterações anteriores que alterava somente o fluxo no produto para o arco que entrava. A alteração de fluxo nos dois produtos pode ser observada através do teste de razão para o arco complementar x_{19} que passa a ter seu fluxo aumentado pela quantidade Δ devido à entrada do arco x_{25} .

$$y^2 = \left[\begin{array}{c|c} 1 & 1 \\ \hline 0 & 1 \end{array} \right] \left\{ \begin{array}{l} [0] \\ [0] \end{array} \right\} - \begin{array}{l} [-1] \\ [1] \end{array} \left\} = \begin{array}{l} [0] \\ [-1] \end{array} \left\} \leftarrow \begin{array}{l} S_1 \\ x_{19} \end{array} \text{ onde, obtém-se:}$$

$$y^{11} = -(B^1)^{-1} R^1 y^{21} = \begin{bmatrix} x_1 & x_{11} & x_{12} & x_{13} & x_{14} & x_{18} & x_{16} & x_{17} \\ 0 & 1 & -1 & 1 & -1 & 0 & 1 & 0 \end{bmatrix},$$

$$y^{12} = (B^2)^{-1} \alpha^2 = \begin{bmatrix} x_2 & x_{21} & x_{22} & x_{23} & x_{24} & x_{28} & x_{26} & x_{210} \\ 0 & -1 & 1 & -1 & 1 & 1 & 0 & 0 \end{bmatrix},$$

$$\Delta_1 = \min\{x_{11}, x_{13}, x_{16}, x_{22}, x_{24}, x_{28}, \} = \min\{25, 25, 5, 5, 5, 5\} = 5,$$

$$\Delta_2 = \infty,$$

$$\Delta = 5.$$

Atualização dos Fluxos

$$x_{25} \leftarrow 5$$

$$x_{11} = 25 - 1(1)5 = 20 \quad , \quad x_{21} = 10 - 1(-1)5 = 15$$

$$x_{12} = 35 - 1(-1)5 = 40 \quad , \quad x_{22} = 5 - 1(1)5 = 0$$

$$x_{13} = 25 - 1(1)5 = 20 \quad , \quad x_{23} = 10 - 1(-1)5 = 15$$

$$x_{14} = 35 - 1(-1)5 = 40 \quad , \quad x_{24} = 5 - 1(1)5 = 0$$

$$x_{16} = 5 - 1(1)5 = 0 \quad , \quad x_{28} = 5 - 1(1)5 = 0$$

$$x_{19} = 15 - 1(-1)5 = 20.$$

Observa-se que nessa iteração existe atualização nos fluxos para ambos os produtos.

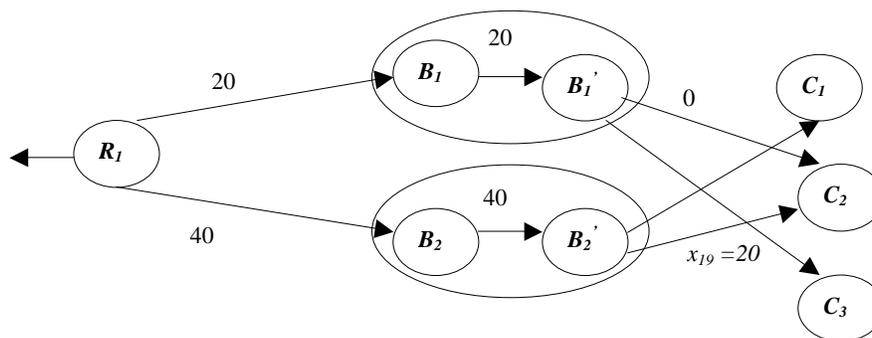


Figura I-12 - Atualização dos Fluxos nos arcos do produto 1 na iteração 4.

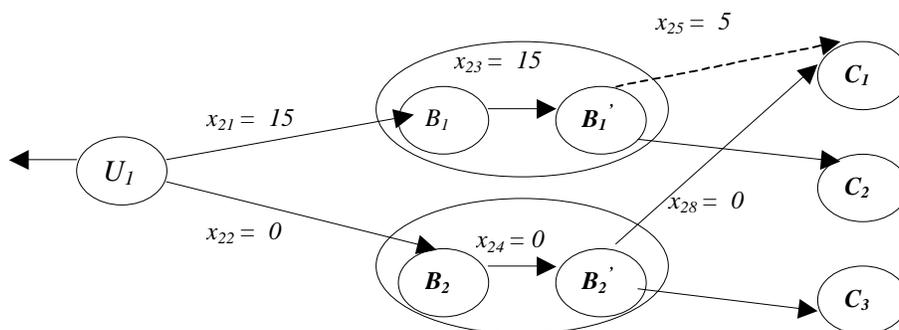


Figura I-13 - Atualização dos Fluxos nos arcos do produto 2 na iteração 4.

Pivoteamento

Nessa iteração, existe a possibilidade da saída de um arco da árvore básica, cujo produto do arco que parte não corresponde ao produto do arco que entra. Portanto, dois casos podem ser considerados.

Caso (a) -Entra: Arco x_{25} do produto 2 Sai: Arco x_{16} da árvore relativa ao produto 1.

Quando a quantidade Δ aumenta o fluxo que passa através do arco complementar x_{19} , o arco x_{16} atinge o limite inferior e, portanto é candidato a sair da base. O arco complementar x_{19} é então buscado para re-conectar a árvore e substituir o arco x_{16} que sai. O arco x_{25} que entra, torna-se o arco complementar associado à folga S_2 . As árvores básicas para ambos os produtos após pivoteamento sobre o arco x_{16} , são mostradas nas figuras 1-14 e I-15.

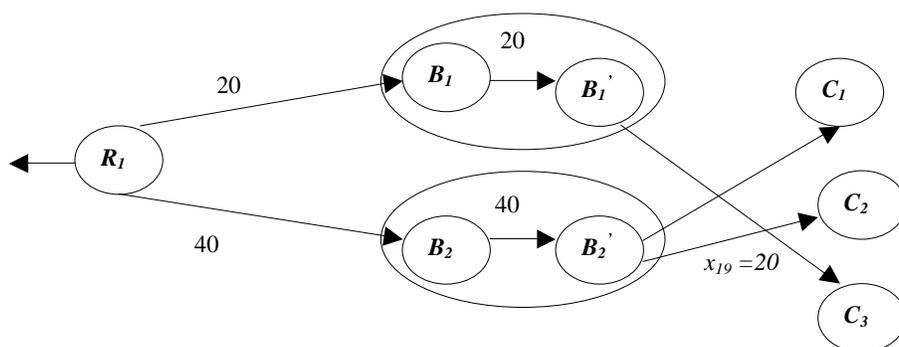


Figura I-14 - Árvore básica do produto 1 após pivoteamento sobre o arco x_{16} .

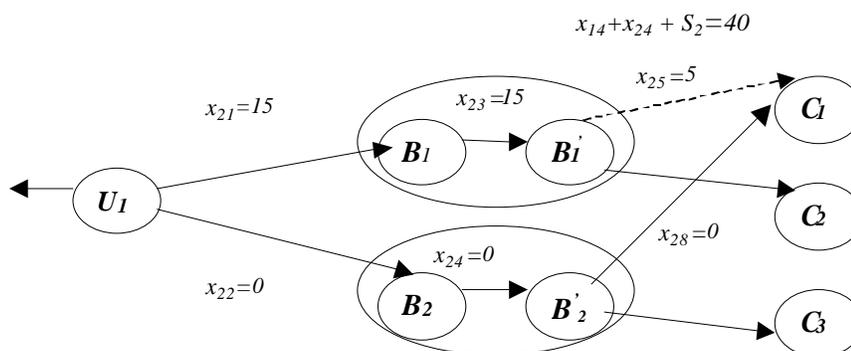


Figura I-15 - Árvore básica do produto 2 após pivoteamento sobre o arco x_{16} .

Caso (b) - Entra: Arco do produto 2 Sai: Arco da árvore relativa ao produto 2.

Suponha que a escolha fosse sobre o arco x_{28} para sair da base. O arco x_{25} que entra torna-se arco da árvore básica para o produto 2. As figuras I-16 e I-17 mostram as árvores básicas para ambos os produtos após pivoteamento sobre o arco x_{28} .

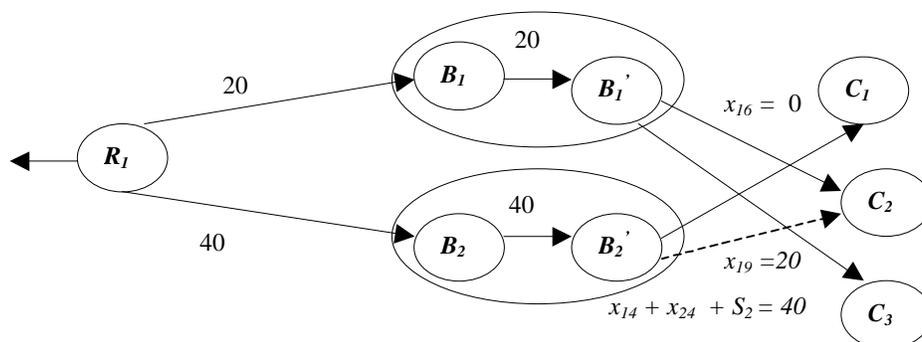


Figura I-16 - Árvore básica do produto 1 após pivoteamento sobre o arco x_{28} .

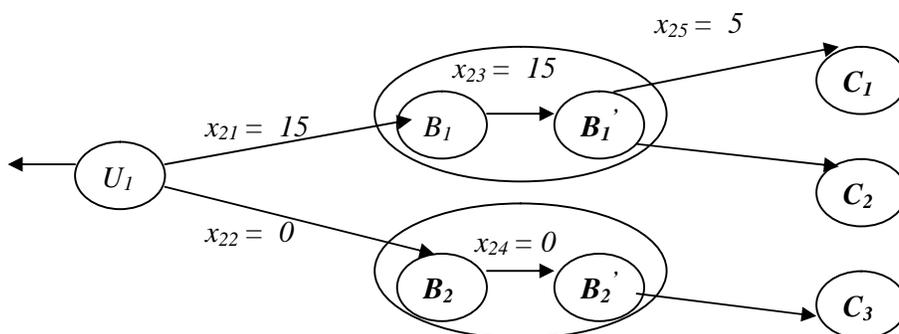


Figura I-17 - Árvore básica do produto 2 após pivoteamento sobre o arco x_{28} .

Na possibilidade, de um ou outro caso descrito acima ocorrer, a preferência dar-se-á pelo caso (b) entrada e saída de um arco correspondente ao mesmo produto na árvore básica.

Iteração 5)

Nessa iteração existe arco complementar, a matriz $Y = \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}$, onde

$$Y^{-1} = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix}.$$

Para as variáveis duais tem-se $\pi^2 = [0 \mid -I]$ e

$$\pi^{11} = \begin{bmatrix} R_1 & B_1 & B_1' & B_2 & B_2' & C_1 & C_2 & C_3 \\ 0 & -5 & -5 & -6 & -7 & -11 & -13 & -14 \end{bmatrix} e$$

$$\pi^{12} = \begin{bmatrix} U_1 & B_1 & B_1' & B_2 & B_2' & C_1 & C_2 & C_3 \\ 0 & -3 & -3 & -4 & -5 & -10 & -11 & -15 \end{bmatrix}.$$

Escolhe-se o arco x_{27} para entrar na base (produto2), $\delta \leftarrow -I$.

Teste de Razão

$$y^2 = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix} \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -I \\ I \end{bmatrix} \right\} = \begin{bmatrix} 0 \\ -I \end{bmatrix} \left\{ \begin{array}{l} \leftarrow S_1 \\ \leftarrow x_{19} \end{array} \right. \text{ onde, obtém-se:}$$

$$y^{11} = -(B^1)^{-1} R^1 y^{21} = \begin{bmatrix} x_1 & x_{11} & x_{12} & x_{13} & x_{14} & x_{18} & x_{16} & x_{17} \\ 0 & 1 & -1 & 1 & -1 & 0 & 1 & 0 \end{bmatrix},$$

$$y^{12} = (B^2)^{-1} \alpha^2 = \begin{bmatrix} x_2 & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & x_{210} \\ 0 & -1 & 1 & -1 & 1 & 0 & 0 & 1 \end{bmatrix},$$

$$\Delta_1 = \min\{x_{11}, x_{13}, x_{16}, x_{22}, x_{24}, x_{210}\} = \min\{20, 20, 0, 0, 0, 0\} = 0,$$

$$\Delta_2 = \infty,$$

$$\Delta = 0.$$

Atualização dos Fluxos

Continua a mesma solução, pois $\Delta = 0$.

Pivoteamento

Entra o arco x_{27} que substitui o arco x_{210} que parte da base, ambos os arcos de um mesmo produto. As novas bases são representadas abaixo:

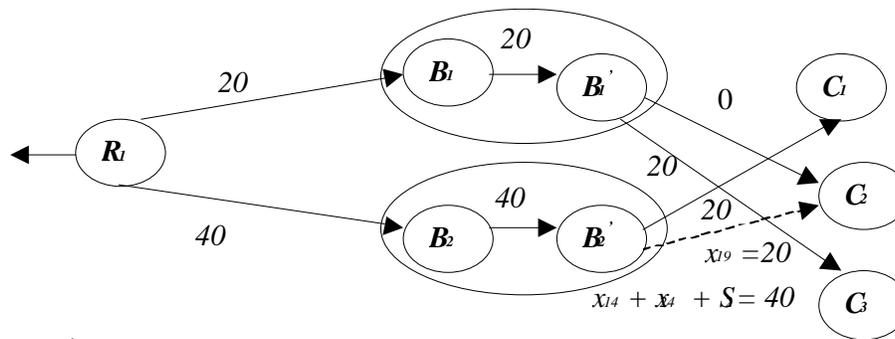


Figura I-18 - Árvore básica do produto 1

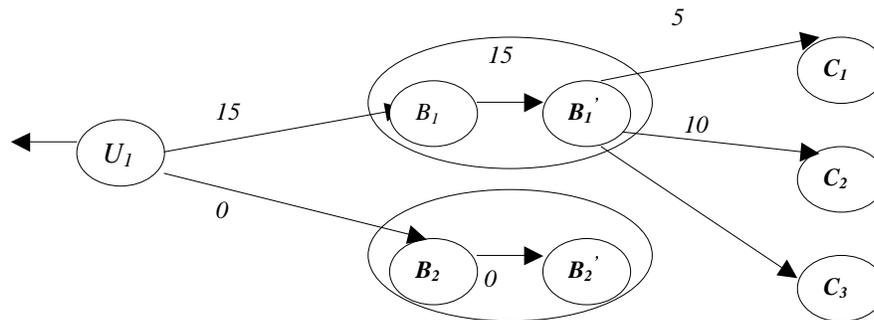


Figura I-19 - Árvore básica do produto 2

Iteração 6)

Como na iteração anterior tem-se

$$Y = \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}, \text{ onde } Y^{-1} = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix}.$$

Para as variáveis duais tem-se $\pi^2 = [0 \mid -I]$ e

$$\pi^{11} = \begin{bmatrix} R_1 & B_1 & B_1' & B_2 & B_2' & C_1 & C_2 & C_3 \\ 0 & -5 & -5 & -6 & -7 & -11 & -13 & -14 \end{bmatrix} e$$

$$\pi^{12} = \begin{bmatrix} U_1 & B_1 & B_1' & B_2 & B_2' & C_1 & C_2 & C_3 \\ 0 & -3 & -3 & -4 & -5 & -10 & -11 & -12 \end{bmatrix}.$$

Verifica-se que, folga S_2 e nenhum arco é mais candidato a entrar na base para ambos os produtos e, portanto estamos na solução ótima, com custo ótimo equivalente a 880.

II REDE COM DISTRIBUIÇÃO DE PRODUTOS COMPOSTOS PARA ILUSTRAR PASSO A PASSO TRÊS ITERAÇÕES DO ALGORITMO PARTICIONADO PRIMAL QUE UTILIZA A MATRIZ CICLO Y .

Três iterações do algoritmo primal particionado são realizadas sobre o exemplo do problema MPC apresentado no capítulo IV. Na iteração 1, a matriz ciclo Y é a matriz identidade, pois nessa iteração todas as folgas se encontram na base e o valor dual $\pi^2 = 0$.

As variáveis duais para os produtos A e B são:

$$\pi^{11} = \begin{bmatrix} S_a & R_1 & R_2 & A_1 & M_1 & M_{11} & A_2 & M_2 & M_{22} & C_1 & C_2 & C_3 \\ 0 & 0 & 0 & -3 & -3 & -3 & -2 & -2 & -2 & -100 & -100 & -100 \end{bmatrix} \text{ e}$$

$$\pi^{12} = \begin{bmatrix} S_b & U_1 & B_2 & M_1 & M_{11} & B_2 & M_2 & M_{22} & C_1 & C_2 & C_3 \\ 0 & 0 & -3 & -3 & -3 & -4 & -4 & -4 & -100 & -100 & -100 \end{bmatrix}.$$

O arco escolhido para entrar na base é o arco $M_{22}C_1(a)$, $\delta \leftarrow -1$.

No teste de razão, determina-se:

$$y^{11} = \begin{bmatrix} S_a & S_a R_1 & S_a R_2 & R_1 A_1 & A_1 M_1 & M_1 M_{11}(a) & R_2 A_2 & A_2 M_2 & M_2 M_{22}(a) & S_a C_1 & S_a C_2 & S_a C_3 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & -1 & -1 & 1 & 0 & 0 \end{bmatrix},$$

$$y^{12} = 0,$$

$$y^2 = I[\mu^k - D^k(B^k)^{-1}\alpha^k] = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1.3333 \end{bmatrix} \leftarrow \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{matrix},$$

$$\Delta_1 = \min \{S_a C_1, S_3, S_4\} = 0,$$

$$\Delta_2 = \infty.$$

Portanto, $\Delta = 0$, a folga S_4 é candidata a entrar na base. A restrição de proporcionalidade é ativada. O arco $M_{22}C_1(a)$ torna-se complementar associado à folga S_4 .

A solução não é modificada.

Na iteração 2, é necessário determinar a matriz ciclo.

$$Y = \left[\begin{array}{ccc|c} 1 & 0 & 0 & M_{22}C_1(a) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1.333 \end{array} \right] \text{ com } Y^{-1} = \left[\begin{array}{ccc|c} 1 & 0 & 0 & M_{22}C_1(a) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.75 \\ \hline 0 & 0 & 0 & 0.75 \end{array} \right]$$

Podemos determinar π^2 diretamente na árvore básica associada ao produto A .

No ciclo $\left\{ \overbrace{M_{22}M_2}^{2.5} \quad \overbrace{M_2A_2}^0 \quad \overbrace{A_2R_2}^{-2} \quad \overbrace{R_2S_a}^0 \quad \overbrace{S_aC_1}^{100} \right\}$ tem-se:

$$\sum_{i=1}^n c_i O_i(C) A(j_i) = 2.5 - (-2+100) = -95.5.$$

O preço π^2 altera o custo nos arcos $A_2M_2(a)$ e $B_2M_2(b)$, pois são os arcos que estão na árvore básica e aparecem na restrição ativa de agregação.

Logo

$$\pi^2 = [0 \ 0 \ 0 \ -95.5] Y^{-1} = [0 \ 0 \ 0 \ -71.625],$$

$$\pi^{11} = \begin{bmatrix} S_a & R_1 & R_2 & A_1 & M_1 & M_{11} & A_2 & M_2 & M_{22} & C_1 & C_2 & C_3 \\ 0 & 0 & 0 & -3 & -3 & -3 & -2 & -97.5 & -97.5 & -100 & -100 & -100 \end{bmatrix} e$$

$$\pi^{12} = \begin{bmatrix} S_b & U_1 & B_1 & M_1 & M_{11} & B_2 & M_2 & M_{22} & C_1 & C_2 & C_3 \\ 0 & 0 & -3 & -3 & -3 & -4 & 281 & 281 & -100 & -100 & -100 \end{bmatrix}.$$

O arco escolhido para entrar na base é o arco $M_{11}C_3(a)$, $\delta \leftarrow 1$.

No teste de razão, tem-se:

$$y^{11} = \begin{bmatrix} S_a & S_a R_1 & S_a R_2 & R_1 A_1 & A_1 M_1 & M_1 M_{11}(a) & R_2 A_2 & A_2 M_2 & M_2 M_{22}(a) & S_a C_1 & S_a C_2 & S_a C_3 \\ 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$y^{12} = 0,$$

$$y^2 = Y^{-1} \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 \\ -1.333 \\ 0 \\ 0 \end{bmatrix} \right\} = \begin{bmatrix} 1 \\ 1.333 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} \leftarrow S_1 \\ \leftarrow S_2 \\ \leftarrow S_3 \\ \leftarrow M_{22}C_1(a) \end{matrix},$$

$$\Delta_1 = \min \{ SaC_3, S_1, S_2 \} = 0,$$

$$\Delta_2 = \infty.$$

Portanto $\Delta = 0$, a folga S_2 é candidata a entrar na base. A restrição de proporcionalidade é ativada. O arco $M_{11}C_3(a)$ torna-se complementar associado à folga S_2 . A solução não é modificada. A figura II-1 mostra a nova base para o produto A.

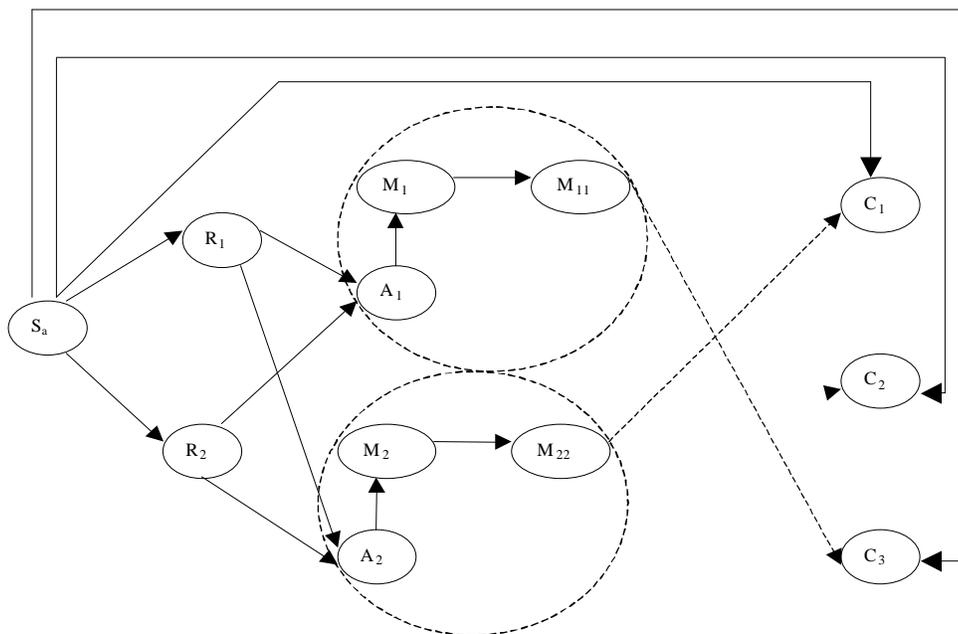


Figura II-1 - Árvore básica para o produto do tipo A contendo dois arcos complementares.

Na iteração 3, é necessário determinar a matriz ciclo.

$$Y = \begin{bmatrix} 1 & M_{11}C_3(a) & 0 & M_{22}C_1(a) \\ 0 & 1.333 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1.333 \end{bmatrix} \text{ com } Y^{-1} = \begin{bmatrix} 1 & -0.75 & 0 & 0 \\ 0 & 0.75 & 0 & 0 \\ 0 & 0 & 1 & -0.75 \\ 0 & 0 & 0 & 0.75 \end{bmatrix}$$

Os valores duais são:

$$\pi^2 = [0 \quad -93 \quad 0 \quad -95.5] Y^{-1} = [0 \quad -69.76 \quad 0 \quad -71.625].$$

O preço π^2 altera o custo nos arcos $A_1M_1(a)$, $A_2M_2(a)$, $B_1M_1(b)$ e $B_2M_2(b)$, pois são os arcos que estão na árvore básica e aparecem na restrição ativa de agregação.

$$\pi^{11} = \begin{bmatrix} S_a & R_1 & R_2 & A_1 & M_1 & M_{11} & A_2 & M_2 & M_{22} & C_1 & C_2 & C_3 \\ 0 & 0 & 0 & -3 & -96 & -96 & -2 & -97.5 & -97.5 & -100 & -100 & -100 \end{bmatrix} \text{ e}$$

$$\pi^{12} = \begin{bmatrix} S_b & U_1 & B_1 & M_1 & M_{11} & B_2 & M_2 & M_{22} & C_1 & C_2 & C_3 \\ 0 & 0 & -3 & 276 & 276 & -4 & 283 & 283 & -100 & -100 & -100 \end{bmatrix}$$

O arco escolhido para entrar na base é o arco $M_{11}C_1(b)$, $\delta \leftarrow 1$.

No teste de razão, tem-se:

$$y^2 = \begin{bmatrix} 4 \\ -3 \\ 0 \\ 0 \end{bmatrix} \left\{ \begin{array}{l} \leftarrow S_1 \\ \leftarrow M_{11}C_3(a) \\ \leftarrow S_3 \\ \leftarrow M_{22}C_1(a) \end{array} \right. ,$$

$$y^{11} = -(B^1)^{-1}R^1y^{21}, \quad y^{21} = -3,$$

$$y^{11} = \begin{bmatrix} S_a & S_aR_1 & S_aR_2 & R_1A_1 & A_1M_1 & M_1M_{11}(a) & R_2A_2 & A_2M_2 & M_2M_{22}(a) & S_aC_1 & S_aC_2 & S_aC_3 \\ 0 & -3 & 0 & -3 & -3 & -3 & 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix},$$

$$y^{12} = \begin{bmatrix} S_b & S_bU_1 & U_1B_1 & U_1B_2 & B_1M_1 & M_1M_{11}(b) & B_2M_2 & M_2M_{22}(b) & S_bC_1 & S_bC_2 & S_bC_3 \\ 0 & -1 & -1 & 0 & -1 & -1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

$$\Delta_1 = \min \{S_aC_3, S_bC_1, S_1\} = \min \left\{ \frac{30}{3}, 7.5, \frac{30}{4} \right\} = 7.5,$$

$$\Delta_2 = \infty. \text{ Portanto } \Delta = 7.5.$$

Atualização dos fluxos para ambos os produtos.

$$M_{11}C_1(a) \leftarrow 0 - (-1)(7.5) = 7.5$$

$$S_aR_1 \leftarrow 0 - (-3)7.5 = 22.5$$

$$R_1A_1 \leftarrow 0 - (-3)7.5 = 22.5$$

$$A_1M_1(a) \leftarrow 0 - (-3)7.5 = 22.5$$

$$M_1M_{11}(a) \leftarrow 0 - (-3)7.5 = 22.5$$

$$S_aC_3 \leftarrow 30 - (-3)7.5 = 7.5$$

$$M_{11}C_3(a) = 0 - (-3)7.5 = 22.5$$

$$S_bU_1 \leftarrow 0 - (-1)(7.5) = 7.5$$

$$U_1B_1 \leftarrow 0 - (-1)(7.5) = 7.5$$

$$B_1M_1 \leftarrow 0 - (-1)(7.5) = 7.5$$

$$M_1M_{11}(b) \leftarrow 0 - (-1)(7.5) = 7.5$$

$$S_bC_1 \leftarrow 7.5 - (7.5) = 0$$

$$S_1 \leftarrow 30 - 4(7.5) = 0.$$

A variável escolhida para sair da base é o arco S_bC_1 . Tem-se o caso entra arco e sai arco da k -ésima árvore. Na próxima iteração a matriz ciclo permanece a mesma.

Mostrou-se nesse exemplo, três iterações completas em uma rede representando dois produtos onde são incluídas as frações dos produtos compostos.

ANEXOS

A.1 CÁLCULO DAS VARIÁVEIS DUAS

A.2 FORMA PRODUTO DA INVERSA

A.3 ALGORITMO DIJKSTRA (Caminhos Mínimos)

A.1. CÁLCULO DAS VARIÁVEIS DUAIS

- P0. **Inicialização.** Faça $\tau_B = [N, A]$ denotar a árvore base com nó raiz ℓ . Faça $\pi_\ell = 0$, $N^L = \{\ell\}$, e $N^U = N - \{\ell\}$.
- P1. **Encontre um arco com Nó $T(j)$ rotulado.** Seja $e_j \in A$ tal que $F(j) \in N^U$ e $T(j) \in N^L$. Se não existe este arco, vá ao passo P3.
- P2. **Rotule o nó $F(j)$.** Faça $\pi_{F(j)} = c_j + \pi_{T(j)}$, $N^L = N^L \cup \{F(j)\}$, $N^U = N^U - \{F(j)\}$, e vá ao passo P1.
- P3. **Encontre uma arco com Nó $F(j)$ rotulado.** Seja $e_j \in A$ tal que $T(j) \in N^U$ e $F(j) \in N^L$. Se não existe este arco, termine.
- P4. **Rotule o nó $T(j)$.** Faça $\pi_{T(j)} = -c_j + \pi_{F(j)}$, $N^L = N^L \cup \{T(j)\}$, $N^U = N^U - \{T(j)\}$ e vá ao passo P1.

A.2 FORMA PRODUTO DA INVERSA

Este anexo apresenta alguns resultados algébricos em relação aos elementos que são introduzidos no algoritmo particionado para inversão da matriz ciclo Y associada ao problema de fluxo em rede para múltiplos produtos. Grande parte do desempenho computacional do algoritmo que soluciona o problema proposto está associado à inversão da matriz ciclo Y . O elemento principal utilizado na inversão da matriz é a forma produto da inversa.

A.2.1 Produto de Matrizes Elementares

Teorema 1: O resultado de efetuar operações elementares com as linhas sobre uma matriz A , mxn , pode também ser achado formando o produto EA , onde E é uma matriz obtida efetuando-se as operações com linhas sobre a matriz unitária $I nxn$.

Prova: Suponha que A é uma matriz mxn e que as posições relativas da p -ésima e q -ésima linhas sejam trocadas. A matriz obtida trocando-se as posições relativas da p -ésima e q -ésima linhas da matriz identidade $I nxn$ é

$$E = \begin{bmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 0 \\ 0 & \dots & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & \dots & 1 \end{bmatrix} \quad (\text{A.1})$$

onde se indica as primeiras, p -ésimas, q -ésimas e últimas linhas e colunas. Vê-se facilmente que a matriz EA é de fato precisamente a matriz obtida trocando-se as posições relativas da p -ésima e q -ésima linhas de A . Semelhantemente, a matriz obtida multiplicando-se a p -ésima linha da matriz identidade $I nxn$ pelo escalar não nulo c é

$$E = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & c & \dots & 0 \\ \dots & 0 & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} \quad (\text{A.2})$$

e a matriz EA é precisamente a matriz obtida multiplicando-se a p -ésima linha de A por c . Finalmente, adicionando-se c vezes a q -ésima linha da matriz unitária I à p -ésima linha, obtém-se:

$$E = \begin{bmatrix} I & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & I & c & \cdots & 0 \\ 0 & \cdots & 0 & I & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & \cdots & I \end{bmatrix} \quad (\text{A.3})$$

onde é indica-se as primeiras, p -ésimas, q -ésima e últimas linhas e colunas. A matriz EA é precisamente a matriz obtida adicionando c vezes a q -ésima linha de A à p -ésima. Isso completa a demonstração do teorema.

Definição 1: Qualquer matriz E que difere da matriz identidade I somente por uma coluna é chamada de matriz coluna elementar.

É um fato notável e importante que cada matriz elementar E é não singular, ou seja, $H = E^{-1}$ existe e, além disso, E^{-1} é também uma matriz elementar do mesmo tipo. Ainda, se E_1, E_2, \dots, E_r são matrizes não singulares $n \times n$, então o produto $E_1 E_2 \dots E_r$ é não singular e $(E_1 E_2 \dots E_r)^{-1} = E_r^{-1} E_{r-1}^{-1} \dots E_1^{-1}$.

Pelo Teorema 1, cada operação com linhas pode ser efetuada por meio de pré-multiplicação por uma matriz elementar, e, portanto, que qualquer seqüência de operações com linhas pode ser representada por pré-multiplicações sucessivas por matrizes elementares E_1, E_2, \dots, E_r ; o efeito total de tais pré-multiplicações sucessivas é naturalmente o mesmo que o de uma pré-multiplicação por $F = E_r E_{r-1} \dots E_1$. Como o produto de matrizes não singulares é não singular e cada matriz elementar é não singular, tem-se o seguinte teorema.

Teorema 2: Se a matriz B resulta da matriz A por meio da aplicação de uma seqüência de operações elementares sobre A , então existe uma matriz não singular F tal que $B = FA$.

Pode-se ainda dizer que a matriz B obtida a partir de A por meio de uma seqüência de operações elementares é uma matriz equivalente por linhas a matriz A , e indica-se $B \sim A$.

A.2.2 Um método para determinar a inversa de uma matriz

Baseado no *teorema 1*, um método prático para calcular a inversa de uma matriz é o seguinte:

Forme a matriz $n \times 2n$ $[A \mid I_n]$ obtida juntando a matriz identidade I_n à matriz A que se quer inverter. Leve a matriz $[A \mid I_n]$ à forma reduzida escalonada por linhas usando operações elementares sobre linhas, tudo que for feito com uma linha de A deve também ser feito com a linha correspondente I_n . A matriz reduzida escalonada por linhas obtida é $[I \mid A^{-1}]$.

Considere o exemplo abaixo:

Se utilizar na matriz $A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 5 & 5 & 1 \end{bmatrix}$ uma seqüência de operações elementares com linhas

para reduzi-la a forma reduzida escalonada por linhas; a cada operação com linhas utilizada em A encontrar-se-á uma matriz elementar correspondente E^{-1} , verificar-se-á que a pré-multiplicação por E^{-1} terá o mesmo resultado que a operação com linhas.

Primeiro Passo: Ache a primeira coluna de A (contando da esquerda para direita) cujos coeficientes não são todos nulos. Esta coluna é chamada de coluna pivô e o primeiro elemento não nulo é o número 1 chamado de elemento pivô.

Segundo Passo: Subtraia 5 vezes a primeira linha da terceira na matriz A , e obtenha

$$A_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 0 & 0 & -4 \end{bmatrix}$$

A mesma operação elementar sobre a matriz A_1 faz por obter a matriz elementar

correspondente $E_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -5 & 0 & 1 \end{bmatrix}$, onde pode ser verificado que o produto $E_1^{-1} A = A_1$.

Terceiro Passo: Divida a segunda linha por 2 na matriz A_1 , assim o elemento pivô da segunda coluna é agora 1. A seguir, subtraia a segunda linha da primeira e obtenha:

$$A_2 = \begin{bmatrix} 1 & 0 & -1/2 \\ 0 & 1 & 3/2 \\ 0 & 0 & -4 \end{bmatrix}$$

As mesmas operações elementares descritas no terceiro passo são aplicadas a matriz I_3 ,

obtendo $E_2^{-1} = \begin{bmatrix} 1 & -1/2 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, onde o produto $E_2^{-1} A_1 = A_2$.

Quarto Passo: A fim de anular todos os coeficientes da coluna pivô, exceto o elemento pivô é que as seguintes operações elementares são realizadas:

Divida a terceira linha da matriz A_2 por (-4) , obtendo a matriz $\begin{bmatrix} 1 & 0 & -1/2 \\ 0 & 1 & 3/2 \\ 0 & 0 & 1 \end{bmatrix}$, dessa forma o

elemento pivô é igual a 1. A seguir adicione $\frac{1}{2}$ vez a terceira linha à primeira e obtenha

$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 3/2 \\ 0 & 0 & 1 \end{bmatrix}$. Finalmente, adicione $-\frac{3}{2}$ vezes a terceira linha à segunda e obtenha a matriz $A_3 = I_3$.

Faça todas as operações elementares na matriz I_3 mencionadas anteriormente e obtenha a

matriz elementar $E_3^{-1} = \begin{bmatrix} 1 & 0 & -1/8 \\ 0 & 1 & 3/8 \\ 0 & 0 & -1/4 \end{bmatrix}$, onde o produto $E_3^{-1} A_2 = I_3$.

Portanto podemos escrever a matriz $I_3 = E_3^{-1} A_2 = E_3^{-1} (E_2^{-1} A_1) = E_3^{-1} (E_2^{-1} (E_1^{-1} A))$.

A matriz I foi obtida a partir da matriz A através de operações elementares e de acordo com o teorema 2, $I = FA$, onde $F = E_3^{-1} E_2^{-1} E_1^{-1}$, e sendo a matriz A inversível tem-se $I = A^{-1}A$, e portanto $A^{-1} = E_3^{-1} E_2^{-1} E_1^{-1}$. Costuma-se dizer que a inversa da matriz A pode ser obtida utilizando a forma produto da inversa de matrizes elementares. Observe que, com a primeira coluna da matriz A pode-se obter a matriz elementar E_1 , com a segunda coluna da

matriz A_1 obter a matriz elementar E_2 e finalmente com a terceira coluna de A_2 obter a matriz elementar E_3 , tal que $A = E_1 E_2 E_3$.

Para apresentar o algoritmo que utiliza a forma produto da inversa é necessário fazer a inversão de matrizes elementares. A inversão de matrizes elementares sob o ponto de vista computacional é bastante eficiente, pois a matriz elementar difere da matriz identidade somente por uma coluna, denominada coluna r , com $e_r^r \neq 0$ (elemento pivô), onde ($e_h^i = E$) e ($\eta_h^i = E^{-1}$). Computar os elementos η_h^i a partir de e_h^i é como segue:

Seja ($e_h^i = E$)

$$\eta_h^i = e_h^i, \text{ quando } h \neq r \quad (\text{A.4})$$

$$\eta_r^r = \frac{1}{e_r^r} \quad (\text{A.5})$$

$$\eta_r^i = \frac{-e_r^i}{e_r^r}, \text{ quando } i \neq r. \quad (\text{A.6})$$

Para provar a veracidade das equações acima basta levar a matriz $[E \mid I_n]$ à forma reduzida escalonada por linhas usando operações elementares sobre linhas e obter a matriz reduzida escalonada por linhas $[I \mid E^{-1}]$.

Agora, a pré-multiplicação de uma matriz elementar E^{-1} por um vetor coluna a_j^i tem como resultado um vetor coluna α_j^i , a grande vantagem computacional desse produto é que em relação à matriz E^{-1} somente é necessário registrar o índice r e η_r^i (incluindo $i = r$). Com relação ao exemplo anterior, cada elemento da j -ésima coluna de A é indicado por a_j^i com $i = 1, \dots, n$. Quando é efetuado o produto $E^{-1} a_j^i = \alpha_j^i$, $i = 1, \dots, n$, com $j = 2$ indicando a segunda coluna da matriz A tem-se:

$$\begin{bmatrix} \eta^1 & & \\ \eta^2 & 1 & \\ \eta^3 & & 1 \end{bmatrix} \begin{bmatrix} a_2^1 \\ a_2^2 \\ a_2^3 \end{bmatrix} = \begin{bmatrix} \alpha_2^1 \\ \alpha_2^2 \\ \alpha_2^3 \end{bmatrix}, \text{ onde } \eta^1 = 1, \eta^2 = 0, \eta^3 = -5, a_2^1 = 1, a_2^2 = 2, a_2^3 = 5$$

$$\alpha_2^1 = a_2^1 \eta^1 = 1$$

$$\alpha_2^2 = a_2^1 \eta^2 + a_2^2 = 2$$

$$\alpha_2^3 = a_2^1 \eta^3 + a_2^3 = 0$$

é portanto fácil deduzir que

$$\alpha_j^i = a_j^i + a_j^r \eta^i, \quad i \neq r \quad (\text{A.7})$$

$$\alpha_j^r = a_j^r \eta^r \quad (\text{A.8})$$

Um procedimento para determinação das matrizes inversas elementares necessárias para a obtenção da forma produto da inversa é como segue:

Denota-se por

α_r o vetor correspondente a coluna r na matriz $A_{n \times n}$.

E_r matriz coluna elementar que difere da identidade pela coluna r .

E_r^{-1} inversa de uma matriz coluna elementar.

A.2.3 Procedimento para determinar as matrizes inversas elementares

Passo1 Inicialize $r = 1$.

Passo2 Determine a matriz coluna elementar E_r com $\alpha_r^r \neq 0$.

Passo3 Determine a inversa da matriz coluna elementar E_r^{-1} de acordo com (A.4), (A.5) e (A.6)

Passo 4 Registre E_r^{-1} .

Passo 5 Se $r < n$, atualize as colunas da matriz A de acordo com (A.7) e (A.8)

$$\alpha_r \leftarrow E_r^{-1} \alpha_r$$

$$\alpha_{r+1} \leftarrow E_r^{-1} \alpha_{r+1}$$

\vdots

$$\alpha_n \leftarrow E_r^{-1} \alpha_n$$

Incremente r , e retorne ao passo 2.

Senão vá para o passo 6.

Passo 6 O procedimento termina com as matrizes inversas elementares registradas

$$E_n^{-1}, E_{n-1}^{-1}, \dots, E_1^{-1}.$$

Considerando que a cada iteração a matriz ciclo Y possui uma coluna modificada pela substituição de uma coluna que sai por uma coluna que entra é que se pretende determinar uma expressão para Y_{i+1}^{-1} em termos de Y_i^{-1} . Em geral, a matriz ciclo Y_i (para a iteração i) e Y_{i+1} (para a iteração $i+1$) são diferentes.

De acordo com o teorema 4.2.1

$$(A.9) \quad Y_{i+1}^{-1} = E^{-1} Y_i^{-1}$$

onde, $Y_{i+1} = Y_i E$ e as matrizes Y_i e Y_{i+1} possuem uma coluna modificada pelo pivoteamento na iteração simplex.

Suponha que em uma iteração $(i+1)$ do algoritmo simplex se queira encontrar o vetor coluna $c^T = [w_1 \quad w_2 \quad w_3]$ da matriz elementar E , de modo a obter o produto $Y'_{i+1} = Y_i E$, tal que Y'_{i+1} difere apenas por uma coluna da matriz Y_i . Suponha que $Y_i = A$, onde A corresponde a matriz do exemplo anterior.

Dado que $Y'_{i+1} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 5 & 4 & 1 \end{bmatrix}$, difere da matriz Y_i apenas pela segunda coluna, tem-se

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 5 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1 & w_1 & 0 \\ 0 & w_2 & 0 \\ 0 & w_3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 5 & 4 & 1 \end{bmatrix},$$

onde a solução do sistema $\begin{cases} w_1 + w_2 + w_3 = 1 \\ 0 + 2w_2 + 3w_3 = 1 \\ 5w_1 + 5w_2 + w_3 = 4 \end{cases}$, fornece o vetor coluna $c = \begin{pmatrix} 5/8 \\ 1/8 \\ 1/4 \end{pmatrix}$.

Equivalentemente, $Y_i c = v$, onde $v^T = [1 \quad 1 \quad 4]$, portanto $c = Y_i^{-1} v = E_3^{-1} (E_2^{-1} (E_1^{-1} v))$.

Logo, a matriz elementar correspondente a segunda coluna modificada é a matriz

$$E_2 = \begin{bmatrix} 1 & 5/8 & 0 \\ 0 & 1/8 & 0 \\ 0 & 1/4 & 1 \end{bmatrix}, \text{ a matriz obtida para se ter } Y'_{i+1} = Y_i E_2. \text{ Por outro lado, pretende-}$$

se a cada iteração, obter a matriz inversa da nova matriz em função da matriz inversa da iteração anterior, busca-se resolver:

$$E_2^{-1} Y_i^{-1} = Y'_{i+1}$$

Como a matriz Y_i^{-1} se encontra sob a forma produto da inversa e já foi obtida em uma iteração anterior, é preciso então, inverter a matriz elementar E_2 que foi obtida a partir do vetor coluna c . Como no problema proposto a matriz ciclo é bastante esparsa, a forma produto

da inversa é bastante eficiente. Além do mais o custo computacional para inverter uma matriz elementar é pequeno.

Toda técnica de inversão utilizando matrizes elementares pode ser encontrada em [May84].

A.3. ALGORITMO DE DIJKSTRA ($c_{ij} > 0$)

Dado um grafo avaluado $G = [\mathcal{N}, \mathcal{R}]$, com arcos de custos dados em uma matriz $C=[c_{ij}]$, o problema do Caminho Mínimo, é um problema para encontrar o mínimo caminho de um específico vértice inicial $s \in \mathcal{N}$ para um específico vértice final $t \in \mathcal{N}$, sendo que este caminho existe, isto é, contanto que $t \in \mathbf{R}(s)$, onde $\mathbf{R}(s)$ é o conjunto de alcance do vértice s .

O mais eficiente algoritmo para a solução de problemas de caminhos mínimos de s para t , foi dado por Dijkstra. O método é baseado na atribuição de rótulos temporários para vértices, o rótulo sobre o vértice é o limite superior do comprimento do caminho de s para aqueles vértices. Esses rótulos são unicamente reduzidos por processo de iteração e, a cada iteração exatamente um dos rótulos temporários fica permanente, indicando que ele não é mais o limite superior, mas o exato comprimento de caminho mínimo de s para o vértice em questão. O algoritmo de Dijkstra segue abaixo:

c_{ij} matriz custo com valores positivos

Faça $l(x_i)$ ser rotulado sobre o vértice x_i

Inicialização

P1. Faça $l(s) = 0$ e faça o rótulo como permanente.

Faça $l(x_i) = \infty$ para todos $x_i \neq s$ e marque esses rótulos temporariamente.

Faça $p = s$

Atualização dos rótulos

P2. Para todos $x_i \in \mathcal{R}(p)$ os quais possuem rótulos temporários, atualize os rótulos de acordo com a equação:

$$l(x_i) = \min[l(x_i), l(p) + c(p, x_i)]$$

Fixando um rótulo como permanente

P3. De todos os vértices rotulados temporariamente encontre x_i^* para o qual

$$l(x_i^*) = \min [l(x_i)]$$

P4. Marque o rótulo permanente para x_i^* e faça $p = x_i^*$

P5. (i) (Se somente o caminho de s para t é desejado)

Se $p = t$, $l(p)$ é requerido como comprimento de mínimo caminho. Pare.

Se $p \neq t$, vá para P2.

(ii) (Se o caminho de s para todos os outros vértices é requerido)

Se todos os vértices estão permanentemente rotulados, então os rótulos estão no comprimento do mínimo caminho. Pare.

Se alguns rótulos são temporários vá para P2.

Uma vez o comprimento do mínimo caminho de s for obtido como o valor final do vértice rotulado, o caminho mínimo pode ser obtido por uma avaliação da equação abaixo:

$$l(x_i'') + c(x_i', x_i) = l(x_i)$$

Se o caminho mínimo de s para algum x_i é único, então o arco (x_i', x_i) é o mínimo caminho para uma árvore direcionada, com s sendo o nó raiz.

Se existe mais um “mínimo” caminho de s para algum outro vértice, então a equação acima será satisfeita por mais um vértice x_i' para algum x_i . Nesse caso cada escolha arbitrária pode ser feita (se somente um mínimo caminho entre s e x_i é requerido).