

Índice

1-BREVE HISTÓRIA SOBRE A PROGRAMAÇÃO LINEAR	1
2- INTRODUÇÃO MATEMÁTICA	5
2.1- Notações e convenções.....	5
2.2- Programação linear em redes.....	7
2.3- Exemplos de modelos em programação linear	15
2.3.1- Produção - distribuição da polímeros	15
2.3.2- Problema das dietas	22
2.3.3- Financiamento de projectos	23
2.3.4- Planeamento em aproveitamentos hidroeléctricos, curto prazo, modelo LP	24
2.3.5- Planeamento em aproveitamentos de biogas, curto prazo, modelo LP.....	32
3- PROGRAMAÇÃO LINEAR (PL).....	36
3.1- Propriedades do espaço vectorial euclidiano	39
3.2- Caracterização do domínio das soluções possíveis	45
3.3- Método primal simplex	50
3.4- Condições de Karush-Kuhn-Tucker para a programação linear.....	71
3.5- Teoria da dualidade	74
4- MÉTODO SIMPLEX PARA PROGRAMAÇÃO EM REDES.....	83
4.1- Resultados da teoria dos grafos	84
4.2- Caracterização de uma rede.....	97
4.3- Especialização primal simplex	107
4.4- Solução básica possível inicial	115
4.5- Estratégias de avaliação.....	117
5-ALGORITMO DA CONDIÇÃO PARA A PROGRAMAÇÃO EM REDES ...	118
5.1- Condições de óptimo	119
5.2- Metodologia para o algoritmo.....	121
5.3- Fase primal	123
5.4- Fase dual	127
5.5- Algoritmo da condição	131
5.6- Exemplo de iniciar o algoritmo da condição por nó e arcos artificiais	142
5.7- Convergência do Algoritmo da Condição	155
6- CÓDIGO LSNNO, CASCATA HÍDRICA	156
6.1- Simulação da Gestão de cascatas hídricas	156
6.2- Gestão de uma cascatas hídricas no curto prazo.....	158
6.3- Limites para as variáveis	159
6.4- Função objectivo.....	162
6.5- Equações de balanço	164
6.6- Construção dos ficheiros para a aplicação LSNNO	164
6.7- Dados para um caso exemplo	173
6.8- Manual do utilizador do LSNNO.....	174

1-BREVE HISTÓRIA SOBRE A PROGRAMAÇÃO LINEAR

A programação linear surgiu como um dos mais importantes ramos da programação matemática com uma vasta aplicação prática. Inovações da última metade do século passado fizeram com que os algoritmos de programação linear sejam eficientes e favoráveis para a resolução de um larga variedade de problemas envolvendo questões de decisão em vários domínios. Por exemplo: no planeamento da distribuição e produção de produtos, no planeamento de curto prazo em aproveitamento hidroeléctricos, nas decisões ligadas às políticas micro-económicas e macro-económicas de governação dos países, na utilização como subrotinas para suporte de tarefas específicas em códigos de programação não linear. Ainda, são aspectos positivos a considerar o facto da programação linear ser uma teoria de optimização significativamente completa, de existirem códigos para computadores podendo suportar problemas de muito grande dimensão.

O problema de optimizar uma função linear sujeita a restrições lineares teve as sua origem com os estudos de Fourier sobre sistemas lineares de inequações em 1826. No

entanto, só em 1939 Kantorovich faz notar a importância prática destes problemas, tendo criado um algoritmo para a sua solução. Num documento cujo objectivo era expor conceitos, Kantorovich apresentou exemplos para a aplicação da programação linear, sendo a ideia fundamental de cada exemplo a obtenção da maior produção possível com base numa utilização óptima dos recursos disponíveis. Um desses exemplos envolvia a distribuição de fluxos de carga (distribuídos através de veículos de transporte), usando diferentes rotas em redes rodoviárias de forma a satisfazer os requisitos e as restrições de capacidade das rotas, minimizando o consumo de combustível. Infelizmente, durante vários anos o trabalho de Kantorovich não só foi insuficientemente conhecido na Europa de Leste, mas também foi totalmente desconhecido na Europa Ocidental. Este amplo trabalho que aborda a discussão do fluxo de tráfego óptimo na antiga URSS foi efectuado por Kantorovich e Gavurin. O conhecimento sobre este trabalho chegou só ao ocidente depois de 1950.

O problema de otimizar uma função linear sujeita a restrições lineares tem o seu auge com George Dantzig na década de 1940, consultor de matemática do US Air Force Comptroller, e com o prémio Nobel da Economia George Stigler, que formulou o problema das dietas como um problema de mistura de componentes. Dantzig não só formula o problema de programação linear, mas também cria o Algoritmo do Simplex para a sua solução em 1947. Ainda em 1947, Koopmans mostra que a programação linear é um modelo apropriado para a análise da teoria económica clássica. Entretanto, nos EUA, Frank L. Hitchcock apresentou o que é hoje a formulação base do problema de transporte. Independentemente, o professor Koopmans formulou o mesmo problema em ligação com o seu trabalho efectuado na “Combined Shipping Adjustment Board”. Por isso, o problema de transporte é referido, na literatura científica, quer como

problema de transporte de Hitchcock, quer como problema de transporte de Hitchcock-Koopmans.

Em 1956 Alex Orden propôs a generalização do modelo de transporte em que eram permitidos pontos de transbordo de carga. Esta formulação é conhecida hoje como um problema de transbordo sem limite de capacidade. Ao mesmo tempo, o problema de fluxos máximos e o problema do fluxo do custo mínimo em rede foi formulado e investigado pela famosa equipa de Lester Ford e Delbert Fulkerson.

Entre 1950 a 1965 muita actividade foi dirigida para o desenvolvimento de algoritmos para modelos de programação linear em rede. Os algoritmos desenvolvidos podem ser classificados em duas classes, como se segue:

- 1) Especialização do Método Simplex;
- 2) Método Primal-Dual.

A especialização primal simplex começou com o trabalho de Dantzig e atingiu o apogeu com o documento de Ellis Johnson. Os fundamentos para o documento de Johnson podem ser encontrados basicamente em dois livros: um de Dantzig e outro de Charnes e Cooper.

O método primal-dual originado com o algoritmo húngaro de Harold Kuhn para o problema de atribuição é concluído com o algoritmo da condição de Delbert Fulkerson em 1961.

Muita da actividade de investigação envolveu uma eficiente implementação das técnicas básicas, a particularização dos códigos à programação linear em rede acrescentou vantagens no desempenho, comparativo, dos procedimentos para a resolução de problemas de programação linear em rede. O primeiro problema de tamanho considerável resolvido pelo Algoritmo do Simplex foi o das dietas de Stigler com nove equações e setenta e sete variáveis não negativas, gastando 120 Horas Homem nas calculadoras de secretária existentes na altura e menos de um segundo nos actuais computadores pessoais.

Em 1975, a Academia Real de Ciência atribuiu o prémio Nobel da Ciência em Economia a Kantorovich e Koopmans pelas a suas contribuições para a teoria da alocação de recursos, considerando a contribuição de Dantzig mais no âmbito matemático, não havendo prémio para o ramo científico da matemática, não atribuiu prémio a Dantzig. No entanto, Dantzig permanecerá para a história da construção da programação linear como um dos arquitectos fundamentais.

2- INTRODUÇÃO MATEMÁTICA

2.1- Notações e convenções

Ao longo deste texto as matrizes são indicadas por letras maiúsculas do alfabeto latino, a cheio. Os grafos ou conjuntos de nós e arcos dos grafos são indicados por letras maiúsculas do alfabeto latino. O símbolo \mathbf{I} indica a matriz identidade com dimensão apropriada para o contexto das operações matriciais onde está inserida. Uma matriz com todos os elementos nulos é indicada por $\mathbf{0}$. O elemento da linha i e da coluna j da matriz \mathbf{A} é indicado por \mathbf{A}_{ij} . O vector cujas coordenadas são os elementos da coluna j da matriz \mathbf{A} é indicado por $\mathbf{A}(j)$.

Determinadas constantes inteiras indicando valores máximos são indicadas por letras maiúsculas do alfabeto latino com uma barra por cima, como por exemplo \bar{K} . Letras minúsculas do alfabeto latino e letras do alfabeto grego, a cheio, são usadas para indicar vectores. O símbolo \mathbf{e}^i , indica um vector cuja coordenada i é igual a 1 e todas as outras coordenadas são iguais a 0, enquanto que o símbolo $\mathbf{1}$ indica um vector com todas as

coordenadas iguais a 1. A coordenada i do vector \mathbf{c} é indicada por c_i . O conjunto vazio é indicado por ϕ . Será adoptado o ponto de vista de que os vectores pertencentes a um espaço vectorial (com dimensão igual a n) não são necessariamente nem vectores linha ($1 \times n$ matrizes) nem vectores coluna ($n \times 1$ matrizes), sendo considerados de modo que as dimensões fiquem compatíveis com a multiplicação, isto é, um vector é tratado como um vector linha quando aparece à esquerda de outro vector ou de uma matriz e como um vector coluna quando aparece à direita. Assim, o produto interno de dois vectores \mathbf{a} e \mathbf{b} é indicado simplesmente por \mathbf{ab} . Portanto, no contexto matricial e da forma que for mais conveniente, um vector será inserido como uma linha ou como uma coluna. Um “Superscripts” usualmente indica membros de uma classe em vez de expoentes. Sigma é usado para indicar a função sinal de um número definida por:

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

Serão ainda, empregues outras notações matemáticas, que serão apresentadas ao longo do texto quando forem requeridas. No que se refere aos conceitos de grafo e de rede, que surgem em numerosos ramos da ciência, onde têm existido contribuições próprias no contexto destes ramos da ciência para a evolução do estudo de problemas de fluxo em redes, que é a forma mais simples de modelar matematicamente muitos problemas desses ramos, utilizando o conceito de rede. No entanto, as perspectivas de cada ramo são não convergentes, sendo facto que a literatura da teoria de grafos e de redes não possui uniformização, existindo diversidade variedade de convenções e notação. Por isso, neste ponto será feita informalmente uma descrição mais exhaustiva do conceito de grafo dirigido para fixar a linguagem que será usada neste texto. Um grafo é constituída por dois conjuntos finitos. O conjunto dos nós, vértices ou nodos, indicado por N . O conjunto das arcs, arestas ou ramos, indicado por A . Assim, um grafo G é indicado

como um par ordenado $G = [N, A]$ e é de ordem n se $\#A=n$. Cada arco é representada por um par ordenado de nós (i, j) , com $i \neq j$ e $i, j \in N$ em que i é o nó de onde provém o arco o nó da cauda e j e o nó para onde se dirige o arco ou o nó cabeça; diz-se que arco (i, j) sai do nó i e chega ao nó j . Portanto, $A \subseteq N \times N$. O arco (i, j) é incidente nos vértices i e j . O arco (i, j) é um arco de saída do nó i e um arco de chegada do nó j . O nó j diz-se adjacente ao nó i se existir o arco (i, j) . Dois arcos são adjacentes se forem ambos incidentes no mesmo nó. A lista de adjacência dos arcos de um nó i , indicada por $A(i)$, é o conjunto de arcos que saem do nó i ; isto é, $A(i) = \{(i, j) \in A : j \in N\}$. A lista de adjacência de nós de um nó i , indicada por $V(i)$, é o conjunto de nós adjacentes ao nó i , isto é, $V(i) = \{j \in N : (i, j) \in A\}$. Claro que o $\#A(i)$ é igual ao $\#V(i)$ e dado pelo o número de arcos que saem do nó i . Um grafo diz-se completo se entre quaisquer dois vértices existir uma arco. A densidade de um grafo é a razão entre o número de arcos de grafo e o número de arcos do grafo completo com o mesma número de nós. Dado um grafo $G = (N, A)$, $G' = (N', A')$ é um subgrafo de G se $N' \subseteq N$ e $A' \subseteq A$.

Uma rede é um grafo cujos arcos têm rótulas, i.e, existem valores numéricos associados, por exemplo: custos unitário de transporte num arco, capacidades máxima de um arco. Assim, rede é um terno ordenado $R = [N, A, L]$, sendo $[N, A]$ um grafo e L é o conjunto dos valores numéricos associados aos arcos.

2.2- Programação linear em redes

O nome programação em redes faz lembra de imediato a rede eléctrica, a rede telefónica, a rede rodoviária, a rede ferroviária etc. Em todas estas redes existe em trânsito algo, a electricidade na rede eléctrica, a mensagem na rede telefónica, o ser

humano, bens ou resíduos nas redes rodoviária e ferroviária. Este trânsito tem que ser decidido e claro deve ser o mais racional e eficiente possível.

Para que a decisão seja eficiente em redes de tamanho considerável é necessário recorrer à modelação matemática da estrutura da rede para o suporte da decisão racional e ao desenvolvimento de algoritmos para o tratamento de problemas em redes. Uma rede envolve o conceito de grafo, que é uma representação gráfica de dados e da ligação existente entre esse dados. A representação gráfica tem vantagens na construção do modelo matemático para a resolução do problema. Muitas áreas do conhecimento, como por exemplo, a matemática, as engenharias em geral e a gestão usam a teoria de grafos para descrever diversas situações dentro do seu âmbito

Uma rede é uma estrutura caracterizada por três tipos de objectos, constituintes da rede é que são os arcos, os nós e as etiquetas associadas com os nós ou com os arcos. Os arcos podem ser considerados como abstracções de meios unidireccionais, como por exemplo meios físicos para o transporte de mercadorias. Os nós podem ser considerados como abstracções interpretados como locais de injeção de fluxo ou terminais de recepção do fluxo. As etiquetas são dados associados com os objectos arcos ou nós. Os nós são geralmente ligados por arcos, podendo corresponder com a realidade de um qualquer meio físico de transporte. Assim, os arcos podem representar linhas eléctricas, ruas ou estradas numa rede urbana de transportes, tubagens numa rede de distribuição de água, linhas telefónicas numa rede de comunicações, etc. No estudo que será feito as considerações serão limitadas a redes com um número finito de nós e arcos e sem laços. No entanto, se existir a necessidade de considerar laços eles podem ser introduzidos considerando o laço substituído por dois arcos e um nó entre eles. Para

uma rede com \bar{I} nós e \bar{J} arcos, será admitida uma ordem para os nós e para os arcos, através de uma qualquer enumeração para os nós de $1, \dots, \bar{I}$, e para os arcos de $1, \dots, \bar{J}$.

Uma rede é ilustrada por um grafo orientado sobre o qual se escrevem as etiquetas. Os nós ilustrados por círculos, e os arcos ilustrados por linhas orientadas ligando dois nós. A orientação do arco é indicada por uma seta colocada sobre o respectivo o arco. A Figura 2.1, ilustra a forma como se representa o grafo usando uma representação sagital.

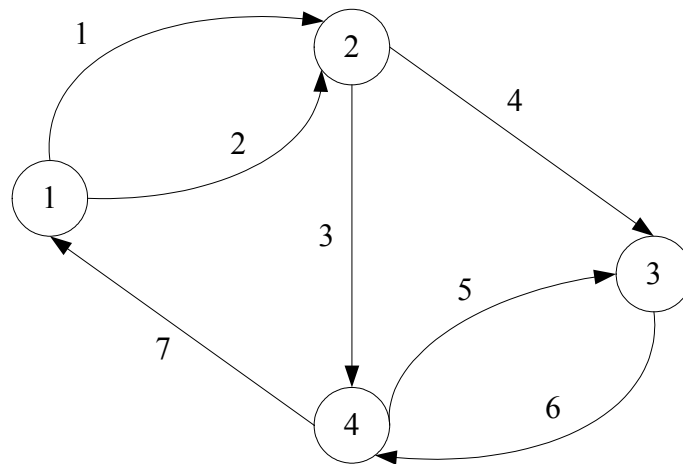


Figura 2.1 – Ilustração de uma rede.

Observe que é possível mais do que um arco a ligar os mesmos dois nós com a mesma orientação. Por exemplo na Figura 2.1, ambos os arcos 1 e 2 saem do nó 1 e entram ambos no nó 2. A rede em termos matemáticos pode ser descrita por uma matriz cuja dimensão é o número de nós pelo número de arcos, $\bar{I} \times \bar{J}$, sendo os seus elementos definidos da seguinte forma:

$$A_{ij} = \begin{cases} +1 & \text{se do nó } i \text{ sai o arco } j; \\ -1 & \text{se no nó } i \text{ entra o arco } j; \\ 0 & \text{caso contrário.} \end{cases}$$

A matriz A acima definida é chamada a matriz de incidência dos nós nos arcos. A matriz de incidência dos nós nos arcos correspondente à rede da Figura 2.1 é a seguinte:

$$\text{nós} \left\{ \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} \right. \left[\begin{array}{ccccccc} \text{arcos} \\ \begin{array}{c} 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \end{array} \\ \begin{array}{ccccccc} 1 & & & & & & -1 \\ -1 & -1 & 1 & 1 & & & \\ & & & -1 & -1 & 1 & \\ & & -1 & & 1 & -1 & 1 \end{array} \end{array} \right]$$

Como cada arco é só incidente em dois nós uma propriedade que caracteriza esta matriz é que cada coluna tem exactamente dois elementos não nulos, um igual a $+1$, no nó de onde saí o fluxo para o arco, e outro igual a -1 , no nó onde entra o fluxo do arco. Qualquer matriz, independentemente da sua origem, que tenha esta propriedade será chamada matriz de incidência dos nós nos arcos, visto que, pode ser representada por uma rede.

A cada arco j da rede está associado uma variável de decisão x_j , definindo a quantidade de fluxo através do arco j , sendo o vector, cuja dimensão é \bar{J} e as coordenadas são os fluxos nos arcos indicado por \mathbf{x} . O custo unitário de transporte do fluxo do arco j é indicada por c_j , tendo o vector \mathbf{c} dos custos unitários dimensão \bar{J} e coordenadas dadas pelos custos unitários. A capacidade do arco j em termos de fluxo máximo é indicada por u_j , tendo vector dos fluxos máximos \mathbf{u} dimensão \bar{J} e coordenadas dadas pelos os fluxos máximos nos arcos. A cada nó está associado um requisito do nó, o requisito do nó i é indicado por r_i , tendo o vector dos requisitos \mathbf{r} dimensão \bar{I} e coordenadas dadas pelos requisitos de cada nó. Se $r_i > 0$, o nó i é de fornecimento ou oferta com uma oferta igual a r_i , i.e., é injectado fluxo na rede por este nó. Se $r_i < 0$ o nó i é de consumo ou procura com um consumo igual a $|r_i|$, i.e., é retirado fluxo da rede por este nó. Se $r_i = 0$,

o nó i é um nó de transbordo, i.e., não é injectado nem retirado fluxo da rede neste nó. Em formulação matemática, o problema da minimização do custo do fluxo da rede é representado da seguinte forma

$$\left. \begin{array}{l} \min \quad \mathbf{c} \mathbf{x} \\ \text{s.a.} \\ \mathbf{A} \mathbf{x} = \mathbf{r} \\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \end{array} \right\} \begin{array}{l} \text{PR} \\ \text{(NP)} \end{array}$$

Onde \mathbf{A} é a matriz de incidência dos nós nos arcos, e “s.a.” é sigla abreviatura de “sujeito a”.

Diversos casos particulares da programação linear em rede, PR Programação em Redes (NP Network Programming), mereceram estudos particulares que originaram códigos mais eficientes para a sua resolução em comparação com a utilização de códigos mais genéricos. Uma classificação desses casos é apresentada de seguida.

1. *Problema de transbordo com capacidade não limitada.*

Este problema é uma particularização do programação linear em rede, na qual $u_j = \infty$ para todo j , i.e., não existe uma limitação de fluxo máximo dos arcos.

2. *Problema de transporte com capacidade limitada.*

Este problema é uma particularização do programação linear em rede, na qual um nó ou é de oferta ou é de procura. Ainda, os arcos são dirigidos dos nós de oferta para os nós de procura. A Figura 2.2 ilustra um problema deste tipo em que existem três nós de oferta e cinco de procura.

3. *Problema de Transporte*. Este problema é um caso especial problema de transporte com capacidade limitada no qual $u_j = \infty$ para todo j . Portanto a Figura 2.2 também serve para ilustração do problema de transporte.

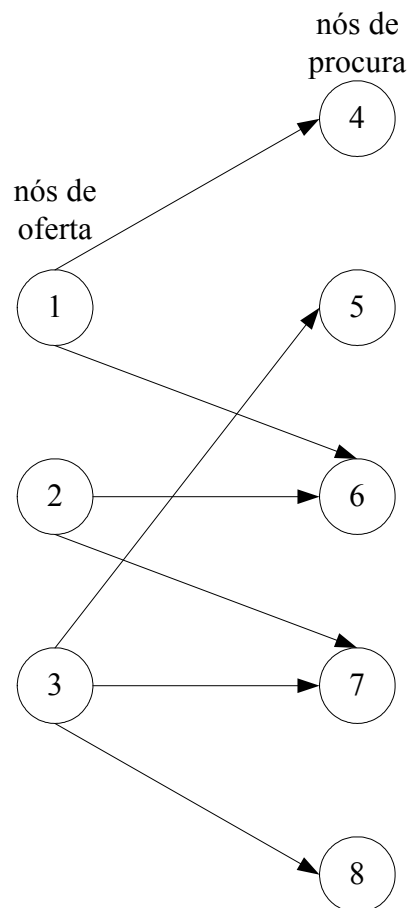


Figura 2.2 – Problema de transporte.

4. *Problema de afectação*.

Este problema consiste em associar entidades uma a uma da forma mais vantajosa. As entidades podem ser tarefas, indivíduos, máquinas, etc. Exemplos: associar pessoas a funções, carros a pessoas, funções a máquinas. Este problema é um caso especial do problema de transporte no qual todos os requisitos são iguais a ± 1 , i.e., $|r_i|=1$ para $i=1, \dots, \bar{I}$.

5. Problema do fluxo máximo.

Este problema consiste em encontrar uma solução admissível que envie a máxima quantidade de fluxo de um nó origem para um nó destino, satisfazendo as restrições de capacidade dos arcos da rede. Por exemplo: determinar o número máximo de pessoas que podem ser transportadas por uma rede aérea de transportes entre duas localizações, ou o volume máximo de petróleo numa rede de oleodutos, ou de energia eléctrica numa rede eléctrica. Em geral são dados dois nós de uma rede: s nó de oferta e t nó procura. Os arcos da rede têm capacidade u_j para $j = 1, \dots, \bar{J}$, o problema consiste em encontrar o fluxo máximo, satisfazendo as restrições de capacidade $\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$, que pode ser enviado através da rede do nó s para o nó t . Este problema é um caso especial do problema PR no qual $\mathbf{r} = \mathbf{0}$ e só um arco tem custo não nulo. Para exemplificar, suponha que se pretende determinar o fluxo máximo entre os nós 1 e 4 da rede da Figura 2.1.

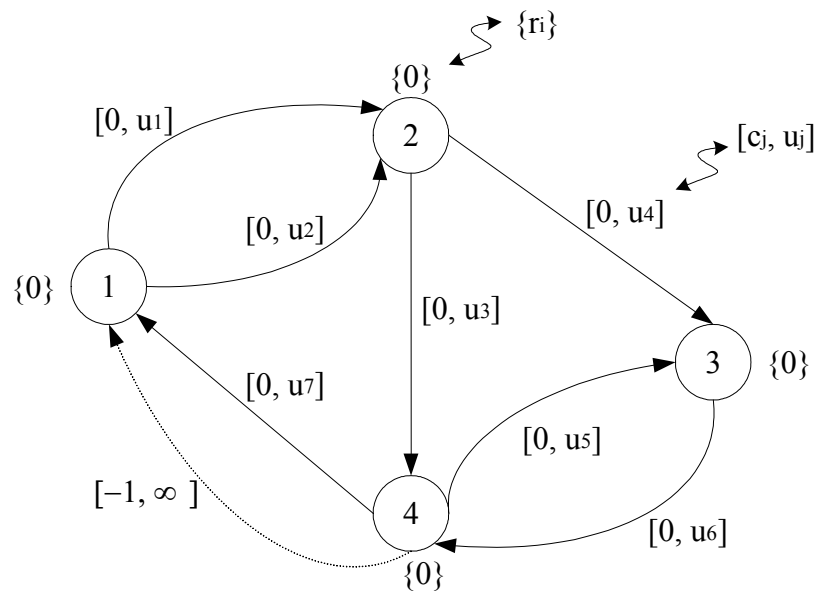


Figura 2.3 – Problema de maximização do fluxo entre nós.

Aumenta-se a rede com um arco fictício de retorno, número de ordenação do arco 8, do nó 4 para o nó 1, sendo $u_8 = \infty$ e $c_8 = -1$ e sendo $r_i = 0$ para $i = 1, \dots, 4$ e $c_j = 0$ para $j = 1, \dots, 7$. A rede para este problema é ilustrada na Figura 2.3.

6. Problema do caminho mais curto.

Este problema consiste em encontrar o caminho mais curto entre dois nós s e t de uma dada rede cujos arcos têm custos $c_j \geq 0$ para $j = 1, \dots, \bar{J}$. Estes custos podem identificar-se, com o valor da distância a percorrer quando se opta pelo trajecto determinado pelo respectivo arco. Este caso corresponde a fazer $r_s=1$, $r_t=-1$ e os outros requisitos são iguais a zero. Supondo que se pretende encontrar o caminho mais curto entre os nós 1 e 3 na rede da Figura 2.1, o problema programação é então descrito pela rede ilustrada na Figura 2.4.

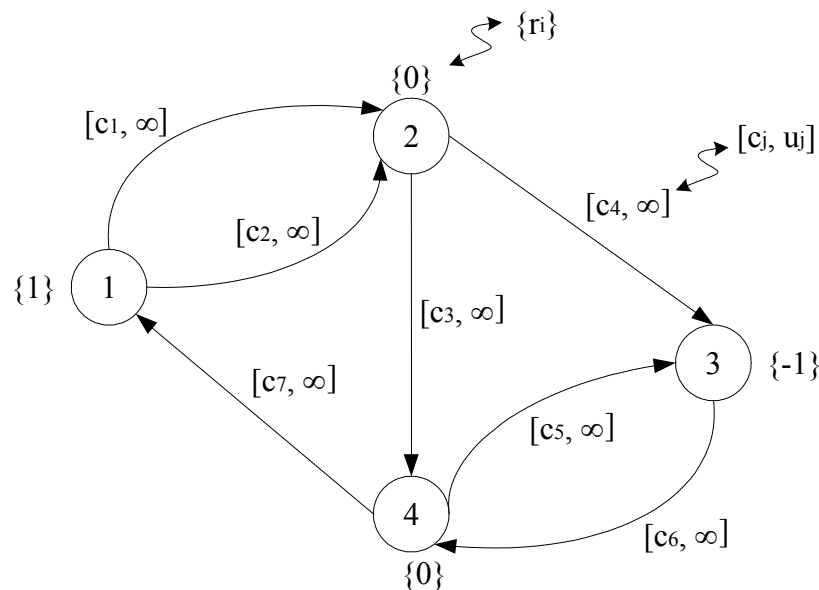


Figura 2.4 – Problema do caminho mais curto.

Na solução óptima a unidade de fluxo irá percorrer o caminho mais curto entre os nós s e t . Caso a unidade de fluxo se fraccionar em quantidades iguais que fluirão

através de mais do que um arco, existirão caminhos alternativos cujo percurso é igual à distância mais curta.

2.3- Exemplos de modelos em programação linear

A programação linear tem uma parte que é arte e outra que é ciência. Enquanto que o processo de construção da estrutura proveniente de enunciado de um problema real, permitindo a análise do problema real, pode ser encarado como uma arte, existindo várias maneiras de proceder a arte guia na direcção da mais conveniente. O desenvolvimento dos algoritmos para resolução dos problemas de programação linear é uma ciência. Exemplos práticos que ilustram a construção de um problema de programação linear são apresentados em seguida.

2.3.1- Produção - distribuição da polímeros

A Polímeros é uma empresa de polímeros europeia. O director de produção decide o nível de produção de cada uma das quatro fábricas na Europa. O plano de produção-distribuição é feito com dois meses de antecipação. O director de produção está a trabalhar nos pedidos para o mês de Julho. Existem diversos clientes os quais foram agrupados em cinco regiões. Existe aqui já uma primeira decisão que foi tomada anteriormente que é a formação de regiões de clientes, ou seja, construir pontos de procura que aglomeram clientes. Claro que ao nível de transporte os clientes localizados numa cidade são potencialmente aglomerados num ponto, de qualquer modo considere que essa decisão já foi tomada e que o director de produção tem que respeitar o que se entendeu pontos de procura que aglomeram clientes.

A procura de polímero para o mês de Julho para cada uma das regiões é indicada no Quadro 2.1. As fábricas da Poliremos estão localizadas em Dresden na Alemanha, Lie na França, Milão na Itália e Aveiro em Portugal.

Quadro 2.1 – Procura para Julho

Região de Vendas	Pedidos para Julho (Ton.)
1	2840
2	2800
3	2600
4	2820
5	2750

Dadas as diferenças existentes nas várias fábricas, as capacidades e os custos de produção são distintos de fábrica para fábrica e são indicadas no Quadro 2.2.

Quadro 2.2 – Capacidade de produção e custo para Julho

Fábrica/País	Capacidade (Ton.)	Custo por Tonelada (€)
Dresden/Alemanha	4000	2100
Lille/França	4500	2000
Milão/Itália	2700	1600
Aveiro/Portugal	3000	1700

O custo de transporte por tonelada e a respectiva capacidade de transporte, entre cada uma das quatro fábricas e as cinco regiões de venda, são indicados no Quadro 2.3.

Por exemplo: o custo de produção de uma tonelada de polímero na fábrica de Aveiro é de 1700€; o custo transporte de uma tonelada de polímero fabricado em Aveiro para ser vendido na região 3 é de 400€. Então o custo total de produção e transporte de uma tonelada de polímero de Aveiro para ser distribuído na região 3 de $1700+400=2100$ €.

Quadro 2.3 – Custo e capacidade de transporte para Julho

Região de Vendas	Fábricas							
	Dresden		Lille		Milão		Aveiro	
	Custo (€)	Capac. (Ton.)	Custo (€)	Capac. (Ton.)	Custo (€)	Capac. (Ton.)	Custo (€)	Capac. (Ton.)
1	800	0	200	3000	500	1000	0	0
2	600	2000	400	2000	200	2000	400	1000
3	300	2000	500	1000	200	1000	400	1000
4	200	2000	900	200	200	1000	300	1000
5	400	500	0	0	400	300	100	2000

No Quadro 2.3 os valores nulos correspondem a situação a onde não é possível meios de transporte para o mês de Julho, sempre que o máximo fluxo de um arco seja nulo o custo unitário desse arco pode tomar qualquer valor que não vai influenciar a decisão.

O problema do director de produção é fazer a programação da produção e da distribuição de cada uma das quatro fábricas para as regiões de procura dos polímeros, de modo a que a soma do custo de produção mais transporte seja mínima. Para a resolução deste problema o director de produção pode por abstracção considerar que cada uma das quatro fábricas é um nó por onde se injecta fluxo na rede.

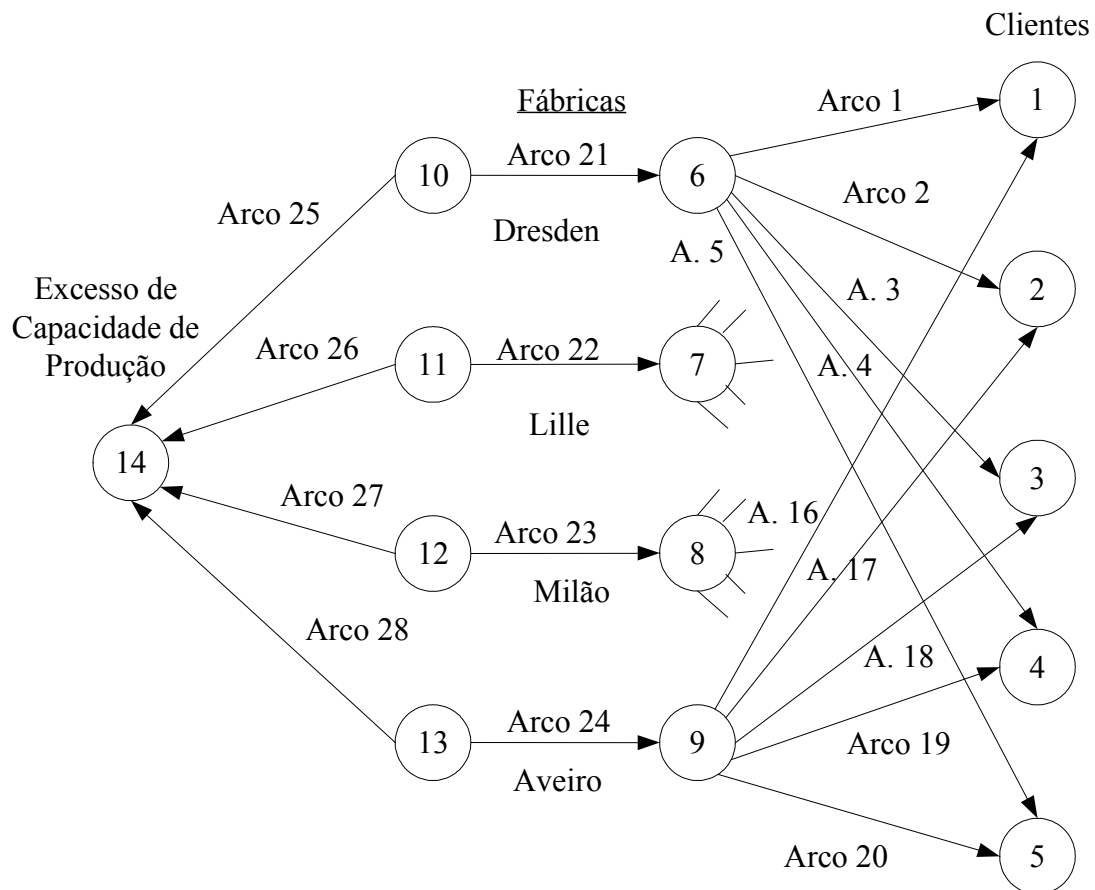


Figura 2.5 – Produção e distribuição na Polímeros.

No entanto, como quer obter o plano de produção e transporte tem que considerar que em cada fábrica existe um percurso que corresponde à produção dos polímeros. Portanto, este percurso cujo custo unitário é o custo de produção da tonelada de polímero será descrito por um arco que liga um nó que pode ser considerado como o início de produção com o nó que pode ser considerado como o fim de produção da fábrica. A rede será constituída como se indica na Figura 2.5, sendo os nós número: 1 a 5, os nós clientes, as regiões que o director de produção tem que considerar; 6 a 9 os nós fim de produção para fábricas; 10 a 13 os nós de início de produção para fábricas. No entanto, estes últimos nós tem um significado que deriva do facto dos seus requisitos serem iguais à capacidade máxima de produção das respectivas fábricas, como está indicado no Quadro 2.4.

Quadro 2.4 – Requisitos dos nós (Figura 2.5)

N.º do Nó	Requisito	N.º do Nó	Requisito
1	-2840	8	0
2	-2800	9	0
3	-2600	10	4000
4	-2820	11	4500
5	-2750	12	2700
6	0	13	3000
7	0	14	-390

Quadro 2.5 – Especificações dos arcos

N.º do Arco	Nó “De”	Nó “Para”	Custo Unit.	Capac. Arco	N.º do Arco	Nó “De”	Nó “Para”	Custo Unit.	Capac. Arco
1	6	1	800	0	15	8	5	400	300
2	6	2	600	2000	16	9	1	0	0
3	6	3	300	2000	17	9	2	400	1000
4	6	4	200	2000	18	9	3	400	1000
5	6	5	400	500	19	9	4	300	1000
6	7	1	200	3000	20	9	5	100	2000
7	7	2	400	2000	21	10	6	2100	4000
8	7	3	500	1000	22	11	7	2000	4500
9	7	4	900	200	23	12	8	1600	2700
10	7	5	0	0	24	13	9	1700	3000
11	8	1	500	1000	25	10	14	0	∞
12	8	2	200	2000	26	11	14	0	∞
13	8	3	200	1000	27	12	14	0	∞
14	8	4	200	1000	28	13	14	0	∞

No nó início de produção é injectado o fluxo que será igual à capacidade máxima de produção da fábrica, mas como nem toda a capacidade necessitava de ser utilizada para o mês de Julho surge um outro nó para onde será localizado o excesso de capacidade de produção, sendo o custo de transporte nos arcos que ligam este nó nulo.

Os dados e a estrutura para o sistema de produção/distribuição é apresentado no Quadro 2.5. Note que o excesso de capacidade de produção é concretizado na representação do grafo por o nó extra número 14 que liga as quatro fábricas, através de arcos de custo zero e capacidade de transporte ilimitada. O director de produção pode também considerar que a capacidade de transporte nestes arcos é limitada e igual a capacidade de produção das respectivas fábricas ou até considerar que capacidade de transporte é qualquer número superior ao excesso de capacidade de produção.

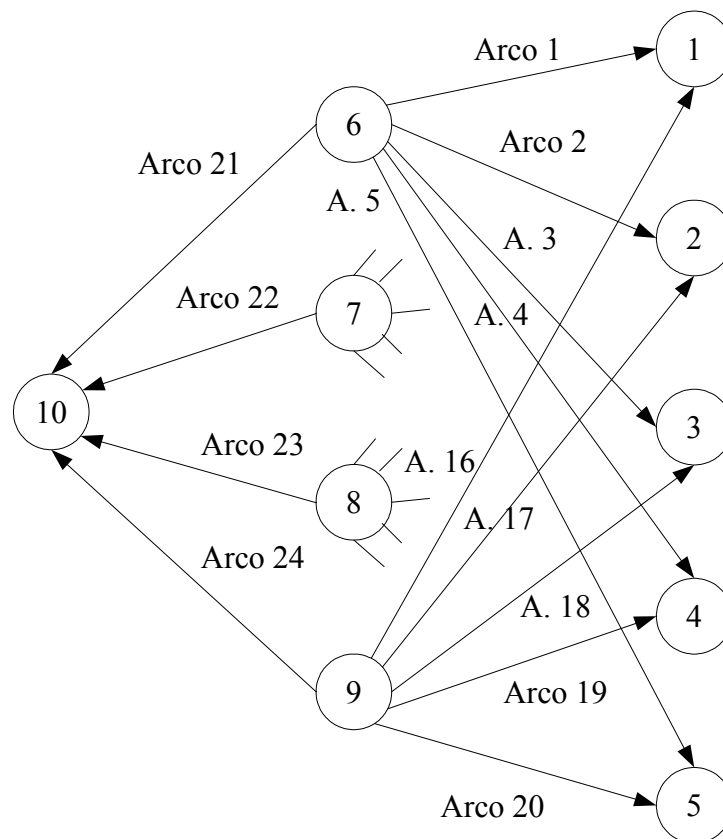


Figura 2.6 – Produção e distribuição na Poliremos.

Na Figura 2.6 e nos Quadros 2.6 e 2.7 é apresentada outra opção possível para o director de produção ilustrar a rede deste problema. Esta variedade de ilustração de um problema corresponde a uma variedade na sua formulação que de facto pode ser encarada como

uma arte, visto que é não sistemática deixando alguma margem de liberdade, que deve ser convenientemente explorada.

Nesta última opção o director de produção considera só um nó para cada fábrica e aglomera o custo de produção da fábrica com o custo de transporte para cada uma das regiões, criando arcos com custo unitário igual ao custo de produção mais o custo de transporte por tonelada de polímero de cada fábrica para cada uma das regiões, correspondendo a uma simplificação na formulação do problema. No entanto, perde em descrição do processo de produção e distribuição, visto que, agora só existe um arco na rede que está associados aos dois processos. O valor do fluxo máximo num arco que substitui outros é o mínimo entre os fluxos máximos dos arcos que foram substituídos por esse arco.

Embora nada tenha sido dito o nó número 14 na primeira opção ou o nó número 10 na segunda opção pode efectivamente ser também representativos de um nó de armazém, i.e., de facto se o director de produção não têm a liberdade de definir o nível de produção de cada fábrica, visto que, ambas têm que produzir a capacidade máxima, então a produção não requerida pelos clientes é dirigida para um armazém.

Quadro 2.6 – Requisitos dos nós (Figura 2.6)

N.º do Nó	Requisito	N.º do Nó	Requisito
1	-2840	6	4000
2	-2800	7	4500
3	-2600	8	2700
4	-2820	9	3000
5	-2750	10	-390

Quadro 2.7 – Especificações dos arcos (Figura 2.6)

N.º do Arco	Nó “De”	Nó “Para”	Custo Unit.	Capac. Arco	N.º do Arco	Nó “De”	Nó “Para”	Custo Unit.	Capac. Arco
1	6	1	2900	0	13	8	3	1800	1000
2	6	2	2700	2000	14	8	4	1800	1000
3	6	3	2400	2000	15	8	5	2000	300
4	6	4	2300	2000	16	9	1	0	0
5	6	5	2500	500	17	9	2	2100	1000
6	7	1	2200	3000	18	9	3	2100	1000
7	7	2	2400	2000	19	9	4	2000	1000
8	7	3	2500	1000	20	9	5	1800	2000
9	7	4	2900	200	21	6	10	0	∞
10	7	5	0	0	22	7	10	0	∞
11	8	1	2100	1000	23	8	10	0	∞
12	8	2	1800	2000	24	9	10	0	∞

2.3.2- Problema das dietas

Os alimentos $1, \dots, i, \dots, m$ com custo unitário c_i , disponibilidade d_i e percentagem a_{ij} dos componentes $j = 1, 2, \dots, n$, tem que originam a quantidade “s” de ração alimentar, tendo as percentagem do componente j que ser compreendida no intervalo

$$[\underline{p}_j, \bar{p}_j].$$

Determinar a quantidade de cada produto x_i na mistura, que corresponde ao menor custo total da ração.

O problema de programação linear para obter o menor custo tem como função objectivo o custo total da mistura, sendo as restrições a satisfazer: a quantidade “s” de ração; percentagem do componente j compreendidas no intervalo respectivo; e finalmente a

limitação da disponibilidade dos alimentos. O problema de programação linear para o problema das dietas é então descrito como

$$\begin{aligned} & \min \sum_{i=1}^m c_i x_i \\ & \text{s.a} \\ & \sum_{i=1}^m x_i = s \\ & s p_{-j} \leq \sum_{i=1}^m a_{ij} x_i \leq s \bar{p}_j \quad j = 1, 2, \dots, n \\ & 0 \leq x_i \leq d_i \quad i = 1, 2, \dots, m. \end{aligned}$$

Outro problema semelhante a este é problema de encontrar a ração com maior quantidade para um dos elementos, mas cujo custo seja inferior a um determinado valor. Fica ou cuidado do leitor escrever a formulação para este último problema.

2.3.3- Financiamento de projectos

Os projectos $1, \dots, i, \dots, n$ têm valor actual líquido v_i e requerem respectivamente para o financiamento o capital a_i , podendo ser financiados totalmente ao em parte. O capital total disponível para financiar os projectos é b , pretende-se determinar quanto deve ser financiado cada um dos projectos de forma a maximizar o valor actual líquido do investimento de capital investido nos projectos. O problema de programação linear para obter o financiamento óptimo é

$$\begin{aligned} & \max \sum_{i=1}^n v_i x_i \\ & \text{s.a} \\ & \sum_{i=1}^n a_i x_i \leq b \\ & 0 \leq x_i \leq 1. \end{aligned}$$

Caso o financiamento seja na globalidade, a variável de decisão tem que ser inteira.

2.3.4- Planeamento em aproveitamentos hidroeléctricos, curto prazo, modelo LP

Numa aproveitamento hidroeléctrico a optimização da gestão dos recursos consiste em decidir com racionalidade a gestão da água disponível, sujeita a condicionantes que são impostas quer pelo equipamento, condicionantes técnicas, quer por razões ecológicas e de segurança, condicionantes políticas. As variáveis ligadas aos trânsitos de volume de águas estão relacionadas pelas equações de balanço da água, complicando o problema. Consequentemente, para atingir racionalidade na gestão é necessário a consideração de variáveis e restrições que ultrapassam a capacidade humana mental de decidir sem um suporte racional auxiliar para descrever o comportamento do aproveitamento hidroeléctrico. Assim, o recurso à simulação computacional do modelo matemático do aproveitamento hidroeléctrico a gerir e a técnicas de optimização, permite gerar um sistema de informação para o suporte á tomada das decisões. Considere a figura seguinte, que representa um aproveitamento hidroeléctrico em cascata genérico, com as albufeiras $i = 1, 2, \dots, n$ e admita que existem centrais hidroeléctricas entre albufeiras consecutivas até ao mar, considerando que o mar é o fim da cascata.

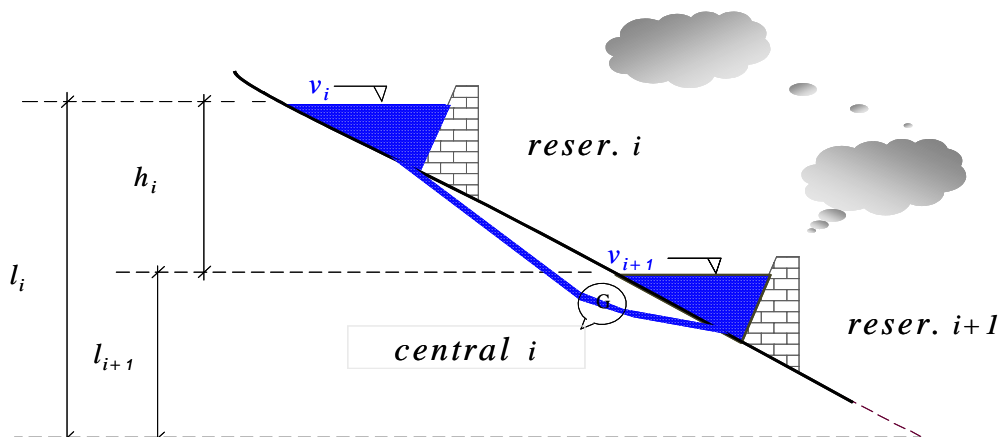


Figura 2.7 – Albufeiras e central hidroeléctrico para a cascata.

Numa cascata hídrica, a gestão de curto prazo do aproveitamento hidroeléctrico decide o volume de água armazenado nas albufeiras e o turbinado ou descarregado em cada

período, tipicamente de uma hora, durante o horizonte temporal, tipicamente de uma dia até uma semana, correspondendo este domínio ao ciclo diário até ao semanal dos diagramas de carga. As equações de balanço de água numa albufeira e em cada período determinam um nó da rede, considerando uma albufeira i no período k a equação de balanço de água é

$$v_i^{k-1} + r_i^k + t_{i-1}^k + s_{i-1}^k = v_i^k + t_i^k + s_i^k$$

Caso k seja a segunda hora, a equação determina o nó indicado na Figura 2.8.

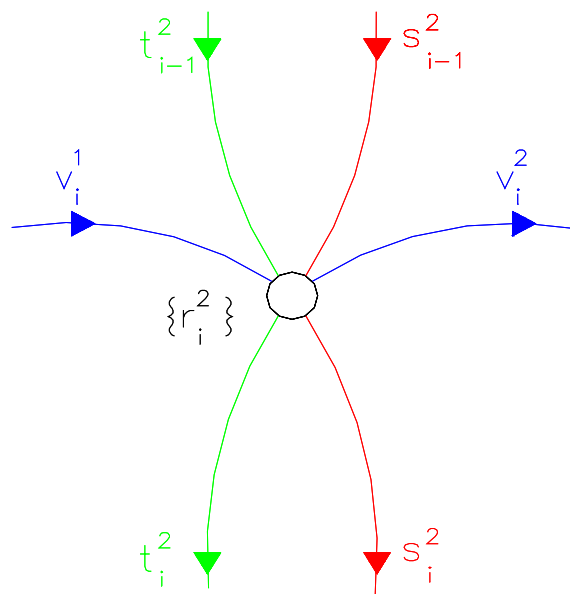


Figura 2.8 – Nó para a equação do balanço da água na segunda hora.

O lado esquerdo da igualdade é o valor que toma o volume de água disponível na albufeira i para ser gerido no período k , igual ao que transita anterior, $k-1$, mais o requisito r (afluência natural ou um valor pré-especificado) à albufeira i no período k e mais o volume que provém da albufeira de montante $i-1$, turbinado ou descarregado no período k , i.e., considerando o tempo de transito entre albufeiras para o volume turbinado ou descarregado nulo. O lado direito da igualdade é o volume de água disponível na albufeira i para ser gerido no período k igual ao volume que transita para a hora $k+1$ mais o turbinado e o descarregado para a albufeira de jusante no período k .

Na Figura 2.9, está representada a rede para um aproveitamento hídrico em cascata com três albufeiras, que será considerado neste exemplo. Com o fim de ilustrar o exemplo será considerado um horizonte só com três horas. Esta rede é uma expansão temporal da arborescência relativa à primeira hora.

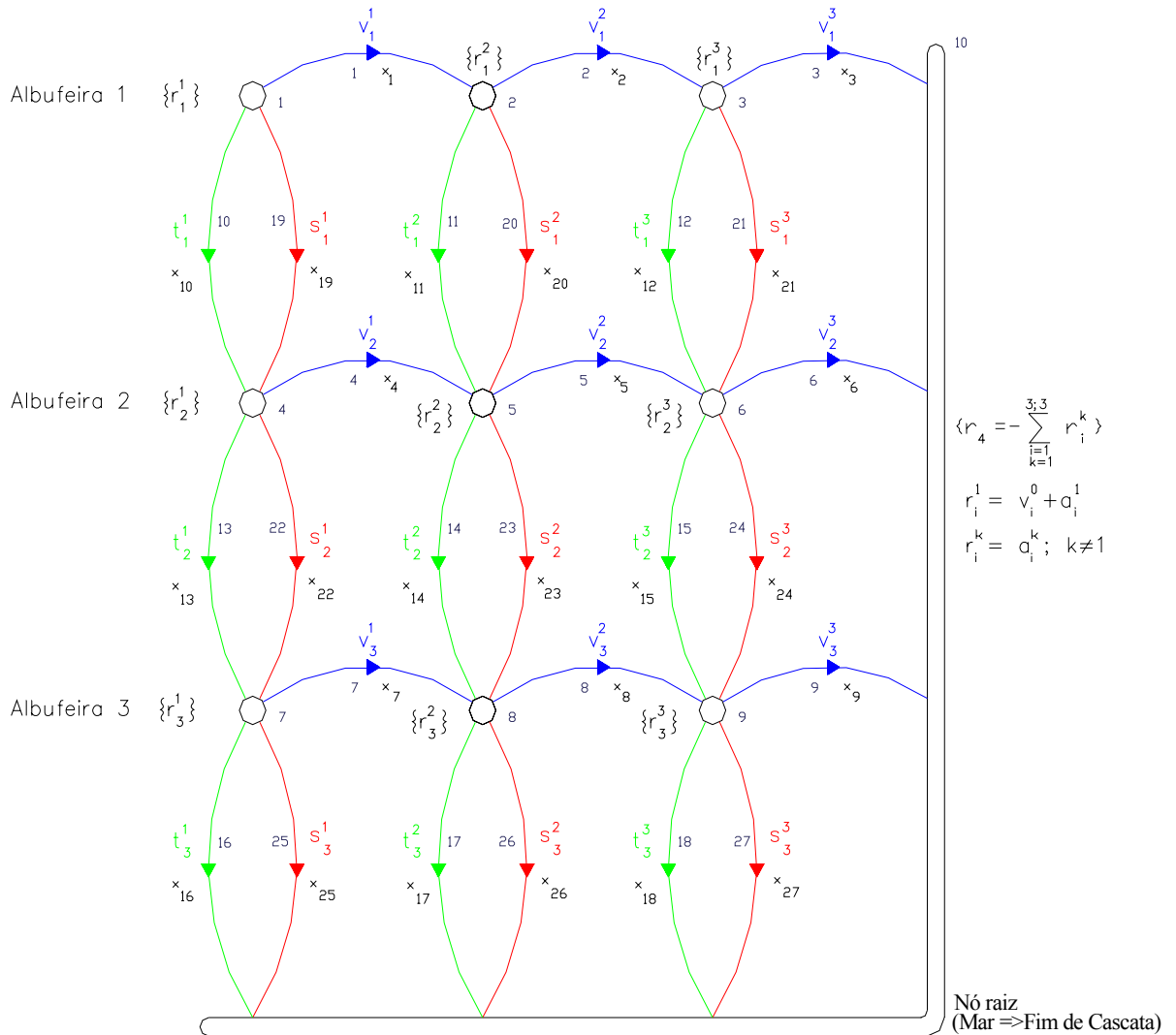


Figura 2.9 – Expansão temporal da arborescência da primeira hora até a terceira hora.

A função objectivo é determinada pela soma do lucro obtido com a exploração de cada central hidroeléctrica. Para obter um modelo linear será considerada como hipótese que a queda útil permanece constante durante o horizonte temporal. Assim, a expressão do lucro para a central i no período k é escrita como

$$\text{lucro}_i^k = \lambda_k p_i^k \quad \text{com} \quad p_i^k = \rho_i t_i^k \quad (2)$$

sendo λ_k o valor económico da energia eléctrica, preço unitário, no período k . Para obviar a que no fim do horizonte o volume de água armazenado seja o mínimo é normal impor cotas finais dos reservatórios, ditas cotas objectivo, que corresponde a introduzir no problema de programação linear restrições de igualdade para impor os volumes finais de água armazenados nos reservatórios.

$$\max \sum_{k=1}^K \lambda_k \sum_{i=1}^I p_i^k(t_i^k)$$

s.a

para $i = 1, 2, \dots, I$ $k = 1, 2, \dots, K$

$$v_i^{k-1} + r_i^k + t_{i-1}^k + s_{i-1}^k = v_i^k + t_i^k + s_i^k$$

$$\underline{v}_i \leq v_i^k \leq \bar{v}_i$$

$$0 \leq t_i^k \leq \bar{t}_i$$

$$0 \leq s_i^k \leq \bar{s}_i$$

$$v_i^K = \bar{v}_i^f$$

Este problema de programação linear é um problema em redes cuja formulação é do tipo

$$\left. \begin{array}{l} \max \quad \boldsymbol{\pi} \mathbf{x} \\ \text{s.a} \quad \mathbf{A} \mathbf{x} = \mathbf{r} \\ \quad \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \end{array} \right\} \text{problema para a gestão da cascata no curto prazo}$$

O vector \mathbf{x} dos fluxos é um vector cujas coordenadas são os volumes de água quer em trânsito, turbinado ou descarregado, quer os volumes de água armazenados nas albufeiras. A função objectivo é uma função linear do volume de água turbinado, visto

que, a produção de energia eléctrica na central hidroeléctrica i é proporcional ao caudal turbinado nessa central no período k .

Considere o seguinte caso: uma empresa gere um aproveitamento hidroeléctrico. O director de produção decide o nível de produção de cada uma das três centrais de uma cascata com três albufeiras que termina no mar. O director de produção está a planear tomar as melhores decisões para um horizonte temporal com três horas. O volume máximo, mínimo, inicial e final requerido em cada albufeira, o factor de eficiência da conversão, o máximo caudal e as afluências às albufeiras estimadas para as respectivas centrais e os preços para a energia eléctrica são os indicados no quadro seguinte.

Quadro 2.7 – Dados para do aproveitamento hidroeléctrico

volume [hm ³]	reser. 1	reser. 2	reser. 3	
\bar{v}	9.9	13.5	26.4	
v^o	7.9	10.8	21.1	
v^f	7.9	10.8	22.1	
\underline{v}	4	5	10	
ρ [MWh/hm ³]	121.03	164.14	181.09	
\bar{t} [hm ³ /h]	1.404	1.188	1.512	
hora	r_1 [hm ³]	r_2 [hm ³]	r_3 [hm ³]	λ [€/MWh]
1	1.0	0.0	0.0	14,8
2	0.0	0.0	0.0	16,8
3	0.0	0.0	0.0	15,8

Os quadros que contêm os requisitos dos nós e especificação dos arcos são respectivamente o Quadro 2.8 e o Quadro 2.9 seguidamente indicados.

Quadro 2.8 – Requisitos dos nós (Figura 2.9)

Nº do Nó	Requisito	Nº do Nó	Requisito
1	8.9	6	0.0
2	10.8	7	0.0
3	21.1	8	0.0
4	0.0	9	0.0
5	0.0	10	-40.8

Quadro 2.9 – Especificações dos arcos (Figura 2.9)

Nº do Arco	Nó “De”	Nó “Para”	Lucro Unit.	Capac.	Nº do Arco	Nó “De”	Nó “Para”	Lucro Unit.	Capac.
1	1	2	0	[4.0, 9.9]	14	5	8	2758	[0.0, 1.188]
2	2	3	0	[4.0, 9.9]	15	6	9	2593	[0.0, 1.512]
3	3	10	0	[4.0, 9.9]	16	7	10	2680	[0.0, 1.404]
4	4	5	0	[5.0, 13.5]	17	8	10	3042	[0.0, 1.188]
5	5	6	0	[5.0, 13.5]	18	9	10	2861	[0.0, 1.512]
6	6	10	0	[5.0, 13.5]	19	1	4	0	[0.0, —]
7	7	8	0	[10.0, 26.4]	20	2	3	0	[0.0, —]
8	8	9	0	[10.0, 26.4]	21	3	6	0	[0.0, —]
9	9	10	0	[10.0, 26.4]	22	4	7	0	[0.0, —]
10	1	4	1791	[0.0, 1.404]	23	5	8	0	[0.0, —]
11	2	3	2033	[0.0, 1.188]	24	6	9	0	[0.0, —]
12	3	6	1912	[0.0, 1.512]	25	7	10	0	[0.0, —]
13	4	7	2429	[0.0, 1.404]	26	8	10	0	[0.0, —]
					27	9	10	0	[0.0, —]

Este planeamento é simples de efectuar sem o recurso a um sistema de suporte à decisão, visto que, o que se pretende é planear a gestão de 1 hm^3 de afluência de água à primeira albufeira de forma a que este seja armazenado no fim das três horas na última albufeira, como o preço da energia é mais favorável na segunda hora, a solução óptima

deve contemplar que o máximo volume de água possível seja turbinado na segunda hora nas duas primeiras albufeiras, sendo o lucro total de €4788.

Quadro 2.10 – Solução para o planeamento.

k	r_1	v_1	t_1	s_1	r_2	v_2	t_2	s_2	r_3	v_3	t_3	s_3
o	—	7.9	—	—	—	10.8	—	—	—	21.1	—	—
1	1.0	8.9	0.0	0.0	0.0	10.8	0.0	0.0	0.0	21.1	0.0	0.0
2	0.0	7.9	1.0	0.0	0.0	10.8	1.0	0.0	0.0	22.1	0.0	0.0
3	0.0	7.9	0.0	0.0	0.0	10.8	0.0	0.0	0.0	22.1	0.0	0.0
f	—	7.9	—	—	—	10.8	—	—	—	22.1	—	—

t em percentagem do máximo volume de água turbinado

k	t_1	%	t_2	%	t_3	%
1		0,00%		0,00%		0,00%
2		71,20%		84,18%		0,00%
3		0,00%		0,00%		0,00%

Caso o horizonte temporal seja maior ou existam mais aflúncias já não seria tão fácil encontrar a solução óptima por tentativa. Por exemplo, considere que os dados para o planeamento eram os do Quadro 2.11.

Quadro 2.11 – Dados para do aproveitamento hidroeléctrico

volume [hm ³]	reser. 1	reser. 2	reser. 3	
\bar{v}	9.9	13.5	26.4	
v^o	7.9	10.8	21.1	
v^f	7.9	10.8	22.1	
\underline{v}	4	5	10	
ρ [MWh/hm ³]	121	164	181	
\bar{t} [hm ³ /h]	1.404	1.188	1.512	
hora	r_1 [hm ³]	r_2 [hm ³]	r_3 [hm ³]	λ [€/MWh]
1	3.5	0.0	0.0	14,8
2	0.0	0.0	0.0	16,8
3	0.0	0.0	0.0	15,8

Ainda, seria fácil concluir que na primeira albufera há afluência de água excessiva, i.e., tem que se descarregar 0.096 hm^3 na primeira hora. Como proceder para tomar as melhores decisões em cada hora? Um processo de tentativa e erro permite concluir que o melhor lucro é de € 23011 e que o quadro de solução é o seguinte

Quadro 2.12 – Solução para o planeamento.

k	r_1	v_1	t_1	s_1	r_2	v_2	t_2	s_2	r_2	v_3	t_3	s_3
o	—	7.9	—	—	—	10.8	—	—	—	21.1	—	—
1	3.5	9.9	1.4	0.1	0.0	11.2	1.1	0.0	0.0	22.2	0.0	0.0
2	0.0	8.5	1.4	0.0	0.0	11.4	1.2	0.0	0.0	21.9	1.5	0.0
3	0.0	7.9	0.6	0.0	0.0	10.8	1.2	0.0	0.0	22.1	1.0	0.0
f	—	7.9	—	—	—	10.8	—	—	—	22.1	—	—

t em percentagem do máximo volume de água turbinado

k	t_1	%	t_2	%	t_3	%
1	100,0%		94,6%		0,0%	
2	100,0%		100,0%		100,0%	
3	42,5%		100,0%		65,3%	

No entanto, para obter esta solução já é mais significativo o dispêndio de tempo e a dificuldade de antever a solução óptima. Embora, após a obtenção desta, uma análise da solução recorrendo à rede apresentada na Figura 2.9 permite concordar com os valores apresentados. Neste exemplo os requisitos para os nós passam a ser os indicados seguidamente no Quadro 2.13

Quadro 2.13 – Requisitos dos nós (afluência $r_1^1 = 3.5 \text{ hm}^3$)

Nº do Nó	Requisito	Nº do Nó	Requisito
1	10.4	6	0.0
2	10.8	7	0.0
3	21.1	8	0.0
4	0.0	9	0.0
5	0.0	10	-43.3

2.3.5- Planeamento em aproveitamentos de biogás, curto prazo, modelo LP

Em aterros sanitários, o biogás produzido é uma mistura cujos principais gases são o metano (40 a 60 %), dióxido de carbono (20 a 40 %) e nitrogénio (2 a 20 %). A produção de biogás varia ao longo dos anos, à medida que se processa a decomposição dos resíduos. O metano é sob o ponto de vista do efeito de estufa um gás cerca de vinte e uma vezes mais prejudicial que o dióxido de carbono. Portanto, mesmo que não seja aproveitado deve ser queimado. O aproveitamento energético do biogás para a conversão para energia eléctrica conduz à valorização económica proveniente da venda da energia eléctrica, com preços melhores que os das fontes convencionais, e representa um benefício ambiental. A Gestão de Curto-Prazo para conversão da energia do biogás para a forma de energia eléctrica consiste em estabelecer um plano possível para o funcionamento das unidades, podendo serem tomadas decisões em estádios de uma hora durante um horizonte temporal de um dia a uma semana.

A gestão desta fonte de energia no curto-prazo tem que considerar a variação dos preços e a emissão de metano estimada recorrendo ao histórico e ou particularmente numa fase de projecto ao modelo LandGEM da USEPA, <http://www.epa.gov>, utilizado na Europa. O comportamento do jogo dos mecanismos técnico-económico neste tipo de aproveitamento pode ser modelizado por um problema de programação matemática linear. A função objectivo é o proveito económico total obtido ao longo do horizonte temporal calculado por

$$\sum_{k \in K} \lambda_k \left(\sum_{i \in I} p_i^k - d_k \right)$$

sendo p_i^k a potência eléctrica na unidade i no estádio k e d_k a potência eléctrica associada ao consumo próprio de energia eléctrica, energia não exportada, no estádio k .

O consumo do biogás e o nível de emissão poluente de uma unidade é calculado respectivamente por

$$C_{ik}(p_i^k) = \alpha_{ik} + \beta_{ik} p_i^k \quad E_{ik}(p_i^k) = a_{ik} + b_{ik} p_i^k$$

As restrições podem ser limitações no consumo de biogás em cada estádio, a emissão poluente limitada ao longo do horizonte temporal para um conjunto de unidades.

$$\underline{C}_{jk}^{req} \leq \sum_{i \in A_{jk}} C_{ik}(p_i^k) \leq \bar{C}_{jk}^{req} \quad j \in J, k \in K$$

$$\underline{E}_n^{req} \leq \sum_{i \in B_n} \sum_{k \in K} E_{ik}(p_i^k) \leq \bar{E}_n^{req} \quad n \in N$$

Ainda podem ser imposições no nível de potência das unidades: se a unidade estiver em funcionamento, potência compreendida entre um valor mínimo e o valor máximo de potência possível; caso contrário nula, impondo a introdução de variáveis inteiras. No entanto, para exemplo de aplicação, será admitido que o grupo motogerador tem que estar em funcionamento durante o horizonte temporal considerado. Portanto, não é necessário introduzir variáveis inteiras. O problema de programação linear para este planeamento será escrito como

$$\max \sum_{k \in K} \lambda_k (\sum_{i \in I} p_i^k - d_k)$$

s.a

para $j \in J, k \in K$ e $n \in N$

$$\underline{C}_{jk}^{req} \leq \sum_{i \in A_{jk}} C_{ik}(p_i^k) \leq \bar{C}_{jk}^{req}$$

$$\underline{E}_n^{req} \leq \sum_{i \in B_n} \sum_{k \in K} E_{ik}(p_i^k) \leq \bar{E}_n^{req}$$

$$\underline{p}_i \leq p_i^k \leq \bar{p}_i$$

Considere um aproveitamento de biogás com um motogerador com a potência eléctrica máxima de 800 kW, sendo o horizonte temporal de um dia dividido em períodos de decisão de uma hora. Considere as seguintes restrições: manter o aproveitamento em conversão durante as 24 horas; satisfazer o consumo de energia eléctrica próprio, indicado nas colunas ‘d’ do Quadro 2.11, exportando a restante energia para venda de acordo com os preços; respeitar a previsão de 4000 unidades de volume disponível de biogás durante as 24 horas, devendo utilizar em cada hora uma quantidade não inferior a 60 e não superior a 300 unidades de volume, sendo a emissão poluente total no fim das 24 horas não superior a 3500 unidades de massa. Ainda, considere os seguintes dados que caracterizam respectivamente o consumo de biogás com a operação da unidade e o nível de emissão poluente

$$\alpha_k = 10 \quad \beta_k = 0.35 \quad a_k = 45 \quad b_k = 0.2$$

Os preços para a energia eléctrica estão indicados nas colunas ‘preço’ do Quadro 2.14.

A solução óptima está também indicada no Quadro 2.14.

Quadro 2.14 – Solução para o planeamento.

h	d	fuel	emi	Preço	p	€	h	d	fuel	emi	preço	p	€
1	21	60	74	0,080	143	10	13	20	290	205	0,105	800	82
2	20	60	74	0,080	143	10	14	21	290	205	0,122	800	95
3	20	60	74	0,080	143	10	15	23	290	205	0,122	800	95
4	19	60	74	0,080	143	10	16	24	290	205	0,122	800	95
5	19	60	74	0,080	143	10	17	24	254	184	0,105	696	71
6	18	60	74	0,080	143	10	18	25	290	205	0,122	800	95
7	17	60	74	0,080	143	10	19	25	290	205	0,122	800	95
8	17	60	74	0,080	143	10	20	24	211	160	0,105	575	58
9	17	60	74	0,080	143	10	21	24	211	160	0,105	575	58
10	18	60	74	0,080	143	10	22	23	211	160	0,105	575	58
11	18	60	74	0,080	143	10	23	22	211	160	0,105	575	58
12	19	290	205	0,105	800	82	24	21	211	160	0,105	575	58
	total	950	1014	—	2371	191		total	4000	3229	—	10743	1107

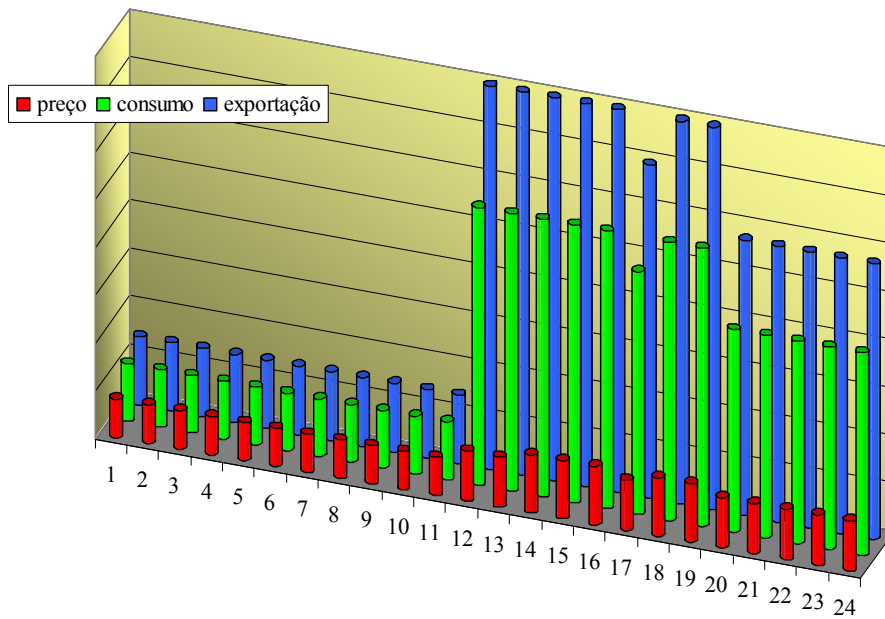


Figura 2.10 – Preços, consumo de metano, energia eléctrica exportada.

Observe que este planeamento, suportado por um problema de programação linear, necessita de dados que são estimados, atendendo ao facto do horizonte temporal ser de curto prazo o problema é geralmente tratado em âmbito determinístico, i.e., os valores médios são considerados.

3- PROGRAMAÇÃO LINEAR (PL)

O problema de programação linear (PL) é expresso em programação matemática pela minimização (ou maximização) de uma função linear, que se designa por *função objetivo*, sujeita a restrições, normalmente expressas na forma de inequações. O problema PL será formulado como se segue:

$$\begin{aligned} \min(\text{máx}) z &= c_1x_1 + c_2x_2 + \dots + c_Nx_N \\ \text{sujeito a :} \\ a_{11}x_1 + a_{12}x_2 + \dots + a_{1N}x_N &\geq (\leq) b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2N}x_N &\geq (\leq) b_2 \\ &\cdot \qquad \qquad \qquad \cdot \\ &\cdot \qquad \qquad \qquad \cdot \\ &\cdot \qquad \qquad \qquad \cdot \\ a_{M1}x_1 + a_{M2}x_2 + \dots + a_{MN}x_N &\geq (\leq) b_M \end{aligned}$$

$$0 \leq x_i \leq u_i, i = 1, 2, \dots, N \text{ (restrições de não negatividade)}$$

O uso de restrições de igualdade é mais conveniente para o estudo da metodologia que será apresentada neste texto para solução do problema PL. Por isso, convém converter as inequações que expressam as restrições em equações, introduzindo no problema novas variáveis, também não negativas, designadas por variáveis auxiliares ou de desvio. Esta conversão conduz a uma formulação do problema que se convencionou chamar *forma canónica ou normalizada*.

O problema na forma canónica é descrito como se segue:

$$\min(\text{máx}) z = c_1 x_1 + c_2 x_2 + \dots + c_N x_N + 0x_{N+1} + \dots + 0x_{N+M}$$

sujeito a :

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1N}x_N - (+) x_{N+1} = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2N}x_N - (+) x_{N+2} = b_2$$

$$\cdot \quad \quad \quad \cdot$$

$$\cdot \quad \quad \quad \cdot$$

$$\cdot \quad \quad \quad \cdot$$

$$a_{M1}x_1 + a_{M2}x_2 + \dots + a_{MN}x_N - (+) x_{N+M} = b_M$$

$$0 \leq x_j \leq u_j, j = 1, 2, \dots, N, N+1, \dots, N+M$$

ou ainda

$$\min(\text{máx}) z = c_1 x_1 + c_2 x_2 + \dots + c_N x_N + c_{N+1} x_{N+1} + \dots + c_{N+M} x_{N+M}$$

sujeito a :

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1N}x_N + a_{1,N+1}x_{N+1} + \dots + a_{1,N+M}x_{N+M} = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2N}x_N + a_{2,N+1}x_{N+1} + \dots + a_{2,N+M}x_{N+M} = b_2$$

$$\cdot \quad \quad \quad \cdot$$

$$\cdot \quad \quad \quad \cdot$$

$$\cdot \quad \quad \quad \cdot$$

$$a_{M1}x_1 + a_{M2}x_2 + \dots + a_{MN}x_N + a_{M,N+1}x_{N+1} + \dots + a_{M,N+M}x_{N+M} = b_M$$

$$0 \leq x_j \leq u_j, j = 1, 2, \dots, N, N+1, \dots, N+M$$

com

$$c_{N+i} = 0, a_{i,N+i} = \pm 1, a_{i,N+j} = 0, i = 1, \dots, M; j = 1, \dots, M (i \neq j)$$

Sendo

$$m = M \quad \text{número de restrições}$$

$$n = N + M \quad \text{número de variáveis, incluindo as principais e auxiliares}$$

Atendendo a que $\min z = -\text{máx}(-z)$, poderemos optar por um dos operadores, no caso vamos optar pelo de minimização. O problema de PL estudado neste texto está na seguinte forma canónica:

$$\min z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

sujeito a :

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$\cdot \qquad \qquad \qquad \cdot$$

$$\cdot \qquad \qquad \qquad \cdot$$

$$\cdot \qquad \qquad \qquad \cdot$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

$$0 \leq x_j \leq u_j, j = 1, 2, \dots, n$$

Ou na forma matricial como se segue:

$$\min \mathbf{cx}$$

s.a.

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$$

Em que \mathbf{A} é uma matriz $m \times n$, com $m < n$, dos coeficientes das equações associadas às restrições; \mathbf{x} é um vector com n coordenadas que são as variáveis de decisão; $\mathbf{0}$ e \mathbf{u} vectores colunas em que as coordenadas são os valores limite das variáveis de decisão; \mathbf{c} vector linha, dos coeficientes da função objectivo; \mathbf{b} vector coluna dos valores impostos para requisitos determinados pelas restrições. Nesta forma canónica, o problema PL por ser um problema de minimização permite que a função objectivo seja interpretada como um custo. Cada coordenada do vector \mathbf{c} é um coeficiente de custo associado à respectiva coordenada do vector \mathbf{x} .

No texto seguinte será estudado os procedimentos a seguir para a resolução de um problemas de programação linear (PL) que esteja formulado de acordo com forma canónica. Iniciaremos com alguns tópicos matemáticos, importantes para o estudo do problema PL, apresentando de seguida o *método simplex*.

3.1- Propriedades do espaço vectorial euclidiano

Neste subcapítulo será recordado algumas noções fundamentais. Denomina-se \mathbf{R}^n ao espaço Euclidiano n-dimensional, cuja norma ou comprimento do vector $\mathbf{x} \in \mathbf{R}^n$ é calculada por $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$. Um vector cuja norma $\|\mathbf{x}\| = 1$, é dito vector unitário.

Consideremos dois pontos distintos \mathbf{x}_1 e \mathbf{x}_2 do espaço \mathbf{R}^n e o conjunto de escalares $\Lambda \subset \mathbf{R}$, o conjunto de combinações lineares dos pontos \mathbf{x}_1 e \mathbf{x}_2 usando os escalares do conjunto Λ definem o conjunto Γ^\wedge , dos pontos \mathbf{x} que obedecem a:

$$\Gamma^\wedge = \{ \mathbf{x} : \mathbf{x} = (1 - \lambda) \mathbf{x}_1 + \lambda \mathbf{x}_2, \lambda \in \Lambda \}.$$

Se $\Lambda = \mathbf{R}^1$ então o conjunto Γ^\wedge define uma linha que passa pelos pontos \mathbf{x}_1 e \mathbf{x}_2 na direcção do primeiro ponto para o segundo. Se $\Lambda = \{ \lambda : 0 \leq \lambda \}$ o conjunto Γ^\wedge define uma semi-recta fechada em \mathbf{x}_1 e com direcção definida pelo vector $\mathbf{x}_2 - \mathbf{x}_1$. Se $\Lambda = \{ \lambda : 0 \leq \lambda \leq 1 \}$ o conjunto Γ^\wedge define um segmento de recta fechado com os pontos extremos \mathbf{x}_1 e \mathbf{x}_2 . Este segmento de recta é indicado por $[\mathbf{x}_1, \mathbf{x}_2]$.

Um ponto $\mathbf{x} \in X$ é um ponto extremo do conjunto convexo X se não pertence a qualquer segmento de recta que una dois outros pontos distintos quaisquer de X , ou dito de outra maneira, um ponto extremo de X não pode ser obtido pela combinação convexa de quaisquer outros dois pontos distintos de X .

Um conjunto $X \subset \mathbf{R}^n$ é um conjunto convexo se para quaisquer pontos \mathbf{x}_1 e \mathbf{x}_2 pertencentes a X , o conjunto dos pontos do segmento de recta fechado de extremos \mathbf{x}_1 e \mathbf{x}_2 é um subconjunto de X .

O termo fechado (semi-recta fechada, segmento de recta fechado) significa que os pontos extremos também pertencem ao conjunto dos pontos. Se nas definições de Λ , expressas acima, substituirmos o símbolo \leq por $<$, usa-se a designação de abertos (semi-recta aberta, segmento de recta aberto), ou seja, os pontos extremos não pertencem ao conjunto de pontos considerados. Dado um ponto $\mathbf{x}^0 \in \mathbb{R}^n$ e um escalar $\varepsilon > 0$, o conjunto dos pontos $\mathbf{x} \in \mathbb{R}^n$ cuja distância ao ponto \mathbf{x}^0 é inferior a ε e que se indica da forma $B_\varepsilon(\mathbf{x}^0) = \{\mathbf{x}: \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x} - \mathbf{x}^0\| < \varepsilon\}$ é denominado bola aberta com centro \mathbf{x}^0 e raio ε .

Um ponto \mathbf{x} é dito interno ao conjunto $X \subset \mathbb{R}^n$ se, existir uma bola aberta de centro em \mathbf{x} e raio ε incluída em X , isto é, se existir $\varepsilon > 0$ tal que $B_\varepsilon(\mathbf{x}) \subset X$. Um ponto \mathbf{x} é dito externo ao conjunto $X \subset \mathbb{R}^n$ se, existir uma bola aberta de centro em \mathbf{x} e raio ε tal que $B_\varepsilon(\mathbf{x}) \cap X = \emptyset$. Qualquer ponto \mathbf{x} que não é interno nem externo do conjunto $X \subset \mathbb{R}^n$ é dito ponto da fronteira de X . Um ponto extremo de um conjunto convexo é um ponto da fronteira desse conjunto, mas nem todos os pontos da fronteira são extremos, como ilustra a figura seguinte:

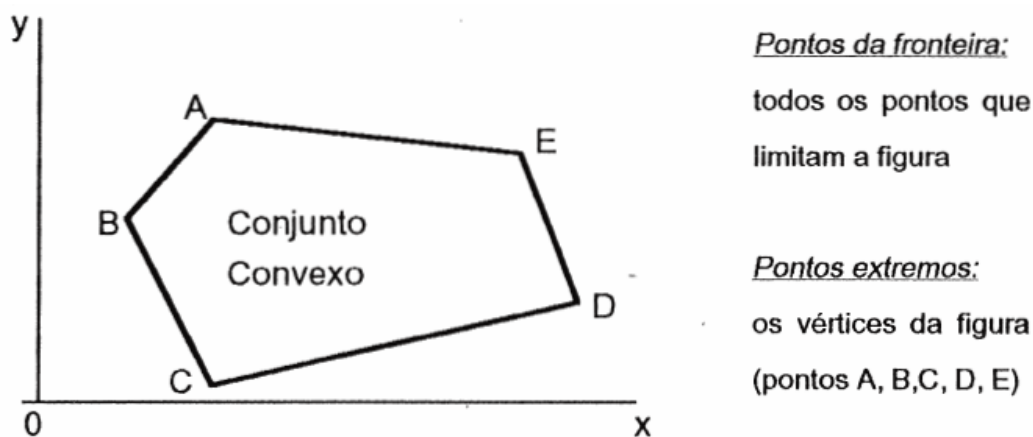


Figura 3.1 – Ilustração do conceito de ponto extremo.

Um conjunto é dito aberto se todos os pontos do conjunto são internos ao conjunto. Um conjunto é dito fechado se os pontos da fronteira do conjunto pertencem ao conjunto.

Um conjunto $X \subset \mathbf{R}^n$ é dito limitado se existe algum $\varepsilon > 0$ tal que o conjunto X está incluído no conjunto $B_\varepsilon(0)$, isto é, existe $B_\varepsilon(0)$ tal que $X \subset B_\varepsilon(0)$.

Uma função f no conjunto $X \subset \mathbf{R}^n$ é contínua, se $f(\mathbf{x})$ existe para qualquer $\mathbf{x} \in X$, e se para qualquer $\varepsilon > 0$ e $\mathbf{x}^0 \in X$ $\exists \delta > 0$ tal que $|f(\mathbf{x}) - f(\mathbf{x}^0)| < \varepsilon$ sempre que $\mathbf{x} \in [X \cap B_\delta(\mathbf{x}^0)] \subset \mathbf{R}^n$.

Uma função f é convexa num conjunto convexo $X \subset \mathbf{R}^n$ se, para quaisquer dois pontos \mathbf{x}_1 e \mathbf{x}_2 pertencentes a X e um escalar λ tal que $0 \leq \lambda \leq 1$ for válido $f[(1-\lambda)\mathbf{x}_1 + \lambda\mathbf{x}_2] \leq (1-\lambda)f(\mathbf{x}_1) + \lambda f(\mathbf{x}_2)$, a figura seguinte ilustra uma função real de variável real convexa.

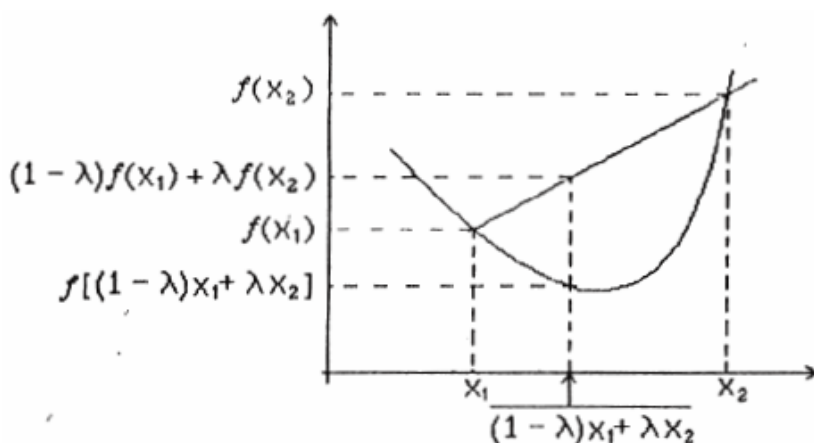


Figura 3.2 – Ilustração de uma função convexa.

Uma função f é côncava num conjunto convexo $X \subset \mathbf{R}^n$ se para quaisquer dois pontos \mathbf{x}_1 e \mathbf{x}_2 pertencentes a X e um escalar λ tal que $0 \leq \lambda \leq 1$ for válido

$f[(1-\lambda)x_1 + \lambda x_2] \geq (1-\lambda)f(x_1) + \lambda f(x_2)$, a figura seguinte ilustra uma função real de variável real côncava.

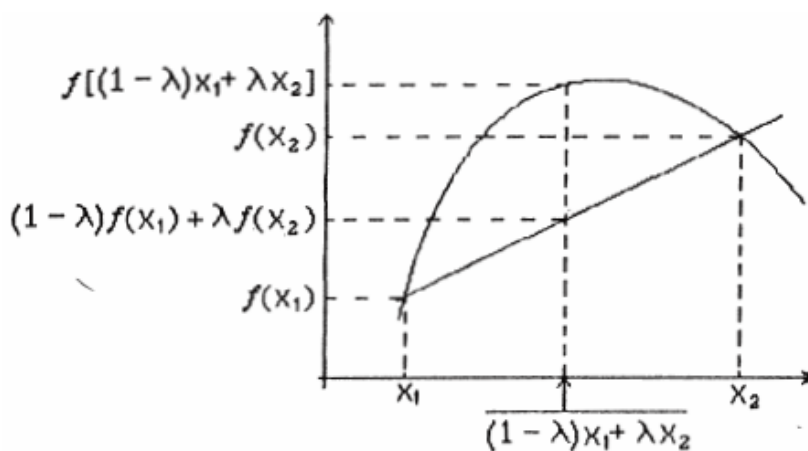


Figura 3.3 – Ilustração de uma função côncava.

Recordamos, em seguida, algumas proposições importantes para o estudo que será feito da resolução de problemas de programação PL na sua formulação canónica.

Proposição 3.1

Qualquer função que é contínua num conjunto $X \subset \mathbb{R}^n$ fechado e limitado é limitada em X .

Proposição 3.2

Qualquer função que é contínua num conjunto $X \subset \mathbb{R}^n$ fechado e limitado tem um valor máximo e mínimo em X .

Recordamos, em seguida, os conceitos combinação convexa, hiperplano e hiper espaço limitado por um hiperplano.

Uma combinação linear de um conjunto finito de pontos $\{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ de \mathbb{R}^n , dada por $\sum_{i=1}^m c_i \mathbf{x}_i$, é chamada combinação convexa se os escalares $c_i \geq 0$ e $\sum_{i=1}^m c_i = 1$.

Dado um conjunto $X \subset \mathbb{R}^n$, o invólucro do conjunto X é definido pela união de todas as combinações lineares convexas de pontos dos subconjuntos finitos de X .

O conjunto de pontos definido por $H(\mathbf{d}, w) = \{\mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \mathbf{d} \cdot \mathbf{x} = w\}$, é designado por hiperplano, sendo \mathbf{d} um vector pertencente a \mathbb{R}^n normal ao hiperplano, $\mathbf{x} \in \mathbb{R}^n$ é um ponto do espaço e w é um escalar. Desenvolvendo $\mathbf{d} \cdot \mathbf{x} = w$ tem-se:

$$d_1 x_1 + d_2 x_2 + \dots + d_n x_n = w$$

pelo que w está relacionado com a distancia do hiperplano à origem.

Associados com a definição de um hiperplano $H(\mathbf{d}, w)$ existem dois conjuntos, denominados semi-hiperespaços fechados, definidos por:

$$H_L(\mathbf{d}, w) = \{\mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \mathbf{d} \cdot \mathbf{x} \leq w\} \text{ semi-hiperespaço fechado inferior}$$

e

$$H_U(\mathbf{d}, w) = \{\mathbf{x} : \mathbf{x} \in \mathbb{R}^n, \mathbf{d} \cdot \mathbf{x} \geq w\} \text{ semi-hiperespaço fechado superior.}$$

Um politopo é por definição a intersecção de um número finito de semi-espacos fechados. E um poliedro é por definição um politopo limitado. Esta definição merece uma observação, i.e., não é concordante o seu uso, alguns definem estes dois conceitos

ao contrário do que se fez aqui, levantando inconvenientes e obrigando a esclarece o que se entende pelos conceitos no texto.

Um ponto extremo de um politopo designa-se vértice. Uma aresta é um segmento de recta fechado que limita um politopo. Estes conceitos são utilizados para enunciar as seguintes preposições sobre poliedros e politopos.

Preposição 3.3

O número de vértices e arestas de um poliedro é finito.

Preposição 3.4

O número de vértices de um poliedro não vazio é diferente de zero.

Preposição 3.5

O invólucro convexo dos vértices de um politopo não vazio é o politopo.

Preposição 3.6

Num poliedro convexo fechado qualquer ponto do poliedro pode ser representado como uma combinação convexa dos pontos extremos desse poliedro.

Feita esta abordagem de conceitos, para apresentar o método simplex vamos fazer a caracterização do domínio das soluções possíveis e a identificação neste conjunto das soluções básicas possíveis, das quais será pelos menos uma ótima.

3.2- Caracterização do domínio das soluções possíveis

Nesta secção estudaremos a caracterização do domínio das soluções possíveis e dentro destas soluções estudaremos as soluções básicas possíveis com o objectivo de determinar a melhor, i.e., estudaremos como seleccionar entre a infinidade de soluções possíveis do problema aquelas que são soluções básicas e dentro destas pelo menos uma é óptima. Esta é a abordagem do método que será apresentado posteriormente dito de método do Simplex. O domínio das soluções possíveis para a PL é descrito como se segue $Q = \{ \mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \}$. Neste estudo assumimos que o problema tem pelo menos uma solução possível, i.e., $Q \neq \emptyset$. Esta hipótese não é restritiva, uma vez que, tanto para problemas PL, como de programação em redes, podemos introduzir variáveis artificiais que tornam o conjunto de soluções Q não vazio, claro num espaço de maior dimensão, i.e., com maior número de variáveis. Partindo da hipótese de que o problema tem soluções possíveis, $Q \neq \emptyset$, vamos mostrar que existe um ponto extremo de Q que é óptimo.

Proposição 3.7

Existe um ponto extremo de Q que é óptimo para o problema PL.

Prova: Primeiro, vamos provar que o domínio das soluções possíveis Q é um poliedro finito. Para isso, considere-se o conjunto de restrições para o problema, escrito na forma matricial, $\mathbf{A}\mathbf{x} = \mathbf{b}$, o qual pode ser reescrito na forma:

$$\begin{cases} \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ \mathbf{A}\mathbf{x} \geq \mathbf{b} \end{cases} \quad (3.1)$$

De igual modo $\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$ (restrições de não negatividade) pode ser reescrito na forma:

$$\begin{cases} \mathbf{x} \leq \mathbf{u} \\ \mathbf{x} \geq \mathbf{0} \end{cases} \quad (3.2)$$

De acordo com o exposto anteriormente, 3.1 e 3.2 definem a intercepção de semi-hiperespaços fechados, sendo Q a intersecção desses semi-espacos, conclui-se que Q é um politopo. Ainda, atendendo a 3.2, o conjunto Q é limitado, ficando assim provado que Q é um poliedro. ♦

A função objectivo, $\mathbf{c} \mathbf{x}$, é linear portanto é contínua em Q . Pelas preposições anteriores 3.1 a 3.2 $\mathbf{c} \mathbf{x}$ é limitada e toma um valor mínimo que designaremos por z^* num ponto $\mathbf{x}^* \in Q$. Pela hipótese $Q \neq \emptyset$ e pelas preposições 3.3 a 3.5 o conjunto Q tem um número finito não nulo de vértices, que designaremos pelos elementos do conjunto $\{\mathbf{x}_1, \dots, \mathbf{x}_m\}$, sendo Q o invólucro convexo desses vértices. Então existe um vector $\mathbf{d} \in \mathbf{R}^m$ tal que

$$\mathbf{x}^* = \sum_{i=1}^m d_i \mathbf{x}_i, \text{ onde } \sum_{i=1}^m d_i = 1 \text{ e } \mathbf{d} \geq 0.$$

Seja o conjunto $J = \{i : d_i > 0\}$, i.e., é o conjunto dos índices i para os quais d_i é não nulo e $J \neq \emptyset$. Então

$$\mathbf{x}^* = \sum_{i \in J} d_i \mathbf{x}_i \quad \text{com} \quad \sum_{i \in J} d_i = 1.$$

Por contradição, vamos provar que pelo menos um ponto extremo de Q \mathbf{x}_i é óptimo.

Seja $\mathbf{c} \mathbf{x}^* = z^*$, recordar que, z^* o mínimo da função objectivo $\mathbf{c} \mathbf{x}$ em Q . Para qualquer

\mathbf{x}_i tal que $i \in J$, então $\mathbf{c}_i \mathbf{x}_i \geq z^*$. Seja $k \in J$ tal que $\mathbf{c} \mathbf{x}_k > z^*$, virá:

$$\begin{aligned} z^* = \mathbf{c} \mathbf{x}^* &= \mathbf{c} \sum_{i \in J} d_i \mathbf{x}_i = \sum_{i \in J} d_i (\mathbf{c} \mathbf{x}_i) = d_k (\mathbf{c} \mathbf{x}_k) + \sum_{i \in J - \{k\}} d_i (\mathbf{c} \mathbf{x}_i) \geq \\ &\geq d_k (\mathbf{c} \mathbf{x}_k) + \sum_{i \in J - \{k\}} d_i z^* > d_k z^* + \sum_{i \in J - \{k\}} d_i z^* = \underbrace{\sum_{i \in J} d_i}_{=1} z^* = z^* \end{aligned}$$

O que é uma contradição, i.e., para qualquer $i \in J$, $\mathbf{c} \mathbf{x}_i = z^*$. ♦

Da prova desta proposição pode também concluir que se existem dois pontos extremos óptimos então um ponto que pertença à combinação convexa desses dois pontos é também óptimo. Em geral se existirem vários pontos extremos óptimos os pontos que pertençam à combinação convexa desses pontos são óptimos. Também se conclui que pelo facto de existir um ponto extremo que é óptimo então os pontos óptimos não podem ser pontos interiores ao conjunto Q a não ser que todos os pontos de Q sejam óptimos. Seja então \mathbf{A} uma matriz $m \times n$ com $m < n$ e característica $\text{rank}(\mathbf{A}) = m$. Então, através de uma permuta de colunas da matriz \mathbf{A} , a matriz pode ser particionada na seguinte forma $\mathbf{A} = [\mathbf{B} : \mathbf{N}]$, onde \mathbf{B} é uma matriz quadrada não singular. Seguindo a mesma organização de forma a tornar os produtos matriciais e as igualdades e desigualdades compatíveis com a partição podem escrever-se os vectores \mathbf{x} e \mathbf{u} de acordo com a partição, i.e., vamos reescrever as seguintes igualdades e desigualdades matriciais:

$$\mathbf{A} \mathbf{x} = \mathbf{b} \Leftrightarrow \mathbf{B} \mathbf{x}^{\mathbf{B}} + \mathbf{N} \mathbf{x}^{\mathbf{N}} = \mathbf{b}$$

As coordenadas do vector \mathbf{x} ligadas com as colunas da matriz \mathbf{B} , i.e., $\mathbf{x}^{\mathbf{B}}$, são chamadas variáveis básicas; e as restantes, ligadas com as colunas da matriz \mathbf{N} , i.e., $\mathbf{x}^{\mathbf{N}}$, são chamadas variáveis não básicas. Uma solução básica possível, também dita de básica admissível, do problema de PL é por definição uma solução tal que:

$$\begin{aligned} &[\mathbf{x}^{\mathbf{B}} : \mathbf{x}^{\mathbf{N}}] \in Q \\ & \text{e} \\ & x_i^{\mathbf{N}} \in \{0, u_i^{\mathbf{N}}\} \text{ para todo o } i \end{aligned}$$

Por outras palavras, cada variável não básica ou tem valor igual a zero ou tem valor igual ao seu valor máximo para a variável do problema a que esta associada.

Vamos demonstrar agora, que um ponto extremo de Q é uma solução básica possível do problema PL, ou seja, é válida a seguinte proposição.

Proposição 3.8

Um ponto extremo de Q é uma solução básica admissível do problema PL.

Prova: Seja \mathbf{x}^* um ponto extremo de Q . E sejam as coordenadas deste ponto x_i^* ordenadas da seguinte forma $0 < x_i^* < u_i$ para $i = 1, \dots, k$ e $x_i^* \in \{0, u_i\}$ para $i = k + 1, \dots, n$, ou seja, primeiramente são escolhidas na ordenação as variáveis que não têm valor igual aos extremos e em seguida as que têm valor igual aos extremos. As colunas de A são reordenadas de modo a corresponder a esta ordem das coordenadas de \mathbf{x}^* . Vamos supor por hipótese que as primeiras k colunas de A são linearmente dependentes e, em consequência, vamos provar uma contradição o que permite concluir que essas colunas formam um sistema de vectores linearmente independentes. Atendendo à hipótese existem escalares λ_i , nem todos nulos, tal que:

$$\sum_{i=1}^{i=k} \lambda_i \mathbf{A}(i) = \mathbf{0}$$

Seja $\lambda_i \neq 0$

$$\alpha_1 = \min_{1 \leq i \leq k} \left\{ \frac{x_i^*}{|\lambda_i|} \right\} \quad \text{e} \quad \alpha_2 = \min_{1 \leq i \leq k} \left\{ \frac{u_i - x_i^*}{|\lambda_i|} \right\}$$

Seleccionando ε tal que $0 < \varepsilon < \min(\alpha_1, \alpha_2)$. Então:

$$0 < x_i^* \pm \varepsilon \lambda_i < u_i \quad \text{para } 1 \leq i \leq k \quad (3.3)$$

Seja $\boldsymbol{\mu}$ o vector n -dimensional dado por $[\lambda_1, \dots, \lambda_k, 0, \dots, 0]$ e seja:

$$\mathbf{x}_1 = \mathbf{x}^* - \varepsilon \boldsymbol{\mu} \quad \text{e} \quad \mathbf{x}_2 = \mathbf{x}^* + \varepsilon \boldsymbol{\mu}$$

por (3.3), $\mathbf{0} < \mathbf{x}_1 < \mathbf{u}$ e $\mathbf{0} < \mathbf{x}_2 < \mathbf{u}$.

Mais ainda, devido à hipótese de dependência linear e ao facto de $\mu_i = 0$ para $i \geq k + 1$, $\mathbf{A} \boldsymbol{\mu} = \sum_{i=1}^{i=n} \mu_i \mathbf{A}(i) = \mathbf{0}$. Então $\mathbf{A} \mathbf{x}_1 = \mathbf{A} \mathbf{x}^* = \mathbf{b}$ e $\mathbf{A} \mathbf{x}_2 = \mathbf{A} \mathbf{x}^* = \mathbf{b}$.

Como $\boldsymbol{\mu} \neq \mathbf{0}$, \mathbf{x}_1 e \mathbf{x}_2 são soluções possíveis, sendo diferentes entre si e diferentes de \mathbf{x}^* .

E como $\mathbf{x}^* = \frac{1}{2} \mathbf{x}_1 + \frac{1}{2} \mathbf{x}_2$, tal contradiz a hipótese de que \mathbf{x}^* é um ponto extremo e

prova que as primeiras k colunas de \mathbf{A} não podem ser linearmente dependentes. ♦

Ainda, como m é a característica da matriz \mathbf{A} e sendo $k \leq m$, então \mathbf{B} é constituído pelas primeiras k colunas de \mathbf{A} e pelas $m - k$ colunas de \mathbf{A} que constituem um sistema de m colunas linearmente independentes. Esta selecção é sempre possível, visto que, a característica de \mathbf{A} é igual a m por hipótese. ♦

Proposição 3.9

Dado um problema de PL, se existe uma solução possível, então existe uma solução básica possível. Se existe uma solução óptima possível, então existe uma solução óptima básica possível.

Prova: Se existe uma solução possível, então o poliedro $Q \neq \emptyset$. Logo pela proposição (3.7) existe um ponto extremo de Q que é um ponto óptimo para o problema PL. Pela proposição (3.8), um ponto extremo é uma solução básica possível. ♦

A proposição (3.9) é conhecida como teorema fundamental da programação linear. A proposição (3.9) permite limitar o estudo para a escolha da solução óptima do problema PL às soluções básicas possíveis, visto que, pela proposição (3.7), uma delas é óptima. O número máximo de bases possíveis para um problema com m restrições de igualdade linearmente independentes e n variáveis não é superior a

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

Com cada base está associado um número de soluções básicas, nem todas necessariamente admissíveis, igual a m^2 . Portanto como sabemos que as soluções básicas possíveis são os pontos extremos do politopo Q e são em número finito, resulta que se soubermos determinar as soluções básicas possíveis podemos por força bruta encontrar a ótima. Força bruta aqui significa determinar todos os pontos extremos de Q , o que apresenta uma ineficácia considerável. Posteriormente, escolhendo o ponto para o qual a função objectivo tem o menor valor fica identificado o ponto ótimo. O método do simplex faz melhor, vai progredir de forma racional entre pontos extremos de Q até concluir que encontrou o ponto para o qual a função objectivo tem o menor valor, i.e., tem um critério de optimilidade que termina o algoritmo. Como o número de pontos extremos de Q é finito o método do simplex é um algoritmo perfeito na verdadeira acepção da palavra, terminando após um número de iterações fixo.

3.3- Método primal simplex

O método primal simplex pode ser considerado como um caso particular dos algoritmos conhecidos por método da direcção admissível de melhoria do valor da função objectivo. Uma direcção \mathbf{d} diz-se admissível no ponto \mathbf{y} possível se existir $\tau > 0$ tal que $\mathbf{y} + \alpha \mathbf{d}$ é admissível para qualquer $0 \leq \alpha \leq \tau$ com o $\tau > 0$.

Para o problema de minimização uma direcção \mathbf{d} designa-se por direcção admissível de melhoria do valor da função objectivo no ponto \mathbf{y} , se \mathbf{d} é uma direcção admissível e a derivada direccional da função objectivo $\mathbf{c}\mathbf{x}$ segundo \mathbf{d} no ponto \mathbf{y} é negativa o que significa que segundo esta direcção existe um intervalo onde a função objectivo

decrece. Dadas as definições acima, o método geral da direcção admissível pode ser descrito como se segue:

Algoritmo (ALG) 3.1 – Método Geral da Direcção Admissível

Passo 0 – Iniciação.

Considere um ponto \mathbf{y}^0 solução possível para o problema PL, inicie a zero o índice k da iteração, i.e., faça a atribuição, $k \leftarrow 0$.

Passo 1 – Determinação de uma Direcção Admissível de melhoria do valor da Função Objectivo.

Determine \mathbf{d} uma direcção admissível de melhoria do valor da função objectivo no ponto \mathbf{y}^k . Se \mathbf{d} não existe, então o algoritmo termina e o ponto corrente \mathbf{y}^k , é o ponto solução. Caso contrário, o algoritmo prossegue para o passo 2.

Passo 2 – Obtenção de um novo ponto.

Procurar α^* tal que:

$$\mathbf{c}(\mathbf{y}^k + \alpha^* \mathbf{d}) = \min_{\alpha > 0} \{ \mathbf{c}(\mathbf{y}^k + \alpha \mathbf{d}) : \mathbf{y}^k + \alpha \mathbf{d} \in Q \}.$$

Isto é, vamos procurar o passo α^* para a translação de \mathbf{y}^k segundo vector \mathbf{d} , de modo que $\mathbf{y}^k + \alpha \mathbf{d}$ pertence ainda ao conjunto Q das soluções possíveis e que seja a melhor solução para o problema de pesquisa em linha segundo esta direcção. Encontrado α^* , actualizamos o ponto para

$\mathbf{y}^{k+1} = \mathbf{y}^k + \alpha^* \mathbf{d}$ nova solução corrente e incrementamos o contador de iterações k de um, $k \leftarrow k + 1$.

O algoritmo prossegue para o passo 1.

O método primal simplex é pois uma especialização do método geral da direcção admissível em que \mathbf{y}^k , para cada k , é um ponto extremo do domínio das soluções possíveis o poliedro Q . Por isso, frequentemente, o método simplex é chamado método do ponto extremo. A direcção \mathbf{d} terá de ser seleccionada de modo que a solução encontrada no Passo 2 resulte num novo ponto extremo. Dada uma solução básica possível para o problema PL, considere a partição das matrizes e a partição de \mathbf{c} , \mathbf{x} e \mathbf{u} de forma compatível em componentes básicos e não básicos

$$\mathbf{A} = [\mathbf{B} \mid \mathbf{N}], \quad \mathbf{c} = [\mathbf{c}^B \mid \mathbf{c}^N], \quad \mathbf{x} = [\mathbf{x}^B \mid \mathbf{x}^N], \quad \text{e} \quad \mathbf{u} = [\mathbf{u}^B \mid \mathbf{u}^N].$$

O problema PL será escrito como se segue:

$$\min \quad \mathbf{c}^B \mathbf{x}^B + \mathbf{c}^N \mathbf{x}^N \tag{3.4}$$

s.a

$$\mathbf{B}\mathbf{x}^B + \mathbf{N}\mathbf{x}^N = \mathbf{b} \tag{3.5}$$

$$\mathbf{0} \leq \mathbf{x}^B \leq \mathbf{u}^B \tag{3.6}$$

$$\mathbf{0} \leq \mathbf{x}^N \leq \mathbf{u}^N \tag{3.7}$$

Nesta forma, o problema PL permite interpretar que cada coordenada do vector \mathbf{c}^B é um coeficiente de custo associado à respectiva variável básica coordenada do vector \mathbf{x}^B ; e que cada coordenada do vector \mathbf{c}^N é um coeficiente de custo associado à respectiva variável não básica coordenada do vector \mathbf{x}^N .

Como \mathbf{B} por hipótese é uma matriz não singular, por (3.5) resulta

$$\mathbf{x}^B = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{N}\mathbf{x}^N).$$

Substituindo este resultado em (3.4) e (3.6) obtemos o seguinte problema equivalente:

$$\min \mathbf{c}^B \mathbf{B}^{-1} \mathbf{b} + (\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}) \mathbf{x}^N \quad (3.8)$$

$$\text{s.a. } \mathbf{0} \leq \mathbf{B}^{-1} \mathbf{b} - \mathbf{B}^{-1} \mathbf{N} \mathbf{x}^N \leq \mathbf{u}^B \quad (3.9)$$

$$\mathbf{0} \leq \mathbf{x}^N \leq \mathbf{u}^N \quad (3.10)$$

Nesta forma, houve no problema PL uma redução do número de variáveis, visto que, nesta forma as variáveis de decisão são só as não básicas. O vector dos coeficientes das variáveis não básicas $\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}$ é dito de vector dos custos reduzidos.

Hipótese: Assumimos para simplificar o estudo que $\mathbf{0} < \mathbf{B}^{-1} \mathbf{b} - \mathbf{B}^{-1} \mathbf{N} \mathbf{x}^N < \mathbf{u}^B$. Tal corresponde a assumir o que frequentemente se designa por hipótese da não degenerência, corresponde a afirmar que as coordenadas básicas não se encontram com os valores limites, como tem que acontecer com as coordenadas não básicas.

Consideremos o conjunto de direcções $D = \{\mathbf{e}^1, \mathbf{e}^2, \dots, \mathbf{e}^{n-m}, -\mathbf{e}^1, -\mathbf{e}^2, \dots, -\mathbf{e}^{n-m}\}$, no espaço não básico. No estudo do algoritmo primal simplex limitaremos a nossa atenção às direcções deste conjunto D com o objectivo de obter sempre uma solução básica possível.

O vector \mathbf{e}^i é uma direcção admissível se $x_i^N < u_i^N$, i.e., a variável de decisão não básica x_i^N é inferior ao valor máximo admissível e $-\mathbf{e}^i$ será uma direcção admissível se $x_i^N > 0$, i.e., a variável de decisão não básica x_i^N é superior ao valor mínimo admissível.

Recorrendo à derivada direccional no espaço não básico mostraremos como escolher uma direcção admissível de melhoria do valor da função objectivo. A derivada de $g(\mathbf{x})$ segundo \mathbf{v} no ponto \mathbf{y} é por definição calculada como

$$D_{\mathbf{v}}g(\mathbf{y}) = \lim_{t \rightarrow 0^+} \frac{g(\mathbf{y} + t \mathbf{v}) - g(\mathbf{y})}{t}$$

recorrendo ao gradiente a fórmula para o cálculo desta derivada direccional é seguinte:

$$D_{\mathbf{v}}g(\mathbf{y}) = \nabla g(\mathbf{y}) \mathbf{v}$$

Assim para o problema (3.8) a (3.10)

$$g(\mathbf{x}^N) = \mathbf{c}^B \mathbf{B}^{-1} \mathbf{b} + (\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}) \mathbf{x}^N$$

portanto

$$\nabla g(\mathbf{y}) = \mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}$$

é independente do ponto em que é calculado como seria de esperar. A procura de uma direcção admissível de entre as direcções do conjunto D , para melhorar o valor da função objectivo, será segundo o vector \mathbf{e}^i direcção admissível de melhoria da função objectivo se $x_i^N < u_i^N$ e $(\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_i < 0$ ou segundo $-\mathbf{e}^i$ se $x_i^N < u_i^N$ e $(\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_i > 0$.

Para às soluções básicas possíveis, $x_i^N \in \{0, u_i^N\}$, as condições que nos permitem escolher direcção admissível de melhoria da função objectivo são dadas por

$$\left\{ \begin{array}{l} \mathbf{e}^i : x_i^N = 0 \quad \text{e} \quad (\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_i < 0 \\ -\mathbf{e}^i : x_i^N = u_i^N \quad \text{e} \quad (\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_i > 0 \end{array} \right\}.$$

Agora, só falta descrever como estabelecer a metodologia para o passo 2 do algoritmo anterior, obtenção de uma nova solução. Como já é conhecido a direcção admissível de melhoria da função objectivo no ponto \mathbf{x}^N determinada por

$$\delta \mathbf{e}^i \text{ sendo } \delta \equiv \begin{cases} 1, & \text{se } x_i^N = 0 \\ -1, & \text{se } x_i^N = u_i^N \end{cases}$$

falta determinar um translação do ponto corrente segundo o vector da direcção admissível de melhoria da função objectivo.

O problema a resolver para estabelecer a metodologia para o passo 2 tem só uma variável de decisão, a amplitude do passo, e é então descrito como se apresenta de seguida

$$\begin{aligned} \min_{\alpha > 0} \quad & \mathbf{c}^B \mathbf{B}^{-1} \mathbf{b} + \sum_{j \neq i} (\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_j x_j^N + (\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_i (x_i^N + \alpha \delta) \\ \text{s.a} \quad & 0 \leq x_i^N + \alpha \delta \leq u_i^N \\ & \mathbf{0} \leq \mathbf{x}^B - \mathbf{B}^{-1} \mathbf{N}(i) \alpha \delta \leq \mathbf{u}^B \end{aligned}$$

ou seja eliminando a presença das constantes na função objectivo

$$\begin{aligned} \min_{\alpha > 0} \quad & (\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_i \alpha \delta \\ \text{s.a} \quad & 0 \leq x_i^N + \alpha \delta \leq u_i^N \end{aligned} \quad (3.11)$$

$$\mathbf{0} \leq \mathbf{x}^B - \mathbf{B}^{-1} \mathbf{N}(i) \alpha \delta \leq \mathbf{u}^B \quad (3.12)$$

É de salientar que, devido à linearidade da função $g(\mathbf{x}^N)$, qualquer incremento em α resulta num decréscimo no valor da função objectivo. Assim, interessa-nos fazer α o maior possível desde que satisfaça (3.11) e (3.12). Se $\mathbf{d} \in \mathbf{D}$, então se nos deslocarmos segundo a direcção \mathbf{d} pode eventualmente acontecer que alguma variável x_j assumam valores fora dos limites, $0 \leq x_j \leq u_j$. A primeira variável para a qual isto ocorra, designaremos por *variável de bloqueio*. Seja:

$$\alpha_1 = \max_{\alpha > 0} \{ \alpha : 0 \leq x_i^N + \alpha \delta \leq u_i^N \},$$

e

$$\alpha_2 = \max_{\alpha > 0} \{ \alpha : \mathbf{0} \leq \mathbf{x}^B - \mathbf{B}^{-1}\mathbf{N}(i) \alpha \delta \leq \mathbf{u}^B \}.$$

Supondo $\alpha_1 \leq \alpha_2$, então a *variável de bloqueio* é a coordenada não básica correspondente à direcção $\mathbf{d} = \delta \mathbf{e}^i$, e as variáveis básicas continuam a ser básicas. Se $\alpha_1 > \alpha_2$, então adicionando a \mathbf{x}^N o incremento $\alpha_2 \mathbf{d}$ forçamos alguma variável básica ou a ser igual ao seu valor mínimo ou a ser igual ao seu valor máximo. Tal obriga a refazer a partição da matriz \mathbf{A} , visto que, esta variável básica passa a não básica e a variável não básica corresponde à direcção \mathbf{d} passa a básica. Obtemos assim uma nova solução básica possível. Com a formulação apresentada, podemos construir o algoritmo primal simplex para os problemas de programação linear como se apresenta em seguida.

ALG 3.2 – Algoritmo Primal Simplex

Passo 0 – Iniciação

Seja $[\mathbf{x}^B \mid \mathbf{x}^N]$ uma solução básica possível, associada com partição $\mathbf{A} = [\mathbf{B} \mid \mathbf{N}]$.

Passo 1 – Avaliação

Seja

$$\begin{aligned} \psi_1 &= \{ i : x_i^N = 0 & \text{e} & (\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_i < 0 \} \\ \psi_2 &= \{ i : x_i^N = u_i^N & \text{e} & (\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_i > 0 \} \end{aligned}$$

Em que ψ_1 é o conjunto das variáveis não básicas que têm valor nulo e a respectiva coordenada do vector do custo reduzido $(\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_i < 0$ e ψ_2 é o conjunto dos índices das variáveis não básicas que têm valor máximo e a respectiva coordenada do vector do custo reduzido $(\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N})_i > 0$.

No óptimo se a coordenada do vector de custo reduzido for negativa a respectiva variável não básica tem que estar no valor máximo, caso contrário existe hipótese de melhoria da função objectivo por aumento desta variável. Ainda, se uma coordenada do vector custo reduzido for positiva a respectiva variável não básica tem que estar no valor nulo, caso contrário existe hipótese de melhoria da função objectivo por diminuição desta variável.

A condição de união $\psi_1 \cup \psi_2 = \emptyset$, termina o algoritmo e o ponto óptimo é o ponto corrente $\mathbf{x} = [\mathbf{x}^B \mid \mathbf{x}^N]$. Caso contrário, o algoritmo prossegue com a selecção de um $k \in \psi_1 \cup \psi_2$, sendo k um índice que identifica uma variável não básica que por alteração do seu valor permite uma melhoria do valor da função objectivo. Então o δ será determinado por

$$\delta \equiv \begin{cases} 1, & \text{se } k \in \psi_1 \\ -1, & \text{se } k \in \psi_2 \end{cases}$$

O significado deste δ é indicar se a coordenada não básica k vai ser incrementada ao decrementada, como se constata no passo 3, seguinte, de actualização. Caso $\delta = 1$, a coordenada não básica k vai ser incrementada. Caso $\delta = -1$, a coordenada não básica k vai ser decrementada.

Passo 2 – Determinação do Passo

A coordenada não básica k vai ser actualizada de acordo com $x_k^N + \alpha \delta$ logo o máximo valor que pode tomar α é u_k^N . As coordenadas básicas são actualizadas de acordo com

$$\mathbf{x}^B - \mathbf{B}^{-1} \mathbf{N}(k) \alpha \delta$$

fazendo $\mathbf{y} = \mathbf{B}^{-1} \mathbf{N}(k) (\mathbf{x}^B - \alpha \delta \mathbf{y})$ tem que se verificar

$$0 \leq x_j^B - \alpha \delta y_j \leq u_j^B \quad \text{isto é} \quad 0 \leq x_j^B - \alpha \delta \sigma(y_j) |y_j| \leq u_j^B$$

Então se o $\sigma(y_j) = \delta$ o máximo valor que α pode tomar é dado por $\frac{x_j^B}{|y_j|}$, se

$\sigma(y_j) = -\delta$ o máximo valor que α pode tomar é dado por $\frac{u_j^B - x_j^B}{|y_j|}$.

Finalmente, há que escolher um α que esteja de acordo com o que se escreveu anteriormente. Pelo que o passo 2 consiste em calcular

$$\mathbf{y} = \mathbf{B}^{-1} \mathbf{N}(k)$$

$$\text{e } \Delta_1 = \min_{\substack{1 \leq j \leq m \\ \sigma(y_j) = \delta}} \left\{ \frac{x_j^B}{|y_j|}, \infty \right\}, \quad \Delta_2 = \min_{\substack{1 \leq j \leq m \\ -\sigma(y_j) = \delta}} \left\{ \frac{u_j^B - x_j^B}{|y_j|}, \infty \right\}.$$

Finalmente calcular o passo $\Delta = \min\{\Delta_1, \Delta_2, u_k^N\}$.

Passo 3 – Actualização

O valor da coordenada de \mathbf{x} correspondente à coluna k da matriz não básica é actualizado de acordo com

$$x_k^N \leftarrow x_k^N + \Delta \delta$$

e as coordenadas básicas são actualizadas de acordo com

$$\mathbf{x}^B \leftarrow \mathbf{x}^B - \Delta \delta \mathbf{y}.$$

Se $\Delta = u_k^N$, i.e., Δ igual ao valor máximo da coordenada não básica k o algoritmo prossegue para o *passo 1*. Caso contrário prossegue com o *passo 4*.

Passo 4 – Substituição e actualização da base

Consideremos os conjuntos

$$\psi_3 = \{j : \sigma(y_j) = \delta \text{ e } x_j^B = 0\} \text{ e } \psi_4 = \{j : -\sigma(y_j) = \delta \text{ e } x_j^B = u_j^B\}$$

Em que ψ_3 é o conjunto dos índices das variáveis básicas que tiveram decremento no seu valor no passo 3, tendo após esse passo valor nulo; e ψ_4 é o conjunto dos índices das variáveis básicas que tiveram incremento no seu valor no passo 3, tendo após esse passo valor máximo. Seleccionar um $i \in \psi_3 \cup \psi_4$, sendo i o índice de uma coluna da matriz \mathbf{B} , substituir na matriz \mathbf{B} e a coluna $\mathbf{B}(i)$ pela coluna $\mathbf{N}(k)$ e na matriz \mathbf{N} a coluna $\mathbf{N}(k)$ pela $\mathbf{B}(i)$. O algoritmo prossegue para o *passo 1*.

Se $\Delta > 0$ para cada iteração a convergência num número finito de iterações é garantida. Pois, o número de vértices do poliedro é finito, e em cada iteração selecciona-se um vértice a que corresponde um melhor valor da função objectivo.

Uma característica importante deste algoritmo é a de convergir para uma solução óptima mesmo que a hipótese de não degenerência não se verifique. A não verificação desta hipótese, só implica numa prova do algoritmo com maior elaboração matemática.

Se $\Delta = 0$, simplesmente deverá ser feita uma nova partição de **A** de modo a obter uma nova matriz **B** e continuar com aplicação do algoritmo. A nova matriz **B** corresponde ao mesmo ponto extremo, de modo a que nenhum progresso é feito no sentido de melhorar o valor da função objectivo.

A única dificuldade que poderá suceder é ao fim de algumas iterações, a mesma matriz **B** reaparecer com $\Delta = 0$. Neste caso o algoritmo repete as iterações, por isso, diz-se que o método entrou em ciclo (cycling). Felizmente, são raras as situações em que os problemas lineares reais entram em ciclo. Contudo, quando isso acontece, pequenas alterações no código evitam a entrada em ciclo. Algumas técnicas simples têm sido criadas para evitar a entrada em ciclo. Está fora do objectivo deste texto a abordagem destas técnicas.

Note-se que os dados necessários à implementação do algoritmo primal simplex são, os dados originais do problema de programação linear (i.e., **c**, **u**, **b** e **A**). Mais ainda, \mathbf{B}^{-1} é usado tanto na avaliação como na determinação do passo. Só é alterada se for necessário actualizar **B** no passo 4 do algoritmo. Para exemplificar o uso deste algoritmo considere os seguintes exemplos:

Exemplo 3.1

$$\begin{array}{ll}
 \max & x_1 + x_2 \\
 \text{s.a} & \\
 & x_1 + x_2 \geq 16 \\
 & x_1 - x_2 \geq 4 \\
 & 0 \leq x_1 \leq 10 \\
 & 0 \leq x_2 \leq 10.
 \end{array}$$

Será necessário escrever este problema na forma canónica, introduzindo duas variáveis de desvio uma para cada uma das restrições de desigualdade. Sejam estas variáveis indicadas por x_3 e x_4 os seus valores máximos tem que verificar respectivamente as desigualdades

$$\begin{array}{ll}
 \bar{x}_3 \geq \max & x_1 + x_2 - 16 \\
 \text{s.a} & \\
 & 0 \leq x_1 \leq 10 \\
 & 0 \leq x_2 \leq 10 & \bar{x}_3 \geq 4
 \end{array}$$

$$\begin{array}{ll}
 \bar{x}_4 \geq \max & x_1 - x_2 - 4 \\
 \text{s.a} & \\
 & 0 \leq x_1 \leq 10 \\
 & 0 \leq x_2 \leq 10 & \bar{x}_4 \geq 6
 \end{array}$$

O problema é equivalente ao seguinte

$$\begin{array}{ll}
 - \min & -(x_1 + x_2) \\
 \text{s.a} & x_1 + x_2 - x_3 = 16 \\
 & x_1 - x_2 - x_4 = 4 \\
 & 0 \leq x_1 \leq 10 \\
 & 0 \leq x_2 \leq 10 \\
 & 0 \leq x_3 \leq 4 \\
 & 0 \leq x_4 \leq 6.
 \end{array}$$

Será então necessário duas coordenadas básicas e duas não básicas.

Passo 0 – Iniciação

Consideremos por escolha o vector das coordenadas não básicas $x^N = [x_1 \ x_2]$,

então $x^B = [x_3 \ x_4]$ e a matriz \mathbf{A} é partida em

$$\mathbf{B} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{e} \quad \mathbf{N} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Portanto, $\mathbf{B}^{-1} = -\mathbf{I}$ e sejam feitas as tentativas:

Primeira tentativa, $[x_1 \ x_2] = [0 \ 0]$ logo $x^B = [x_3 \ x_4] = -[16 \ 4]$ que não é admissível, ponto A da Figura 3.4.

Segunda tentativa, $[x_1 \ x_2] = [0 \ 10]$ logo $x^B = [x_3 \ x_4] = [-6 \ -14]$ não é admissível, ponto B da Figura 3.4.

Terceira tentativa, $[x_1 \ x_2] = [10 \ 0]$ logo $x^B = [x_3 \ x_4] = [-6 \ 6]$ não é admissível, ponto C da Figura 3.4.

Quarta tentativa, $[x_1 \ x_2] = [10 \ 10]$ logo $x^B = [x_3 \ x_4] = [4 \ -4]$ não é admissível, ponto D da Figura 3.4.

Como se esgotou as tentativas para obter soluções básicas admissíveis com esta base, resta escolher uma nova base \mathbf{B} . Seja o vector das coordenadas não básicas

$x^N = [x_1 \ x_3]$, então $x^B = [x_2 \ x_4]$ e a matriz \mathbf{A} é partida em

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix} \quad \text{e} \quad \mathbf{N} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}$$

Portanto, $\mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix}$ e sejam feitas as tentativas:

Primeira tentativa, $[x_1 \ x_3] = [0 \ 0]$ logo $x^B = [x_2 \ x_4] = [16 \ -20]$ que não é admissível, ponto E da Figura 3.4.

Segunda tentativa, $[x_1 \ x_3] = [0 \ 4]$ logo $x^B = [x_2 \ x_4] = [20 \ -24]$ não é admissível, ponto F da Figura 3.4.

Terceira tentativa, $[x_1 \ x_3] = [10 \ 0]$, logo $x^B = [x_2 \ x_4] = [6 \ 0]$ é admissível, ponto G da Figura 3.4.

Pelo que o vector $[x^B | x^N]$ é uma solução básica admissível. Caso não fosse, havia que fazer outra escolha para as coordenadas não básicas ou escolher uma nova base **B**.

O algoritmo começa com a seguinte solução básica admissível $x^B = [x_2 \ x_4] = [6 \ 0]$, $x^N = [x_1 \ x_3] = [10 \ 0]$ a que corresponde a solução $[x_1 \ x_2 \ x_3 \ x_4] = [10 \ 6 \ 0 \ 0]$

Iteração 1

Passo 1 – Avaliação

Calcular o vector dos custos reduzidos

$$c^N - c^B B^{-1} N = [-1 \ 0] - [1 \ 0] \times \begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

logo $c^N - c^B B^{-1} N = [-2 \ 0]$. Pelo que $\psi_1 = \emptyset$ e $\psi_2 = \emptyset$, $\psi_1 \cup \psi_2 = \emptyset$, então o algoritmo termina e o ponto corrente é óptimo. No Quadro 3.1 seguinte é indicada a evolução do vector x , deste o passo 0 até à ultima iteração.

Quadro 3.1 – Evolução do vector x durante o processo iterativo

iteração	x_1	x_2	x_3	x_4
0	^N 10	^B 6	^N 0	^B 0
1	10	6	0	0

Na figura 3.4 é ilustrado o conjunto das variáveis admissíveis para este exemplo.

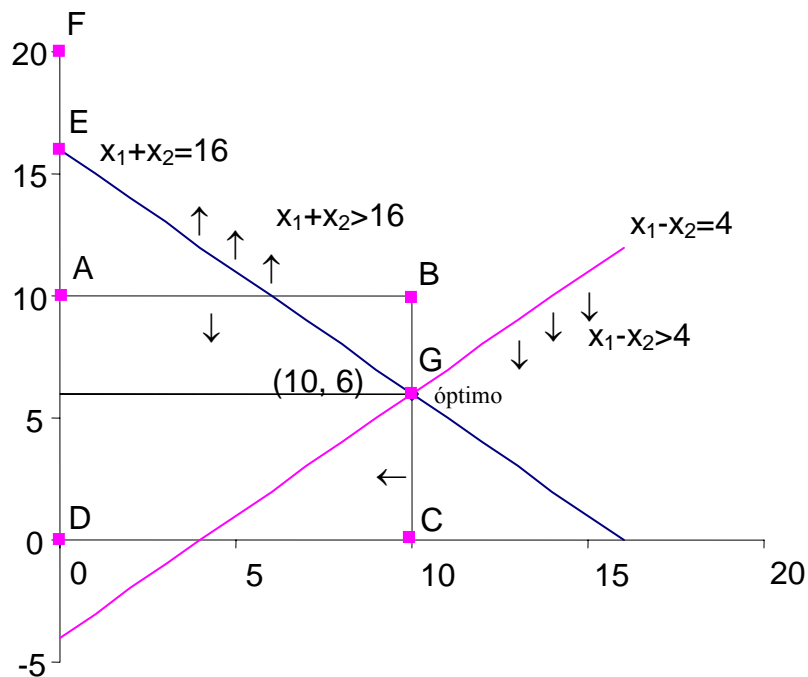


Figura 3.4 – Ilustração do conjunto das variáveis admissíveis.

O conjunto das variáveis admissíveis é singular, i.e., $Q = \{(10, 6)\}$. Consequentemente, justifica a dificuldade em encontrar a solução básica admissível inicial, visto que, como existem quatro variáveis, e duas restrições de igualdade, o número bases possíveis não é superior a $\binom{4}{2} = 6$, cada base está associada a 2^2 soluções básicas. Portanto, o número máximo de soluções básicas admissíveis não é superior 6×4 , mas só existe uma solução básica admissível para este problema.

Exemplo 3.2: Resolução gráfica.

$$\begin{aligned} \max \quad & x_1 + x_2 \\ \text{s.a} \quad & x_1 + x_2 \geq 16 \\ & x_1 - x_2 \geq -1 \end{aligned}$$

restrições de desigualdade:

$$0 \leq x_1 \leq 10, \quad 0 \leq x_2 \leq 10.$$

Na figura 3.5 é ilustrado o conjunto das variáveis admissíveis Q para este exemplo.

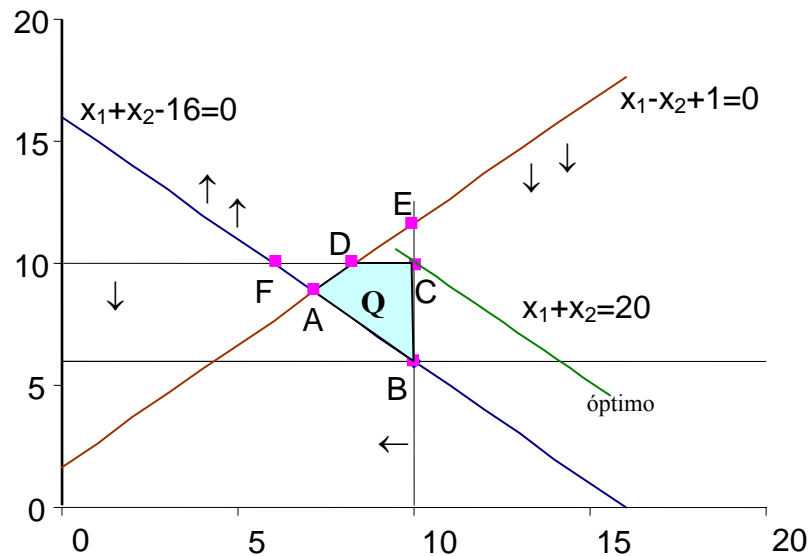


Figura 3.5 – Ilustração do conjunto das variáveis admissíveis.

Na figura 3.5, os pontos A, B, C, D, E, F são seis dos pontos que correspondem a soluções básicas para este problema. No entanto, só os quatro pontos A, B, C, D correspondem a soluções básicas admissíveis para este problema. Caso se escreva o problema na fórmula canónica, existem, como mostra o exemplo 3.1, quatro variáveis e duas restrições de igualdade, sendo o número máximo de soluções básicas de 24, mas só 4 são soluções básicas admissíveis. Observe que no exemplo 3.1 existia o mesmo número máximo de soluções básicas, mas só existia uma solução básica admissível, visto que, o conjunto das soluções admissíveis é singular.

A solução ótima para o exemplo 3.2 corresponde ao ponto solução básica admissível indicado por $C = (10, 10)$, sendo o valor ótimo para a função objectivo igual a 20, como se verifica pela observação da Figura 3.5.

Exemplo 3.3

$$\begin{aligned}
 \min \quad & x_1 + x_2 + 3x_3 + 10x_4 \\
 \text{s.a} \quad & x_1 + x_3 + x_4 = 5 \\
 & -x_1 + x_2 + x_5 = 0 \\
 & -x_2 - x_3 - x_4 = -5 \\
 & 0 \leq x_1 \leq 4 \\
 & 0 \leq x_2 \leq 2 \\
 & 0 \leq x_3 \leq 4 \\
 & 0 \leq x_4 \leq 10 \\
 & 0 \leq x_5 \leq 1 .
 \end{aligned}$$

Será então necessário ter três coordenadas básicas e duas não básicas.

Passo 0 – Iniciação

Consideremos por escolha o vector das coordenadas não básicas $\mathbf{x}^N = [x_2 \ x_3]$.

e seja $[x_1^N \ x_2^N] = [0 \ 4]$, então $\mathbf{x}^B = [x_1 \ x_4 \ x_5]$ e a matriz \mathbf{A} é particionada

em

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

sendo

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

o vector das coordenadas básicas é

$$\mathbf{x}^B = [x_1^B \ x_2^B \ x_3^B] = [0 \ 1 \ 0]$$

Pelo que o vector $[\mathbf{x}^B \ | \ \mathbf{x}^N]$ é uma solução básica admissível. Caso não fosse,

havia que fazer outra escolha para as coordenadas não básicas. O algoritmo

começa com a seguinte solução corrente

$$[x_1 \ x_2 \ x_3 \ x_4 \ x_5] = [0 \ 0 \ 4 \ 1 \ 0]$$

Iteração 1

Passo 1 – Avaliação

Calcular o vector dos custos reduzidos

$$\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N} = [1 \ 3] - [1 \ 10 \ 0] \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & -1 \end{bmatrix}$$

donde

$$\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N} = [-8 \ -7].$$

Pelo que $\psi_1 = \{1\}$ e $\psi_2 = \emptyset$ logo $\psi_1 \cup \psi_2 \neq \emptyset$, sendo $k=1$ como k pertence a ψ_1 é $\delta=1$.

Passo 2 – Atribuição do Passo

Determinar o vector \mathbf{y}

$$\mathbf{y} = \mathbf{B}^{-1} \mathbf{N}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}.$$

De onde resulta :

$$\Delta_1 = 1, \Delta_2 = \frac{(4-0)}{1} = 4 \text{ e } \Delta = \min\{\Delta_1, \Delta_2, u_1^N\} = \min\{1, 4, 2\} = 1.$$

Passo 3 – Actualização

Teremos:

$$x_1^N = x_2 = 0 + 1(1) = 1 \text{ e,}$$

$$\mathbf{x}^B = [x_1^B \ x_2^B \ x_3^B] = [0 \ 1 \ 0] - 1(1) \times [-1 \ 1 \ 0] = [1 \ 0 \ 0].$$

Como $\Delta = 1$ e $u_1^N = u_2 = 2$, logo $\Delta \neq u_1^N$, segue-se para o *passo 4*.

Passo 4 – *Substituição*

Determinar os conjuntos ψ_3 e ψ_4

$$\psi_3 = \{2\} \text{ e}$$

$$\psi_4 = \phi$$

Pelo que $\psi_3 \cup \psi_4 = \{2\}$

vem necessariamente $j = 2$ (uma vez que só há este valor, se houve-se vários escolhia-se um de entre eles). Então as colunas $\mathbf{B}(2)$ troca com a $\mathbf{N}(1)$, resultando as seguintes partições para o vector \mathbf{x}

$$\mathbf{x}^B = [x_1^B \quad x_2^B \quad x_3^B] = [x_1 \quad x_2 \quad x_5] \quad \mathbf{x}^N = [x_3 \quad x_4]$$

pelo que se tem

$$\mathbf{x}^N = [4 \quad 0] \quad \mathbf{x}^B = [1 \quad 1 \quad 0]$$

A partição para a matriz \mathbf{A}

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 1 \\ 0 & -1 & 0 \end{bmatrix} \text{ e} \quad \mathbf{N} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ -1 & -1 \end{bmatrix}$$

$$\text{sendo agora } \mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

E voltamos ao *passo 1*.

O vector \mathbf{x} da 1ª iteração é $[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5] = [1 \quad 1 \quad 4 \quad 0 \quad 0]$.

Iteração 2

Passo 1 – Avaliação

Calcular o vector dos custos reduzidos

$$\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N} = [3 \quad 10] - [1 \quad 1 \quad 0] \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ -1 & -1 \end{bmatrix} = [1 \quad 8]$$

obtemos $\psi_1 = \phi$ e $\psi_2 = \{1\}$, e a união $\psi_1 \cup \psi_2 \neq \phi$ logo $k = 1$.

Como k pertence a ψ_2 será $\delta = -1$.

Passo 2 – Atribuição do Passo

Determinar \mathbf{y}

$$\mathbf{y} = \mathbf{B}^{-1} \mathbf{N}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

de onde resulta

$$\Delta_1 = \infty, \Delta_2 = \min\{(4-1)/1, (2-1)/1\} = 1$$

$$\text{e } \Delta = \min\{\Delta_1, \Delta_2, u_1^N\} = \min\{\infty, 1, 4\} = 1.$$

Passo 3 – Actualização

Teremos:

$$x_1^N = x_3 = 4 + 1(-1) = 3$$

$$\mathbf{x}^B = [x_1 \quad x_2 \quad x_5] = [1 \quad 1 \quad 0] - 1(-1) \times [1 \quad 1 \quad 0] = [2 \quad 2 \quad 0].$$

Como $\Delta = 1$ e $u_1^N = u_3 = 4$ logo $\Delta \neq u_1^N$ segue-se para o passo 4.

Passo 4 – *Substituição*

Determinando os conjuntos ψ_3 e ψ_4 obtemos:

$$\psi_3 = \emptyset \text{ e}$$

$$\psi_4 = \{2\}$$

cuja união $\psi_3 \cup \psi_4 = \{2\}$

Pelo que $j = 2$. Fazemos agora a troca das colunas $\mathbf{B}(2)$ e $\mathbf{N}(1)$,

resultando assim as seguintes partições

$$\mathbf{x}^B = [x_1 \quad x_3 \quad x_5] = [2 \quad 3 \quad 0] \text{ e } \mathbf{x}^N = [x_2 \quad x_4] = [2 \quad 0]$$

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & -1 \end{bmatrix} \quad \text{sendo}$$

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

O vector \mathbf{x} na 2ª iteração é: $[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5] = [2 \quad 2 \quad 3 \quad 0 \quad 0]$

E voltamos ao *passo 1*.

Iteração 3

Passo 1 – *Avaliação*

$$\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N} = [1 \quad 10] - [1 \quad 3 \quad 0] \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -1 & -1 \end{bmatrix}$$

$$\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N} = [-1 \quad 7]$$

obtemos $\psi_1 = \phi$ e $\psi_2 = \phi$, e a união $\psi_1 \cup \psi_2 \neq \phi$, terminando o algoritmo e sendo a solução corrente

$$[x_1 \ x_2 \ x_3 \ x_4 \ x_5] = [2 \ 2 \ 3 \ 0 \ 0]$$

a óptima. O quadro seguinte apresenta a evolução do vector \mathbf{x} durante o processo iterativo.

Quadro 3.2 – Evolução do vector \mathbf{x} durante o processo iterativo

iteração	x_1	x_2	x_3	x_4	x_5
0	^B 0	^N 0	^N 4	^B 1	^B 0
1	1	1	4	0	0
passo 4	^B 1	^B 1	^N 4	^N 0	^B 0
2	2	2	3	0	0
passo 4	^B 2	^N 2	^B 3	^N 0	^B 0

3.4- Condições de Karush-Kuhn-Tucker para a programação linear

Este subcapítulo apresenta um conjunto de condições para o problema de programação linear, conhecidas como condições de Karush-Kuhn-Tucker. Estas condições são condições necessárias para a identificação do óptimo, mas para este problema, visto que, a função objectivo é uma função linear e o conjunto das variáveis de decisão admissíveis é convexo são também suficientes para a identificação de uma solução óptima. Estas condições são as apresentadas na proposição seguinte.

Proposição 3.9

Seja \mathbf{x}^* tal que $\mathbf{A}\mathbf{x}^* = \mathbf{b}$ e $\mathbf{0} \leq \mathbf{x}^* \leq \mathbf{u}$. Então \mathbf{x}^* é um óptimo para o problema de programação linear sse só se existirem vectores multiplicadores de lagrange $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, e $\boldsymbol{\lambda}$ tais que: $\boldsymbol{\pi}\mathbf{A} - \mathbf{c} = \boldsymbol{\mu} - \boldsymbol{\lambda}$, $(\mathbf{x}^* - \mathbf{u})\boldsymbol{\mu} = 0$, $\mathbf{x}^*\boldsymbol{\lambda} = 0$ e $\boldsymbol{\mu} \geq \mathbf{0}$, e $\boldsymbol{\lambda} \geq \mathbf{0}$.

Prova: Considere o problema de programação linear escrito como um problema de maximização, como se segue

$$\begin{array}{ll} \max & -\mathbf{c}\mathbf{x} \quad \equiv \quad -\min \mathbf{c}\mathbf{x} \\ \text{s.a.} & \text{s.a.} \\ & \mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0} \quad \mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0} \\ & \mathbf{u} - \mathbf{x} \geq \mathbf{0} \quad \mathbf{u} - \mathbf{x} \geq \mathbf{0} \\ & \mathbf{x} \geq \mathbf{0} \quad \mathbf{x} \geq \mathbf{0} \end{array}$$

A função de Lagrange será escrita como

$$L(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = -\mathbf{c}\mathbf{x} + \boldsymbol{\pi}(\mathbf{A}\mathbf{x} - \mathbf{b}) + \boldsymbol{\mu}(\mathbf{u} - \mathbf{x}) + \boldsymbol{\lambda}\mathbf{x} \quad (3.13)$$

As condições de acordo com o teorema de Karush-Kuhn-Tucker são:

$$(K_K_T.1) \quad \mathbf{A}\mathbf{x}^* = \mathbf{b} \text{ e } \mathbf{0} \leq \mathbf{x}^* \leq \mathbf{u}$$

$$(K_K_T.2) \quad \exists \boldsymbol{\mu}, \boldsymbol{\lambda} \geq \mathbf{0} \text{ tal que}$$

$$\begin{aligned} \boldsymbol{\mu}(\mathbf{u} - \mathbf{x}^*) &= \mathbf{0} \\ \boldsymbol{\lambda}\mathbf{x}^* &= \mathbf{0} \end{aligned}$$

$$(K_K_T.3) \quad -\mathbf{c} + \boldsymbol{\pi}\mathbf{A} - \boldsymbol{\mu} + \boldsymbol{\lambda} = \mathbf{0}$$

o que prova o teorema \blacklozenge .

Ou seja resolver o sistema de Karush-Kuhn-Tucker:

$$\begin{array}{ll} \mathbf{A}\mathbf{x} = \mathbf{b} & (\mathbf{x} - \mathbf{u})\boldsymbol{\mu} = \mathbf{0} \\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} & \mathbf{x}\boldsymbol{\lambda} = \mathbf{0} \\ \boldsymbol{\pi}\mathbf{A} - \mathbf{c} = \boldsymbol{\mu} - \boldsymbol{\lambda} & \boldsymbol{\mu} \geq \mathbf{0}, \boldsymbol{\lambda} \geq \mathbf{0} \end{array}$$

em ordem a \mathbf{x} , $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, e $\boldsymbol{\lambda}$ é matematicamente equivalente a resolver o problema PL. Este sistema será usado no estudo da programação em redes, para obter o algoritmo da condição.

Exemplo 3.5: Considere o problema do exemplo 3.4, verifique usando o sistema de Karush-Kuhn-Tucker que o ponto óptimo obtido por resolução gráfica é óptimo. O problema será escrito como

$$\begin{aligned} \min \quad & -(x_1 + x_2) \\ \text{s.a} \quad & x_1 + x_2 - x_3 = 16 \\ & x_1 - x_2 - x_4 = -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \\ & 0 \leq x_3 \leq 4 \\ & 0 \leq x_4 \leq 11. \end{aligned}$$

Cuja solução óptima foi obtida na Figura 3.3 $[x_1 \ x_2 \ x_3 \ x_4] = [10 \ 10 \ 4 \ 1]$. Caro que as condições de admissibilidade, K_K_T.1: $\mathbf{Ax}^* = \mathbf{b}$ e $\mathbf{0} \leq \mathbf{x}^* \leq \mathbf{u}$, são satisfeitas. As condições K_K_T.2 obrigam a que $\lambda = \mathbf{0}$ e que $\mu_4 = 0$. Portanto, basta determinar

$$\mu_1, \mu_2, \mu_3 \geq 0 \text{ e } \pi_1, \pi_2 \in \mathbb{R}$$

tais que $\pi \mathbf{A} - \mu = \mathbf{c}$, logo $\pi_2 = 0$, $\pi_1 - \mu_1 = -1$, $\pi_1 - \mu_2 = -1$ e $-\pi_1 - \mu_3 = 0$. Pelo que tem que ser $\pi_1 + 1 \geq 0$, $\mu_1 = \mu_2 = \pi_1 + 1$ e $-\pi_1 = \mu_3 \geq 0$. Consequentemente, são solução do sistema de Karush-Kuhn-Tucker os seguintes vectores

$$[x_1 \ x_2 \ x_3 \ x_4] = [10 \ 10 \ 4 \ 1]$$

$$[\pi_1 \ \pi_2] = [0 \ 0]$$

$$[\mu_1 \ \mu_2 \ \mu_3 \ \mu_4] = [1 \ 1 \ 0 \ 0]$$

$$[\lambda_1 \ \lambda_2 \ \lambda_3 \ \lambda_4] = [0 \ 0 \ 0 \ 0].$$

Observe que a condição necessária para ter uma solução do sistema de Karush-Kuhn-Tucker é escolher um π_1 tal que $-1 \leq \pi_1 \leq 0$. Outra solução possível seria fazer $\pi_1 = -1$, $\pi_2 = 0$, $[\mu_1 \ \mu_2 \ \mu_3 \ \mu_4] = [0 \ 0 \ 1 \ 0]$.

3.5- Teoria da dualidade

Associado ao problema de programação linear, que no contexto da teoria da dualidade também é dito de problema primal, existe o problema dual. Considere a função de lagrange (3.11), o subproblema dual será descrito pelo seguinte problema sem restrições

$$\max L(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda})$$

sendo $L(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = -(\mathbf{c} + \boldsymbol{\pi}\mathbf{A} - \boldsymbol{\mu} + \boldsymbol{\lambda})\mathbf{x} - \boldsymbol{\pi}\mathbf{b} + \boldsymbol{\mu}\mathbf{u}$

e $\boldsymbol{\mu}, \boldsymbol{\lambda} \geq \mathbf{0}$.

A solução do subproblema dual tem que satisfazer à condição $\nabla L(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \mathbf{0}$, i.e., $-\mathbf{c} + \boldsymbol{\pi}\mathbf{A} - \boldsymbol{\mu} + \boldsymbol{\lambda} = \mathbf{0} \equiv \boldsymbol{\pi}\mathbf{A} - \boldsymbol{\mu} = \mathbf{c} - \boldsymbol{\lambda}$. Portanto, o problema dual é escrito como

$$\begin{aligned} \min & -(\boldsymbol{\pi}\mathbf{b} - \boldsymbol{\mu}\mathbf{u}) \\ \text{s.a.} & \\ & \boldsymbol{\pi}\mathbf{A} - \boldsymbol{\mu} = \mathbf{c} - \boldsymbol{\lambda} \\ & \boldsymbol{\mu}, \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned}$$

O problema dual tem algumas propriedades muito úteis sob o ponto de vista da programação em redes. O problema primal e o seu dual podem ser escritos como

$$\left. \begin{array}{l} \min \quad \mathbf{c} \mathbf{x} \\ \text{s.a} \quad \mathbf{A} \mathbf{x} = \mathbf{b} \\ \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \end{array} \right\} \begin{array}{l} \text{PL} \\ \text{(Primal)} \end{array} \quad \left. \begin{array}{l} \text{máx} \quad \mathbf{b} \boldsymbol{\pi} - \mathbf{u} \boldsymbol{\mu} \\ \text{s.a} \quad \boldsymbol{\pi}\mathbf{A} - \boldsymbol{\mu} \leq \mathbf{c} \\ \quad \boldsymbol{\mu} \geq \mathbf{0} \end{array} \right\} \begin{array}{l} \text{PLD} \\ \text{(Dual)} \end{array}$$

No contexto da teoria da dualidade, \mathbf{x} é o vector das variáveis primais e $(\boldsymbol{\pi}, \boldsymbol{\mu})$ é o vector das variáveis duais. A hipótese de o domínio das soluções possíveis para a PL não ser vazio implica que os problemas são equivalentes, isto é, o valor óptimo do problema primal é igual ao valor óptimo do problema dual. Esta equivalência é traduzida pelas seguintes proposições.

Preposição 3.10

Seja \mathbf{x}^* uma solução óptima para o PL e seja $(\boldsymbol{\pi}^*, \boldsymbol{\mu}^*)$ uma solução óptima para o PLD. Então $\mathbf{c} \mathbf{x}^* = \mathbf{b} \boldsymbol{\pi}^* - \mathbf{u} \boldsymbol{\mu}^*$.

Preposição 3.11

Seja \mathbf{x}^* solução óptima para o PL, \mathbf{B} a matriz básica e \mathbf{c}^B o vector cujas componentes são as componentes de \mathbf{c} correspondentes ás coordenadas básicas.

Sejam $\boldsymbol{\pi}^* = \mathbf{c}^B \mathbf{B}^{-1}$ e $\mu_j^* = \begin{cases} \boldsymbol{\pi}^* \mathbf{A}(j) - c_j & \text{se } x_j^* = u_j \\ 0 & \text{c. c.} \end{cases}$, então $(\boldsymbol{\pi}^*, \boldsymbol{\mu}^*)$ é

solução óptima para o PLD.

Prova: Para provar este teorema, simplificando o formalismo, será admitido que a solução óptima do problema primal é não degenerada, hipótese de não degenerência, i.e., $\mathbf{0} < \mathbf{x}^B < \mathbf{u}^B$. Assim, sendo $\boldsymbol{\mu}^B, \boldsymbol{\lambda}^B = \mathbf{0}$, então como $\boldsymbol{\pi} [\mathbf{B} \mid \mathbf{N}] - [\boldsymbol{\mu}^B \mid \boldsymbol{\mu}^N] = [\mathbf{c}^B \mid \mathbf{c}^N] - [\boldsymbol{\lambda}^B \mid \boldsymbol{\lambda}^N]$ pela condição a que o subproblema dual tem que satisfazer, será $\boldsymbol{\pi} \mathbf{B} = \mathbf{c}^B$ e $\boldsymbol{\pi} \mathbf{N} - \boldsymbol{\mu}^N = \mathbf{c}^N - \boldsymbol{\lambda}^N$. Portanto, $\boldsymbol{\pi}^* = \mathbf{c}^B \mathbf{B}^{-1}$ e como pela condição anterior (K_K_T.2) μ_i^N e λ_i^N não podem ser ambos diferentes de zero, facilmente se tem o resultado pretendido \blacklozenge .

A relação entre o problema primal e o respectivo dual obedece ainda ao teorema da complementaridade. Este teorema significa que a uma restrição não activa do problema primal, corresponde uma variável dual nula; e a uma restrição não activa do problema dual, corresponde uma variável primal nula. O teorema da complementaridade resulta directamente da aplicação das condições de Karush-Kuhn-Tucker ao problema primal ou ao problema dual.

Exemplo 3.6: Escreva, usando o formalismo desenvolvido, o problema dual do seguinte problema primal

$$\begin{aligned} \min \quad & x_1 \\ \text{s.a} \quad & x_1 + x_2 = 16 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10. \end{aligned}$$

como $[c_1 \ c_2] = [1 \ 0]$, $\mathbf{A} = [1 \ 1]$ e $[u_1 \ u_2] = [10 \ 10]$ então será

$$\begin{aligned} \max \quad & 16\pi - 10(\mu_1 + \mu_2) \\ \text{s.a} \quad & \pi [1 \ 1] - [\mu_1 \ \mu_2] \leq [1 \ 0] \\ & [\mu_1 \ \mu_2] \geq [0 \ 0] \end{aligned} \quad \text{ou seja} \quad \begin{aligned} \max \quad & 16\pi - 10(\mu_1 + \mu_2) \\ \text{s.a} \quad & \pi - \mu_1 \leq 1 \\ & \pi - \mu_2 \leq 0 \\ & \mu_1 \geq 0 \\ & \mu_2 \geq 0 \end{aligned}$$

Exemplo 3.7: Determine, usando a teoria da dualidade, o problema dual do seguinte problema primal

$$\begin{aligned} \max \quad & 16\pi - 10(\mu_1 + \mu_2) \\ \text{s.a} \quad & 1 - \pi + \mu_1 \geq 0 \\ & \mu_2 - \pi \geq 0 \\ & \mu_1 \geq 0 \\ & \mu_2 \geq 0 \end{aligned}$$

Sejam os multiplicadores de lagrange para as restrições indicados respectivamente por $x_1, x_2, \delta_1, \delta_2$, então a função de lagrange será

$$\begin{aligned} L(\pi, \mu_1, \mu_2; x_1, x_2, \delta_1, \delta_2) = & 16\pi - 10(\mu_1 + \mu_2) + \\ & + x_1(1 - \pi + \mu_1) + \\ & + x_2(\mu_2 - \pi) + \\ & + \delta_1 \mu_1 + \\ & + \delta_2 \mu_2 \end{aligned}$$

O subproblema dual será $\max L(\pi, \mu_1, \mu_2; x_1, x_2, \delta_1, \delta_2)$. A solução do subproblema dual tem que satisfazer à condição $\nabla L(\pi, \mu_1, \mu_2; x_1, x_2, \delta_1, \delta_2) = 0$. Pelo que será

$$\begin{cases} 16 - x_1 - x_2 = 0 \\ -10 + x_1 + \delta_1 = 0 \\ -10 + x_2 + \delta_2 = 0 \end{cases} \quad \text{e sendo } x_1, x_2, \delta_1, \delta_2 \geq 0 \text{ o que é equivalente às restrições}$$

$x_1 + x_2 = 16$, $0 \leq x_1 \leq 10$, $0 \leq x_2 \leq 10$. A função lagrange pode ser escrita como

$$L(\pi, \mu_1, \mu_2; x_1, x_2, \delta_1, \delta_2) = (16 - x_1 - x_2)\pi + (-10 + x_1 + \delta_1)\mu_1 + (-10 + x_2 + \delta_2)\mu_2 + x_1$$

então a função dual será $L_*(x_1, x_2, \delta_1, \delta_2) = x_1$. Pelo que o problema dual é

$$\begin{array}{ll} \min & x_1 \\ \text{s.a} & x_1 + x_2 = 16 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \end{array}$$

Este exemplo particular confirma a afirmação que o problema dual do problema dual é o problema primal.

Exemplo 3.9: Considere o problema do exemplo 3.4, usando a proposição 3.10 determine as variáveis duais ótimas.

Exemplo 3.8: Mostre que o problema dual do problema dual do problema de programação linear é o problema de programação linear.

Seja o problema primal o PLD

$$\begin{array}{ll} \text{máx} & \boldsymbol{\pi b} - \boldsymbol{\mu u} \\ \text{s.a} & \mathbf{c} - \boldsymbol{\pi A} + \boldsymbol{\mu} \geq \mathbf{0} \\ & \boldsymbol{\mu} \geq \mathbf{0} \end{array}$$

a função de lagrange é $L(\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{x}; \boldsymbol{\delta}) = \boldsymbol{\pi}(\mathbf{b} - \mathbf{Ax}) + \boldsymbol{\mu}(-\mathbf{u} + \mathbf{x} + \boldsymbol{\delta}) + \mathbf{cx}$ resolvendo o subproblema dual tem-se $\mathbf{b} - \mathbf{Ax} = \mathbf{0}$ e $-\mathbf{u} + \mathbf{x} + \boldsymbol{\delta} = \mathbf{0}$, sendo $\mathbf{x}, \boldsymbol{\delta} \geq \mathbf{0}$ e a função dual $L^*(\mathbf{x}; \boldsymbol{\delta}) = \mathbf{cx}$, então o problema dual é o problema de programação linear.

Exemplo 3.9: Considere o problema do exemplo 3.4, usando a proposição 3.10 determine as variáveis duais ótimas

A solução ótima é $[x_1 \ x_2 \ x_3 \ x_4] = [10 \ 10 \ 4 \ 1]$. A última coordenada tem que ser necessariamente não básica e seja considerada a penúltima também básica, embora seja degenerada. Portanto, $\mathbf{x}^B = [x_3 \ x_4] = [4 \ 1]$ e $\mathbf{x}^N = [x_1 \ x_2] = [10 \ 10]$.

Consequentemente

$$\mathbf{B} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \text{ e } \mathbf{N} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

pelo que $\boldsymbol{\pi}^* = -\mathbf{c}^B = \mathbf{0}$, sendo $\mu_j^* = -c_j$ se $x_j^* = u_j$, caso contrário $\mu_j^* = 0$, i.e.,

$$[\pi_1^* \ \pi_2^*] = [0 \ 0],$$

$$[\mu_1^* \ \mu_2^* \ \mu_3^* \ \mu_4^*] = [1 \ 1 \ 0 \ 0].$$

Exercícios do Capítulo 3

1. Considere os seguintes problemas de programação linear

$$\begin{aligned} \min \quad & -10 + x \\ \text{s.a} \quad & 10 + x \leq 16 \\ & 10 - x \geq -1 \\ & 0 \leq x \leq 10 \end{aligned}$$

$$\begin{aligned} \min \quad & -10 + \alpha x \\ \text{s.a} \quad & 10/\alpha + x \leq 16/\alpha \\ & 10/\alpha - x \geq -1/\alpha \\ & 0 \leq x \leq 30 \text{ com } \alpha \in \mathbb{R}^+ \setminus \{0\}. \end{aligned}$$

Resolva os problemas pelo algoritmo do simplex.

Solução: $x = 0$

$x = 0$.

Observe que o segundo problema é equivalente ao primeiro, basta fazer a seguinte transformação de variável $x = x' / \alpha$.

2. Considere os seguintes problemas de programação linear

$$\begin{aligned} \min \quad & -x_1 + x_2 \\ \text{s.a} \quad & x_1 + x_2 \leq 16 \\ & x_1 - x_2 \geq -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \end{aligned}$$

$$\begin{aligned} \min \quad & -x_1 + \alpha x_2 \\ \text{s.a} \quad & x_1 + x_2 \leq 16 \\ & x_1 - x_2 \geq -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \text{ com } \alpha \in \mathbb{R}^+ \setminus \{0\}. \end{aligned}$$

Resolva os problemas pelo algoritmo do simplex. Compare a solução destes problemas com os anteriores, justificando os resultados obtidos.

Solução: $\mathbf{x} = [10 \ 0]$

$\mathbf{x} = [10 \ 0]$.

Como $0 \leq x_1 \leq 10$ e $c_1 = -1$, então $x_1 = 10$ é admissível e o menor valor da função objectivo é obtido com este valor, fixando $x_1 = 10$ nos problemas anteriores derivam os problemas indicados em 1.

3. Considere os seguintes problemas de programação linear

$$\begin{aligned} \min \quad & -x_1 - x_2 \\ \text{s.a} \quad & x_1 + x_2 \leq 16 \\ & x_1 - x_2 \geq -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \end{aligned}$$

$$\begin{aligned} \min \quad & x_1 - x_2 \\ \text{s.a} \quad & x_1 + x_2 \leq 16 \\ & x_1 - x_2 \geq -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10. \end{aligned}$$

Resolva os problemas pelo algoritmo do simplex.

Solução: $\mathbf{x} = [10 \ 6]$

$\mathbf{x} = [0 \ 1]$.

4. Considere os seguintes problemas de programação linear

$$\begin{aligned} \min \quad & -x_1 + x_2 + x_3 + x_4 \\ \text{s.a} \quad & x_1 + x_2 - x_3 \leq 16 \\ & x_1 - x_2 + x_4 \geq -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \\ & 0 \leq x_3 \leq 4 \\ & 0 \leq x_4 \leq 11 \end{aligned}$$

$$\begin{aligned} \min \quad & -x_1 + x_2 + x_3 + x_4 \\ \text{s.a} \quad & x_1 + x_2 - x_3 \leq 16 \\ & x_1 - x_2 - x_4 \geq -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \\ & 0 \leq x_3 \leq 4 \\ & 0 \leq x_4 \leq 11. \end{aligned}$$

Escrevas os problemas na forma canónica.

Solução:

$$\begin{aligned} \min \quad & -x_1 + x_2 + x_3 + x_4 \\ \text{s.a} \quad & x_1 + x_2 - x_3 + x_5 = 16 \\ & x_1 - x_2 + x_4 - x_6 = -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \\ & 0 \leq x_3 \leq 4 \\ & 0 \leq x_4 \leq 11 \\ & 0 \leq x_5 \leq 20 \\ & 0 \leq x_6 \leq 22 \end{aligned}$$

$$\begin{aligned} \min \quad & -x_1 + x_2 + x_3 + x_4 \\ \text{s.a} \quad & x_1 + x_2 - x_3 + x_5 = 16 \\ & x_1 - x_2 - x_4 - x_6 = -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \\ & 0 \leq x_3 \leq 4 \\ & 0 \leq x_4 \leq 11 \\ & 0 \leq x_5 \leq 20 \\ & 0 \leq x_6 \leq 22. \end{aligned}$$

5. Considere os problemas de programação linear anteriores, resolva os problemas pelo algoritmo do simplex.

$$\text{Solução: } \mathbf{x} = [10 \ 0 \ 0 \ 0 \ 6 \ 11]$$

$$\mathbf{x} = [10 \ 0 \ 0 \ 0 \ 6 \ 11].$$

Observe que: no primeiro problema, na segunda restrição é adicionado x_4 ; no segundo, é subtraído x_4 . Como $x_4 = 0$ no ótimo do primeiro problema, as soluções são iguais, visto que, somar ou subtrair um valor nulo é indiferente.

6. Considere os problemas de programação linear anteriores, escreva os problemas duais respectivos.

Solução: Problemas na forma canônica

$$\begin{aligned} \min \quad & -x_1 - x_2 \\ \text{s.a} \quad & x_1 + x_2 + x_3 = 16 \\ & x_1 - x_2 - x_4 = -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \\ & 0 \leq x_3 \leq 4 \\ & 0 \leq x_4 \leq 11 \end{aligned}$$

$$\begin{aligned} \min \quad & x_1 - x_2 \\ \text{s.a} \quad & x_1 + x_2 + x_3 = 16 \\ & x_1 - x_2 - x_4 = -1 \\ & 0 \leq x_1 \leq 10 \\ & 0 \leq x_2 \leq 10 \\ & 0 \leq x_3 \leq 4 \\ & 0 \leq x_4 \leq 11. \end{aligned}$$

Problemas duais

$$\begin{aligned} \max \quad & 16\pi_1 - \pi_2 - 10\mu_1 - 10\mu_1 - 4\mu_1 - 14\mu_2 \\ \text{s.a} \quad & -1 - \pi_1 + \pi_2 + \mu_1 \geq 0 \\ & -1 - \pi_1 - \pi_2 + \mu_2 \geq 0 \\ & -\pi_1 + \mu_3 \geq 0 \\ & \pi_2 + \mu_4 \geq 0 \\ & \mu_1, \mu_2, \mu_3, \mu_4 \geq 0 \\ \max \quad & 16\pi_1 - \pi_2 - 10\mu_1 - 10\mu_1 - 4\mu_1 - 14\mu_2 \\ \text{s.a} \quad & 1 - \pi_1 + \pi_2 + \mu_1 \geq 0 \\ & -1 - \pi_1 - \pi_2 + \mu_2 \geq 0 \\ & -\pi_1 + \mu_3 \geq 0 \\ & \pi_2 + \mu_4 \geq 0 \\ & \mu_1, \mu_2, \mu_3, \mu_4 \geq 0 \end{aligned}$$

7. Considere o problema de programação linear

$$\begin{aligned} \min \quad & x_1 \\ \text{s.a} \quad & x_1 - x_2 = 1 \\ & 0 \leq x_1, x_2 \leq 10. \end{aligned}$$

Resolver o problema dual pelo algoritmo do simplex.

Solução: Problema dual

$$\begin{aligned} \max \quad & \pi - 10(\mu_1 + \mu_2) \\ \text{s.a} \quad & 1 - \pi + \mu_1 \geq 0 \\ & \pi_1 + \mu_2 \geq 0 \\ & \mu_1, \mu_2 \geq 0 \end{aligned} \quad \begin{array}{l} \text{óptimo} \\ \pi = 1 \\ \mu_1 = 0 \\ \mu_2 = 0 \end{array}$$

8. Considere o problema de programação linear anterior determine a sua solução usando a proposição 3.10.

Solução: $\mathbf{x} = [1 \ 0]$

9. Considere o problema de programação linear 7. e o seu dual obtido em 8., mostre de pelo sistema de Karush-Kuhn-Tucker, que as soluções obtidas para estes problemas são as ótimas.

4- MÉTODO SIMPLEX PARA PROGRAMAÇÃO EM REDES

Este capítulo apresenta uma especialização sobre o algoritmo primal simplex para o problema de programação linear em rede. Esta especialização, quando usada permite que o método primal simplex possa ser resolvido directamente sobre a rede do problema. Esta especialização é também conhecido por *algoritmo gráfico do simplex*.

Neste capítulo, exclusivamente para as demonstrações considere que a rede não tem arcos que partem de um nó para o mesmo nó, ditos laços, nem tem arcos múltiplos em paralelo. Esta hipótese é apenas uma conveniência de exposição, a não condiciona os resultados que serão apresentados nem a aplicabilidade do algoritmo, visto que, se há um laço no nó i pode ser substituído por dois arcos, (i, k) e (k, i) onde k é um novo nó, se há dois arcos orientados do nó i para o nó j , um deles pode ser substituído por dois arcos, (i, k) e (k, j) onde k é um novo nó. Por exemplo, para forçar a verificação desta consideração, na rede da Figura 2.1 o arco $(1, 2)$ é substituído pelos arcos $(1, 5)$ e $(5, 2)$ e é introduzido um novo nó número 5 como ilustra a Figura 4.1.

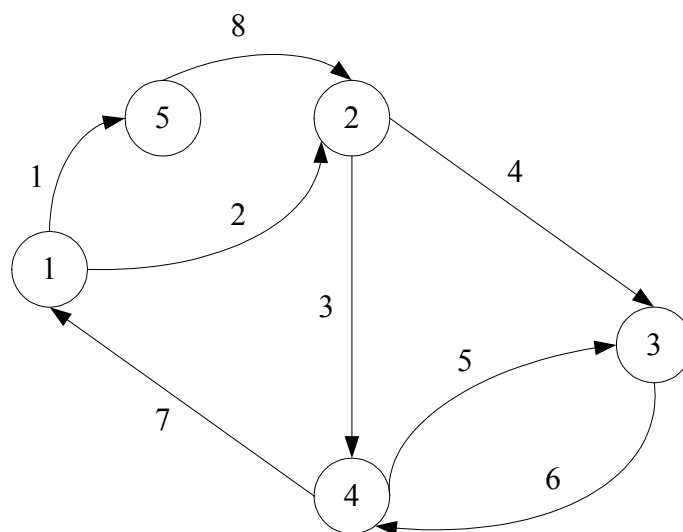


Figura 4.1 – Rede que resulta da rede da Figura 2.1, mas sem arcos em paralelo.

A correspondente matriz de incidências dos nós nos arcos passa a ser a seguinte:

$$\mathbf{A} = \begin{array}{c} \text{arços} \\ \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \left[\begin{array}{cccccccc} 1 & 1 & & & & & -1 & \\ & -1 & 1 & 1 & & & & -1 \\ & & & -1 & -1 & 1 & & \\ & & -1 & & 1 & -1 & 1 & \\ -1 & & & & & & & 1 \end{array} \right] \begin{array}{c} \text{nós} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \end{array}$$

Note que as colunas de \mathbf{A} são distintas. Portanto, existe uma correspondência biunívoca entre aos arcos da rede e as colunas da matriz \mathbf{A} .

4.1- Resultados da teoria dos grafos

Este subcapítulo apresenta os resultados da teoria dos grafos que vão ser utilizados na especialização do algoritmo primal simplex para a programação linear em rede. Considere uma grafo com \bar{I} nós e \bar{J} arcos, ordenando respectivamente de $1, \dots, \bar{I}$, e de $1, \dots, \bar{J}$. Este grafo pode ser descrito por dois conjuntos o conjunto dos nós N e o conjunto dos arcos A . Um grafo G fica descrito pelo par ordenado $[N, A]$.

Seja \mathbf{A} a matriz das incidências dos nós nos arcos associada à rede. Para cada arco j , identificam-se os nós de partida e de chegada do arco através da função nó *de* onde provém o arco $d: A \rightarrow N$

$$\text{se } \mathbf{A}_{ij}=1, \quad d(j)=i$$

e da função nó para onde vai o arco $T: A \rightarrow N$

$$\text{se } \mathbf{A}_{kj}=-1, \quad d(j)=k.$$

Então o arco j é descrito como o par ordenado de nós $(F(j), T(j))$, i.e., $A \subset N^2$. As funções $F(j)$ e $T(j)$ para a rede da Figura 4.1 são apresentadas no Quadro 4.1.

Quadro 4.1 – Funções “de” e “para” grafo da Figura 4.1

Número do Arco j	“De” Nó $d(j)$	“Para” Nó $p(j)$
1	1	5
2	1	2
3	2	4
4	2	3
5	4	3
6	3	4
7	4	1
8	5	2

Um grafo é dito de próprio se tiver um número de nós, $\#N$, superior ou igual a 2 e um número de arcos $\#A$ superiores ou iguais a um. Dado um grafo $G=[N,A]$, $G'=[N',A']$ é um subgrafo de G se $N' \subseteq N$ e $A' \subseteq A$. Um subgrafo $G'=[N',A']$ de $G=[N,A]$ é um subgrafo gerador de G se $N' = N$. Esta definição é ilustrada na Figura 4.2.

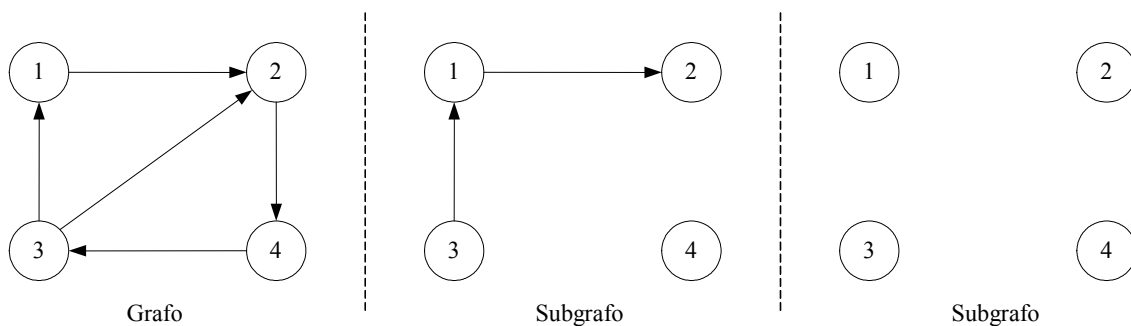


Figura 4.2 – Ilustração de um grafo e subgrafos geradores do grafo.

Note que qualquer grafo G é um subgrafo do grafo G e ainda mais é um subgrafo gerador de G . Ainda, note que um subgrafo gerador do grafo G pode não conter nenhum arco de G .

Uma sequência finita de nós s_j e arcos $e_{j(i)} \in \{(s_i, s_{i+1}), (s_{i+1}, s_i)\}$ de um grafo G escrita como

$$P = \{s_1, e_{j(1)}, s_2, e_{j(2)}, s_3, e_{j(3)}, \dots, s_n, e_{j(n)}, s_{n+1}\}$$

tendo pelo menos um arco, define um trajecto de s_1 para s_{n+1} . Um trajecto é um caminho no grafo G , se os elementos ímpares são nós distintos de G , os elementos pares são arcos de G , e cada arco

$$e_{j(i)} \in \{(s_i, s_{i+1}), (s_{i+1}, s_i)\}.$$

Um caminho é o percurso que liga o primeiro ao último elemento da sequência que o define. No grafo da Figura 4.1, por exemplo, P é um caminho

$$P = \{5, e_1, 1, e_7, 4, e_5, 3\} = \{5, (1, 5), 1, (4, 1), 4, (4, 3), 3\}$$

cuja representação se ilustra na Figura 4.3.

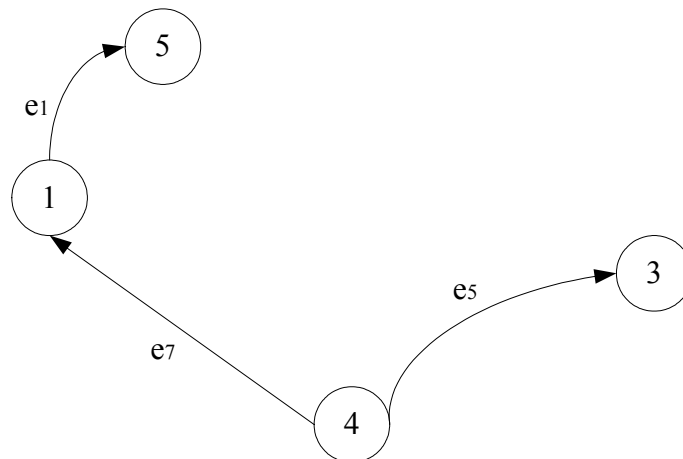


Figura 4.3 – Ilustração do caminho P .

Proposição 4.1:

Todos os arcos de um caminho são distintos.

Prova: Se dois arcos de um caminho forem idênticos, os nós desse caminho não podem ser distintos, portanto, não satisfaz a definição. ♦

Um ciclo num grafo G é um trajecto que começa num nó e acaba no mesmo nó, sendo indicado pela sequência $C = \{s_1, e_{j(1)}, s_2, e_{j(2)}, \dots, s_n, e_{j(n)}, s_{n+1}\}$ tendo pelo menos dois arcos. Um ciclo é simples num grafo G se a subsequência $\{s_1, e_{j(1)}, s_2, e_{j(2)}, s_3, e_{j(3)}, \dots, s_n\}$ é um caminho em G e $e_{j(n)} \in \{(s_n, s_{n+1}), (s_{n+1}, s_n)\}$, sendo $s_n \neq s_{n+1}$ e $e_{j(n)} \neq e_{j(1)}$. Um exemplo ilustrativo de um ciclo simples no grafo ilustrado na Figura 4.1 é

$$C = \{2, e_8, 5, e_1, 1, e_2, 2\}.$$

Proposição 4.2:

Os arcos de um ciclo simples são distintos.

Prova: Suponhamos um ciclo com n arcos, visto que, a subsequência

$$\{s_1, e_{j1}, s_2, e_{j2}, \dots, s_{n-1}, e_{jn-1}, s_n\}$$

é um caminho, os primeiros $n-1$ arcos têm de ser distintos pela Proposição 4.1. A subsequência

$$\{s_2, e_{j2}, s_3, e_{j3}, \dots, s_n, e_{jn}, s_{n+1}\}$$

também satisfaz a definição de caminho, assim os últimos $n-1$ arcos são distintos. Visto que C tem n arcos, a única possibilidade de duplicação é o último arco $e_{j(n)}$ ser igual ao primeiro $e_{j(1)}$, por isso, se impõe que tal duplicação não se pode verificar na definição de ciclo simples. ♦

Na continuação do texto só será utilizada o conceito de ciclo simples, pelo que se escreverá só ciclo, ficando subentendido que se trata de ciclo simples.

O *comprimento* de um caminho ou ciclo é definido pelo número de arcos desse caminho ou ciclo. Para qualquer caminho ou ciclo P , com comprimento n , definimos a sequência de orientação $O(P)$ dos arcos no caminho ou ciclo como sendo a sequência cujas elementos são:

$$O_i(P) = \begin{cases} +1 & \text{se } e_{j(i)} = (s_i, s_{i+1}), \\ -1 & \text{se } e_{j(i)} = (s_{i+1}, s_i). \end{cases}$$

Para o caminho $\{5, e_1, 1, e_7, 4, e_5, 3\}$ da Figura 4.3 $O(P) = \{-1, -1, 1\}$.

Proposição 4.3:

Seja a sequência finita $P = \{s_1, e_{j(1)}, s_2, e_{j(2)}, \dots, s_n, e_{j(n)}, s_{n+1}\}$ um caminho num grafo próprio G e seja A a matriz de incidência dos nós nos arcos. Então:

$$\sum_{i=1}^{i=n} O_i(P) A(j_i) = e^{s_1} - e^{s_{n+1}} \quad (4.1)$$

Prova: Seja $i \in \{1, \dots, n\}$.

Caso 1 – Suponhamos $e_{j(i)} = (s_i, s_{i+1})$, então: $O_i(P)A(j_i) = (+1)(e^{s_i} - e^{s_{i+1}}) = e^{s_i} - e^{s_{i+1}}$

Caso 2 – Suponhamos $e_{j(i)} = (s_{i+1}, s_i)$, então: $O_i(P)A(j_i) = (-1)(e^{s_{i+1}} - e^{s_i}) = e^{s_i} - e^{s_{i+1}}$

Portanto, em qualquer caso $O_i(P)A(j_i) = e^{s_i} - e^{s_{i+1}}$. Assim pela propriedade telescópica¹ da adição, (4.1) verifica-se. ♦

¹ Propriedade telescópica da adição:

$$\begin{aligned} O_1(P)A(j_1) &= e^{s_1} - e^{s_2} \\ O_2(P)A(j_2) &= e^{s_2} - e^{s_3} \\ O_3(P)A(j_3) &= e^{s_3} - e^{s_4} \\ &\dots = \dots \\ O_n(P)A(j_n) &= e^{s_n} - e^{s_{n+1}} \end{aligned}$$

$$\sum_{i=1}^{i=n} O_i(P) A(j_i) = e^{s_1} - e^{s_{n+1}}$$

Proposição 4.4:

Seja $C = \{s_1, e_{j(1)}, s_2, e_{j(2)}, \dots, s_n, e_{j(n)}, s_{n+1}\}$ um ciclo num grafo próprio G e seja A a matriz de incidências dos nós nos arcos. Então:

$$\sum_{i=1}^{i=n} O_i(C) A(j_i) = 0 \tag{4.2}$$

Prova:

$$\sum_{i=1}^{i=n} O_i(C) A(j_i) = \sum_{i=1}^{i=n-1} O_i(C) A(j_i) + e^{s_n} - e^{s_{n+1}}$$

Pela Proposição 4.3,

$$\sum_{i=1}^{i=n-1} O_i(C) A(j_i) = e^{s_1} - e^{s_n} .$$

Então:

$$\sum_{i=1}^{i=n} O_i(C) A(j_i) = e^{s_1} - e^{s_n} + e^{s_n} - e^{s_{n+1}}$$

Como C é um ciclo, $s_1 = s_{n+1}$, logo $\sum_{i=1}^{i=n} O_i(C) A(j_i) = 0$. ♦

Corolário 4.5:

Se $C = \{s_1, e_{j(1)}, s_2, e_{j(2)}, \dots, s_n, e_{j(n)}, s_{n+1}\}$ é um ciclo num grafo próprio G e A a matriz de incidências dos nós nos arcos; então $\{A(j_i): i=1, \dots, n\}$ é um sistema de vectores linearmente dependente.

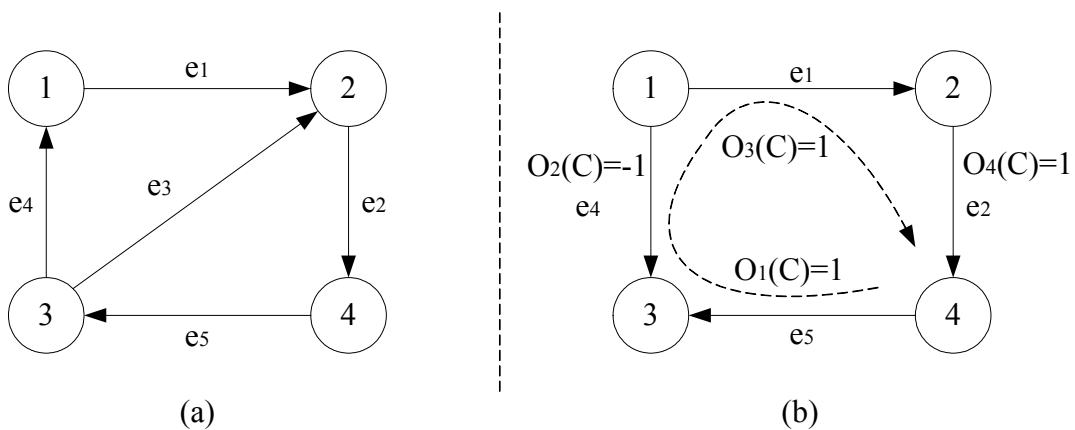


Figura 4.4 – Exemplo de um ciclo num grafo próprio. (a) – Grafo próprio. (b) – Ciclo simples $C = \{4, e_5, 3, e_4, 1, e_1, 2, e_2, 4\}$ num grafo próprio.

Vamos ilustrar o que se afirma no Corolário 4.5 através de um exemplo. Considere o ciclo C indicado na Figura 4.4. Então o conjunto das colunas associados aos arcos do ciclo são:

$$\left\{ \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \right\}$$

e são linearmente dependente, visto que:

$$O_1(C) \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + O_2(C) \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + O_3(C) \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + O_4(C) \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = (1) \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + (-1) \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + (1) \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + (1) \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Observe que se multiplicar aos elementos da sequência de orientação por um número real a relação anterior matem-se. Um grafo $G=[N, A]$ diz-se *acíclico* se não existe um ciclo usando os elementos do conjunto N e A . Um grafo $G=[N, A]$ diz-se *conexo* ou *ligado* se para qualquer par distinto de nós (i, j) de N , podemos formar, usando os elementos dos conjuntos N e A , um caminho que ligue esses nós, i.e., se para qualquer par distinto de nós for ligado por pelo menos um caminho.

Uma árvore é, por definição, um grafo acíclico conexo. Dado um grafo $G=[N, A]$ uma árvore $[N, A']$ com $A' \subseteq A$ é uma árvore geradora também dita de abrangente ou total de G , por definição de grafo gerador de G . Estas definições são ilustradas na Figura 4.5.

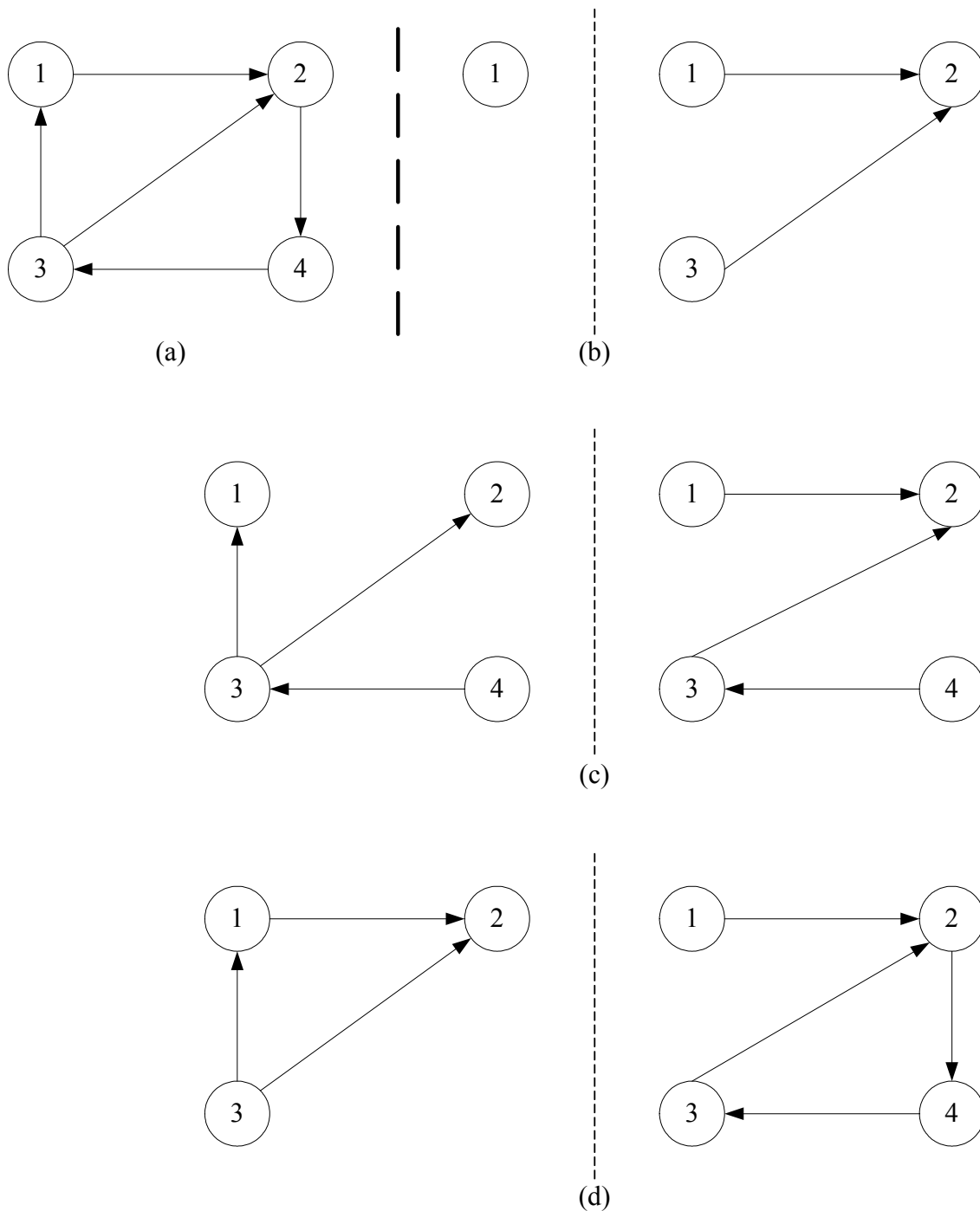


Figura 4.5 – Exemplos de árvores e árvores geradoras. (a) – Grafo G. (b) – Subgrafos de G que são árvores, mas não são árvores geradoras. (c) – Subgrafos de G que são árvores geradoras. (d) – Subgrafos de G que não são árvores.

Seguidamente, sem prova, enunciam-se um conjunto de preposições importantes que nos dão alternativas para a definição de árvore e propriedades dos grafos e das árvores.

Proposição 4.6:

Para um grafo $T=[N, A]$ com pelo menos um nó, as afirmações seguintes são equivalentes:

- 1 – T é uma árvore.
- 2 – T tem apenas para qualquer par distinto de nós (p, q) de N , um único caminho que liga p a q .
- 3 – T tem menos um arco que o número de nós e é conexo.
- 4 – T tem menos um arco que o número de nós e é acíclico.

Dado um grafo G , grau do nó i , $D(i)$, é o número de arcos de G incidentes no nó i . Por exemplo, o grau do nó 3 do grafo da Figura 4.5(a) é 3.

Um nó de uma árvore cujo grau é igual a 1 é chamado um *ponto extremo* da árvore. Semigrau interior (exterior) de um nó é o número de arcos que incidem para o interior (exterior) do nó.

Proposição 4.7:

A soma dos graus de todos os nós, de um grafo, é igual a duas vezes o número de arcos do grafo, i.e., $\sum_{i \in N} D(i) = 2\# A$.

Prova: Qualquer arco é incidente em dois nós, contribui portanto, duas vezes para o resultado da soma dos graus dos nós. ♦

Proposição 4.8:

Uma árvore com pelo menos dois nós tem pelo menos dois pontos extremos.

Prova: Seja $T=[N, A]$ uma árvore com o número de nós $m \geq 2$. Qualquer nó de T é incidente em pelo menos um arco, senão T não seria conexo. Assim o grau de cada nó é pelo menos 1. Pela Proposição 4.6, T tem $m-1$ arcos. Assumindo que T tem menos de dois pontos extremos, resultam as contradições:

Caso 1 – Admita que T não tem pontos extremos. Então, para cada $i \in N$, $D(i) \geq 2$.

Assim, $\sum_{i \in N} D(i) \geq 2m > 2(m-1)$ o que contradiz a Proposição 4.7.

Caso 2 – Admita que T tem só um ponto extremo. Seja j esse ponto extremo, portanto, $D(j)=1$ e para cada um dos restantes nós $i \in N - \{j\}$, $D(i) \geq 2$.

Então, o $\sum_{i \in N} D(i) = D(j) + \sum_{i \in N - \{j\}} D(i) \geq 1 + 2(m-1) \neq 2(m-1)$ o que contradiz a Proposição 4.7.

Portanto, T tem pelo menos dois pontos extremos. ♦

A segunda árvore mostrada na Figura 4.5(c) tem dois pontos extremos respectivamente. No entanto, uma árvore pode ter mais do que dois pontos extremos. Por exemplo, a primeira árvore mostrada na Figura 4.5(c) tem três pontos extremos. É fácil concluir que retirando um arco e um ponto extremo a uma árvore resulta um subgrafo que é também uma árvore. Portanto é válida a seguinte proposição.

Proposição 4.9:

Seja $T=[N, A]$ uma árvore com o número de nós $m \geq 2$. Seja i um ponto extremo de T , e seja e_j o arco incidente no nó i de T . Se $\hat{N}=N-\{i\}$ e $\hat{A}=A-\{e_j\}$, então $\hat{T}=[\hat{N}, \hat{A}]$ é uma árvore.

Prova: Como \hat{T} é um subgrafo da árvore T , pela Proposição 4.6, \hat{T} é acíclico. Por construção, \hat{T} tem menos um arco do que o número de nós. Portanto, pela Proposição 4.6, T é uma árvore. ♦

Estamos agora aptos a apresentar um resultado importante sobre os arcos de uma árvore e as correspondentes colunas na matriz de incidências dos nós nos arcos. Convém recordar que assumimos que a rede não contém arcos múltiplos. Assim, existe uma correspondência biunívoca entre os arcos de um grafo e as colunas da matriz de incidências dos nós nos arcos. Isto é, a um arco e_j de A corresponde uma coluna j da matriz de incidências dos nós nos arcos A e vice-versa.

Proposição 4.10:

Seja A a matriz de incidências dos nós nos arcos para um grafo próprio G e seja $T = [N, A]$ um subgrafo de G que é uma árvore com pelo menos dois nós. Então $\{A(j): e_j \in A\}$ é linearmente independente, i.e., as colunas da matriz A associadas aos arcos da árvore são linearmente independentes².

² Diz-se que o sistema de vectores $\{A_i: i=1, 2, \dots, n\}$ é linearmente independente se não existe uma série $\{\alpha_i\}_{i=1,2,\dots,n} \neq 0$ tal que $\sum_{i=1}^n \alpha_i A_i = 0$.

Prova: Seja m o número de nós da árvore T . Suponhamos $m = 2$. Então, pela Proposição 4.6, \mathbf{A} contém um arco e_j , atendendo a que $\mathbf{A}(j) = \mathbf{e}^{F(i)} - \mathbf{e}^{T(i)} \neq 0$, a proposição verifica-se para $m = 2$. Suponhamos $m > 2$. Seja n o maior dos inteiros tal que, para todos os subgrafos $\hat{T} = [\hat{N}, \hat{A}]$ de G que são árvores com k nós sendo $2 \leq k \leq n$, $\{\mathbf{A}(j): e_j \in \hat{A}\}$ é linearmente independente. Pelo parágrafo anterior sabemos que $n \geq 2$, assumindo por hipótese que $n < m$ conduz a uma contradição. Pela definição de n deverá haver uma árvore $\hat{T} = [\hat{N}, \hat{A}]$ com $n+1$ nós para a qual $\{\mathbf{A}(j): e_j \in \hat{A}\}$ é linearmente dependente. Logo tem de existir um conjunto de constantes $\{v_j: e_j \in \hat{A}\}$ com algum elemento não nulo tal que $\sum_{i \in \hat{A}} v_j \mathbf{A}(j) = 0$. Seja L o conjunto de pontos extremos da árvore \hat{T} . Pela Proposição 5.8 $L \neq \emptyset$ e seja p um ponto qualquer de L sendo e_w o arco incidente em p . Seja $T = [N, A]$, onde $N = \hat{N} - \{p\}$ e $A = \hat{A} - \{e_w\}$. Pela Proposição 5.9 T é uma árvore. Como, o nó p é incidente no arco e_w de \hat{A} , $\mathbf{A}_{pj} = 0$ para todos os $e_j \in A$. Assim, para a componente p

$$\sum_{e_j \in \hat{A}} v_j \mathbf{A}_{pj} = v_w \mathbf{A}_{p,w} + \sum_{e_j \in A} v_j \mathbf{A}_{pj} = v_w \mathbf{A}_{p,w} = 0.$$

Como $\mathbf{A}_{p,w} \in \{+1, -1\}$, seja $v_w = 0$. Então $\sum_{i \in A} v_j \mathbf{A}(j) = 0$ e $\{v_j: e_j \in A\}$ não são todos zeros. Portanto, $\{\mathbf{A}(j): e_j \in A\}$ é linearmente dependente. Mas, T é uma árvore com n nós, isto contradiz a definição de n . Consequentemente, $n = m$, portanto, todos os subgrafos de G que são árvores com dois ou mais nós têm a propriedade dos $\{\mathbf{A}(j): e_j \in A\}$ serem linearmente independente. ♦

Proposição 4.11:

Todo o grafo conexo $G = [N, A]$ com $N \neq \emptyset$ tem um subgrafo que é uma árvore geradora.

Prova. Seja G conexo, e seja n o número de nós de G . Se $n = 1$, então o conjunto dos arcos $A = \phi$, implica que G é uma árvore, sendo ela própria uma árvore geradora de si mesmo. Se $n \geq 2$, seja m o maior inteiro tal que, exista um subgrafo de G que é uma árvore e tem m nós. Qualquer subgrafo de G com um nó é uma árvore, então $m \geq 1$. Será mostrado que se for $m < n$ resulta uma contradição.

Pela definição de m , existe um subgrafo de G indicado por $\hat{T} = [\hat{N}, \hat{A}]$, que é uma árvore que tem m nós. Como, \hat{N} tem $m < n$ nós, $N - \hat{N} \neq \phi$. Seja L o conjunto dos arcos de A incidentes num nó de \hat{N} e noutra de $N - \hat{N}$. O conjunto $L \neq \phi$, visto que, caso contrário o grafo G não seria conexo. Dado um arco $e_i \in L$, seja j pertence a $N - \hat{N}$ o nó onde o arco i incide e seja $\bar{T} = [\bar{N}, \bar{A}]$, onde $\bar{N} = \hat{N} \cup \{j\}$ e $\bar{A} = \hat{A} \cup \{e_i\}$.

Por construção, \bar{T} é um subgrafo de G que é uma árvore e tem $m+1$ nós. Isto contradiz a definição de m . Logo tem que ser $m = n$, portanto, existe sempre a árvore geradora de um grafo G . ♦

As Proposições 4.10 e 4.11 mostram que para um dado grafo G , com a correspondente matriz de incidências dos nós nos arcos A , existe uma árvore geradora de G , indicada por T , tal que as colunas de A correspondentes aos arcos de T são linearmente independentes.

Mostremos agora em primeiro lugar que a característica de \mathbf{A} é igual ao número de nós menos um, e em segundo lugar que o conjunto máximo das colunas linearmente independentes de \mathbf{A} corresponde a uma árvore geradora.

Proposição 4.12:

Seja \mathbf{A} a matriz de incidência dos nós nos arcos do grafo próprio G que é conexo e tem n nós. Então a característica de \mathbf{A} é igual a $n-1$.

Prova: Pela Proposição 4.11, G tem uma árvore geradora T . Pela Proposição 4.6, T tem $n-1$ arcos. Pela Proposição 4.10, $\{\mathbf{A}(j): e_j \in A\}$ são linearmente independentes. Assim, a característica de \mathbf{A} é pelo menos $n-1$. Como $\mathbf{1A}=0$, a característica de \mathbf{A} é inferior a n . Então a característica de \mathbf{A} é $n-1$. ♦

Proposição 4.13:

Seja \mathbf{A} a matriz de incidência dos nós nos arcos do grafo próprio $G = [N, A]$ com n nós. Seja \hat{A} um subconjunto de A tal que $\{\mathbf{A}(j): e_j \in \hat{A}\}$ é linearmente independente e \hat{A} tem $n-1$ arcos. Então $T=[N, \hat{A}]$ é uma árvore.

Prova: O Corolário 4.5, implica que um ciclo não pode ser formado a partir de T , visto que, $\{\mathbf{A}(j): e_j \in \hat{A}\}$ é linearmente independente. Como T tem menos um arco que o número de nós e é acíclico, T é uma árvore pela Proposição 4.6. ♦

4.2- Caracterização de uma rede

Na nossa definição do problema PL normalizado, a matriz das restrições \mathbf{A} tem uma característica igual ao número de linhas. Pela Proposição 4.12, a matriz da incidência

dos nós nos arcos para o problema PR tem uma característica igual ao número de linhas menos um. Logo não se pode aplicar ao problema em redes o método do Simplex usando directamente a formulação anterior.

Existem duas maneiras de proceder, para aplicar o método do Simplex, ambas reformulando o problema PR de modo que a matriz das restrições tenha uma característica igual ao número de linhas, “full row rank”: (1) podemos fazer desaparecer uma linha da matriz de incidências dos nós nos arcos; ou (2) podemos adicionar uma coluna linearmente independente, i.e., adicionar uma nova variável que terá de ter o limite superior igual a zero. Assim consegue-se que a nova matriz das restrições tenha característica igual ao número de linhas. Nesta abordagem para o problema PLR escolhemos a última maneira de proceder. Consideremos a seguinte formulação para o PLR:

$$\left. \begin{array}{l} \min \mathbf{c}\mathbf{x} \\ \text{s.a.} \\ \mathbf{A}\mathbf{x} + a\mathbf{e}^k = \mathbf{r} \\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \\ 0 \leq a \leq 0 \end{array} \right\} \text{PR}$$

Onde \mathbf{A} é a matriz de incidência dos nós nos arcos, e k é um inteiro não superior ao número de nós n . Como a está restringido a zero, qualquer óptimo este PLR será óptimo para o problema PLR original.

Observe que o coeficiente desta variável na função objectivo é nulo. No entanto, o coeficiente não necessita de ser nulo, visto que, o valor que toma a variável é nulo. No subcapítulo 4.3, esta observação corresponde a afirmar que os potenciais dos nós são definidos a menos de uma constante aditiva.

Mostraremos que existe um conjunto de n colunas da matriz $[\mathbf{A} \mid \mathbf{e}^k]$ que é uma base para o espaço euclidiano n -dimensional .

Proposição 4.14:

Seja \mathbf{A} a matriz de incidência dos nós nos arcos para o grafo $G = [N, A]$ próprio e conexo, com n nós. Seja $T = [N, \hat{A}]$, uma árvore geradora de G . Então $\Omega = \{\mathbf{A}(j) : e_j \in \hat{A}\} \cup \{\mathbf{e}^k\}$ é uma base para o espaço \mathbf{R}^n .

Prova: A prova consiste em mostrar que os vetores $\{\mathbf{e}^k : k = 1, \dots, n\}$ são combinações lineares dos vectores de Ω .

O vector \mathbf{e}^k é uma combinação trivial. Seja $p \neq k$ e $p \in \{1, \dots, n\}$. Como T é uma árvore geradora de G pela Proposição 4.6, existe um único caminho $P = \{s_1, e_{j_1}, s_2, e_{j_2}, \dots, s_q, e_{j_q}, s_{q+1}\}$ em que T liga os nós p e k . Pela Proposição 4.3 sabe-se que

$$\sum_{i=1}^q O_i(P) \mathbf{A}(j_i) = \mathbf{e}^p - \mathbf{e}^k.$$

Consequentemente, $\mathbf{e}^k + \sum_{i=1}^q O_i(P) \mathbf{A}(j_i) = \mathbf{e}^p$, portanto, \mathbf{e}^p é uma combinação linear dos vectores de Ω . Pelo que fica provado que Ω é uma base para \mathbf{R}^n . ♦

Assim, conseguimos aumentando a rede que a matriz das restrições passe a ter característica igual ao número de linhas. Este elemento que foi introduzido é um arco que nasce num nó mas não termina em nenhum outro nó.

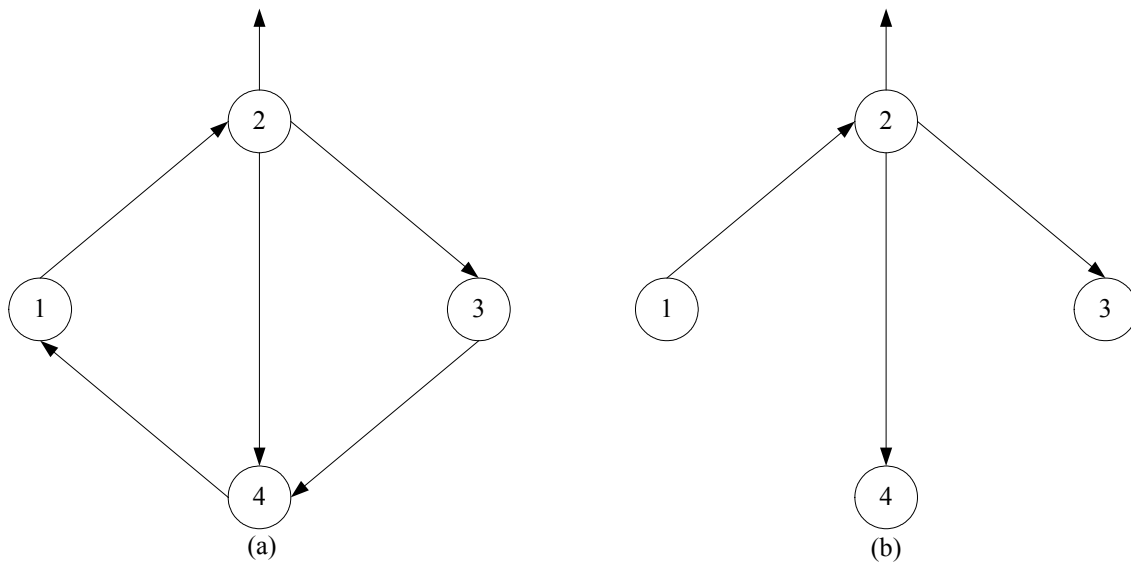


Figura 4.6 – Conceito de raízes. (a) – Grafo com raiz. (b) – Árvore com raiz.

Quando nos referimos a um grafo associado com este problema de PLR aumentado, o arco associado à variável a é dito arco raiz e no k um nó com raiz. Mais ainda, o grafo correspondente é chamado um grafo com raiz, e um grafo com raiz que seja uma árvore é chamado uma árvore com raiz. Os arcos raiz são representados nos desenhos por um arco incidente num único nó, saindo do nó, visto que, o respectivo elemento associado com a matriz A aumentada tem valor igual à unidade. A Figura 4.6 ilustra um grafo com raiz e uma sua árvore com raiz.

Proposição 4.15:

Seja A a matriz de incidência dos nós nos arcos para um grafo próprio com raiz $G = [N, A]$, com a raiz no nó k . Se Ω é o conjunto dos vectores que são as colunas da matriz básica para $[A \mid e^k]$, então $e^k \in \Omega$ e $T = [N, \hat{A}]$ com $\hat{A} = \{e_j; A(j) \in \Omega\}$ é uma árvore geradora de G .

Prova: Seja Ω uma matriz básica para $[\mathbf{A} \mid \mathbf{e}^k]$. Pela Proposição 4.14, a matriz aumentada $[\mathbf{A} \mid \mathbf{e}^k]$ tem a característica igual ao número de linhas. Pela Proposição 4.12, \mathbf{A} não tem a característica igual ao número de linhas, então $\mathbf{e}^k \in \Omega$. Seja n o número de nós de G . Então $\hat{\mathbf{A}}$ deverá corresponder a $n-1$ colunas linearmente independentes de \mathbf{A} . Pela Proposição 4.13, T é uma árvore geradora de G . ♦

Estamos agora aptos a efectuar a caracterização da base para o problema PLR.

Proposição 4.16:

Seja \mathbf{A} a matriz de incidência dos nós nos arcos para um grafo próprio, conexo e com raiz no nó k . O conjunto das colunas de \mathbf{A} correspondente a uma árvore geradora de G mais \mathbf{e}^k , é uma matriz básica para $[\mathbf{A} \mid \mathbf{e}^k]$.

Prova: A prova é uma consequência imediata das Proposições 4.14 e 4.15. ♦

Temos assim uma proposição que faz uma caracterização para uma base do problema PLR. Seguidamente apresentamos um teorema que caracteriza uma propriedade algébrica para a matriz básica de um problema PLR. Estas duas propriedades, em conjunto proporcionam a especialização do algoritmo primal simplex para o problema PLR.

Convém recordar que uma matriz, triangular, não singular, é uma matriz quadrada com valores diferentes de zero na diagonal principal e zeros acima da diagonal principal, ou aquela que é levada a ter esta forma pelo troca de linhas e colunas. A próxima

proposição mostra que a matriz básica para um problema PLR associado a um grafo próprio é triangular.

Proposição 4.17:

Seja \mathbf{A} a matriz de incidência dos nós nos arcos para um grafo próprio G , conexo e com raiz no nó 1. Se \mathbf{B} é uma matriz básica de $[\mathbf{A} \mid \mathbf{e}^1]$, então \mathbf{B} é triangular.

Prova: Seja $T = [N, A]$ uma árvore com raiz determinada pela matriz básica \mathbf{B} , e seja n o número de nós de T . Como G é um grafo próprio $n \geq 2$. Pela Proposição 4.8, T tem pelo menos dois pontos extremos. Seja n_1 um ponto extremo de T que não é nó raiz. Se $e_{j(1)}$ é o arco em T incidente em n_1 , então a linha n_1 de \mathbf{B} tem só um elemento não nulo, igual +1 se o arco está orientado do nó n_1 para outro nó, ou igual -1 se o arco está orientado de outro nó para nó n_1 . Assim, reordenando as linhas e colunas da matriz básica, tomando para a primeira linha e para a primeira coluna, a linha correspondente ao nó n_1 e a coluna correspondente ao arco $e_{j(1)}$, podemos representar \mathbf{B} da seguinte maneira

$$\left[\begin{array}{c|c} \overbrace{\pm 1}^{e_{j1}} & 0 \\ \hline \pm \mathbf{e}^s & \mathbf{B}^1 \end{array} \right] \left\{ \text{nó } n_1. \right.$$

Seja $T^1 = [N - \{n_1\}, A - \{e_{j(1)}\}]$. Pela Proposição 4.9, T^1 é uma árvore com $n - 1$ arcos. Se $n - 1 = 1$, então \mathbf{B}^1 é triangular, visto que, a última coluna de \mathbf{B} corresponde ao vector \mathbf{e}^k . Se $n - 1 > 1$, então, de novo pela Proposição 4.8, T^1 tem pelo menos dois pontos extremos. Considere de novo o processo, seja n_2 um ponto extremo de T^1 que não é nó

raiz e seja $e_{j(2)}$ o arco em T^1 incidente em n_2 . Então a linha n_2 de \mathbf{B}^1 tem só um elemento não nulo, como anteriormente igual a +1 ou -1.

Assim, tomando, respectivamente, para a segunda linha e para a segunda coluna, a linha correspondente ao nó n_2 e a coluna correspondente ao arco $e_{j(2)}$, podemos representar \mathbf{B} da seguinte maneira

$$\left[\begin{array}{cc|c|c} \overbrace{\pm 1}^{e_{j1}} & & & \\ \hline & & & \\ \hline \pm e^s & \pm 1 & & \\ \hline & \pm e^t & & \mathbf{B}^2 \end{array} \right] \begin{array}{l} \left. \begin{array}{l} \text{nó } n_1 \\ \text{nó } n_2 \end{array} \right\} \end{array}$$

Figura 4.7 – Processo para transformar a matriz \mathbf{B} numa matriz triangular.

Em cada repetição deste processo é introduzido na matriz reordenada uma nova linha e coluna, repetindo o processo o número de vezes igual ao número de nós do grafo, \mathbf{B} é transformada por reordenação das linhas e colunas na forma triangular. ♦

Observe que o último nó e arco que é escolhido pela reordenação é o nó raiz e o arco raiz, visto que, o arco raiz é incidente só num nó.

Convém recordar que uma matriz triangular permite uma inversão rápida com menos operações. Assim, determinada a matriz básica \mathbf{B} para um problema PLR após a

reordenação das linhas e colunas, obtendo a forma triangular $\hat{\mathbf{B}}$, é mais facilmente a obtenção das expressões onde existe $\hat{\mathbf{B}}^{-1}$.

A Proposição 4.17, não só mostra que a matriz básica de um problema PLR é triangular, mas também permite construir um algoritmo para determinar uma reordenação das linhas e das colunas que transforma a matriz básica de um problema PLR numa matriz na forma triangular.

ALGORITMO 4.1 – Algoritmo da transformação para a forma triangular

Passo 0 – *Iniciar*. Dada a matriz \mathbf{B} o problema PLR com grafo G . Sejam $T = [N, A]$ uma árvore geradora para G com n de nós e raiz no nó k . Iniciar o índice de iteração $i = 1$.

Passo 1 – *Encontrar um ponto extremo que não seja nó com raiz*. Seja $r \neq k$, um ponto extremo de T , e seja e_s um arco de T incidente em r .

Passo 2 – *Inserir a linha r e a coluna s em $\hat{\mathbf{B}}$* . Se $i \leftarrow n-1$ segue para o passo 4.

Passo 3 – *Reduzir a árvore (podar a árvore)*. Fazer $T = [N - \{r\}, A - \{e_s\}]$ e $i \leftarrow i + 1$, voltar ao passo 1.

Passo 4 – *Inserir o nó com raiz e o arco raiz*.

Como exemplo de aplicação deste algoritmo considere o problema do exemplo 3.1

$$\begin{aligned} \min \quad & x_1 + x_2 + 3x_3 + 10x_4 \\ \text{s.a} \quad & x_1 + \quad x_3 + x_4 = 5 \\ & -x_1 + x_2 \quad \quad + x_5 = 0 \\ & -x_2 - x_3 - x_4 = -5 \end{aligned}$$

as restrições de não negatividade são as seguintes:

$$0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 2, \quad 0 \leq x_3 \leq 4, \quad 0 \leq x_4 \leq 10, \quad 0 \leq x_5 \leq 1 .$$

O grafo com raiz é ilustrado na Figura 4.8. Existem três árvores geradoras possíveis para o grafo G do exemplo 3.1. As árvores geradoras com raiz, e as matrizes básicas já na forma triangular são ilustradas na Figura 4.9.

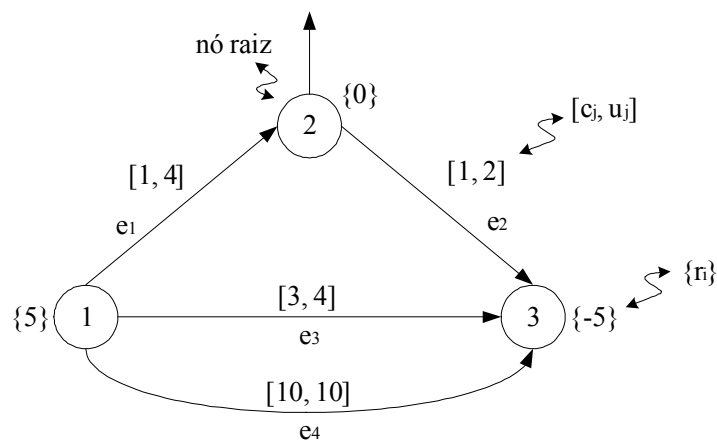
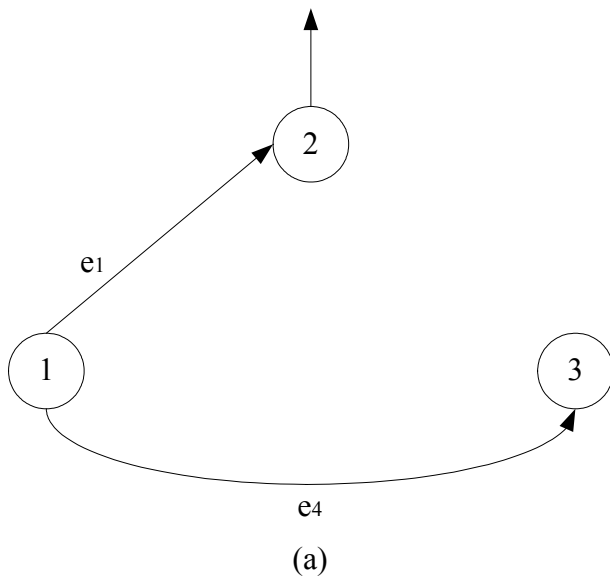


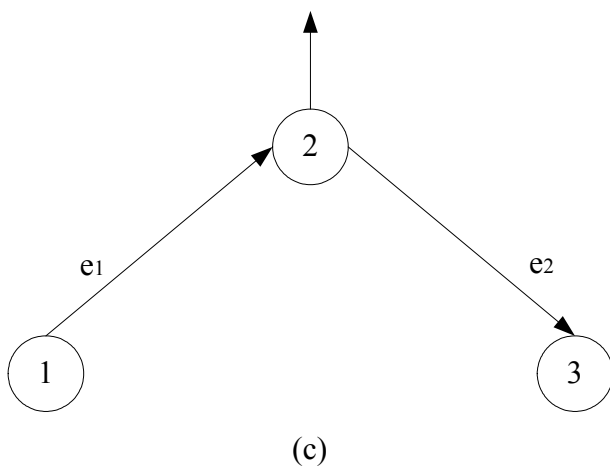
Figura 4.8 – Rede do exemplo 3.1 com raiz.

A representação de árvore geradora com raiz associadas com uma matriz básica do problema PLR torna possível e fácil a transformação para a forma triangular.



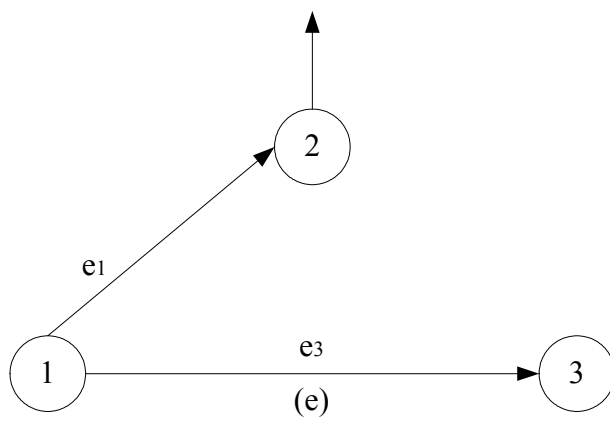
e4	e1	nós	
-1	0	0	3
1	1	0	1
0	-1	1	2

(b)



e1	e2	nós	
1	0	0	1
0	-1	0	3
-1	1	1	2

(d)



e3	e1	nós	
-1	0	0	3
1	1	0	1
0	-1	1	2

(f)

Figura 4.9 – Árvores geradoras com raiz para o grafo da Figura 4.8 e matriz **B**.

4.3- Especialização primal simplex

Considere que a matriz básica \mathbf{B} já está na forma triangular. Convém, recordar o passo do algoritmo primal simplex que necessita da inversão da base \mathbf{B} para saber se uma variável não básica x_i^N é escolhida para uma operação de alteração de fluxo: a componente i correspondente do vector $\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}$ indicada por $[\mathbf{c}^N - \mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}]_i$ tem que ser calculada (ver passo1 do ALG 3.2 – Algoritmo Primal Simplex). A inversão da base \mathbf{B} é necessária para o calculo do vector $\mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}$. Seja $\boldsymbol{\pi} = \mathbf{c}^B \mathbf{B}^{-1}$, observe que esta denominação está de acordo com a Proposição 3.11. As coordenadas do vector $\boldsymbol{\pi}$, os π_i 's, são chamados de variáveis duais ou potenciais dos nós. Portanto, $\boldsymbol{\pi}$ é a solução do sistema das equações lineares $\boldsymbol{\pi} \mathbf{B} = \mathbf{c}^B$. Como \mathbf{B} é triangular, $\boldsymbol{\pi}$ pode ser obtido por substituição de trás para a frente, resolvendo primeiro em ordem à última componente e depois, por substituição até a primeira componente.

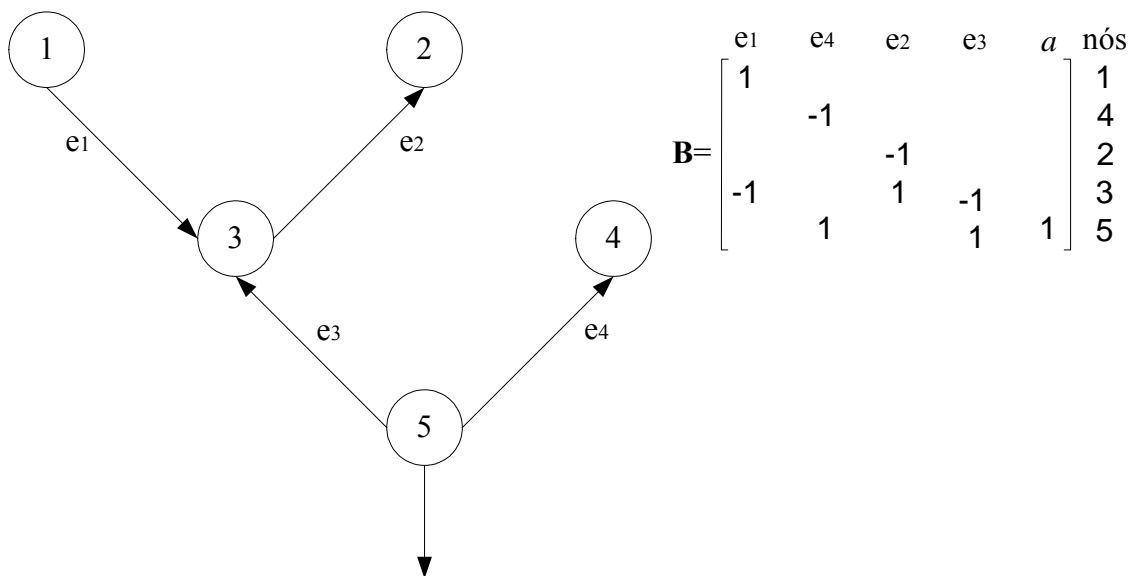


Figura 4.10 – Árvore geradora com raiz.

É fácil usar a árvore com raiz associada com \mathbf{B} , indicada seguidamente por T_B , na obtenção das componentes do vector $\boldsymbol{\pi}$. Como exemplo considere a árvore geradora com raiz com a correspondente base ilustrada na Figura 4.10. O sistema de equações para a árvore geradora T_B , $\boldsymbol{\pi} \mathbf{B} = \mathbf{c}^B$, tem a forma triangular ilustrada na Figura 4.11.

$$\begin{array}{rcl}
 \pi_1 & -\pi_3 & = c_1^B \\
 & & +\pi_5 = c_4^B \\
 & -\pi_4 & \\
 & -\pi_2 & +\pi_3 = c_2^B \\
 & & +\pi_5 = c_3^B \\
 & & \pi_5 = c_5^B
 \end{array}$$

Figura 4.11 – Sistema de equações $\boldsymbol{\pi} \mathbf{B} = \mathbf{c}^B$.

Obviamente, que c_5^B é nulo ou um valor qualquer indeterminado, pelo que já se afirmou no subcapítulo anterior. O sistema de equações pode ser resolvido por substituição de trás para a frente, começando com $\pi_5 = c_5^B$, geralmente é considerado nulo o potencial do nó raiz. No entanto, observe que os potenciais são definidos a menos de uma constante aditiva que é o potencial no nó raiz. Portanto, concluindo para uma base \mathbf{B} associada à árvore T_B com o nó raiz k , resolver em ordem a $\boldsymbol{\pi}$ o sistema $\boldsymbol{\pi} \mathbf{B} = \mathbf{c}^B$ conduz a:

$$\begin{array}{l}
 \pi_k = 0 \\
 \pi_{d(j)} - \pi_{p(j)} = c_j \text{ para } e_j \text{ pertencente a } T_B
 \end{array}$$

O algoritmo seguinte determina as coordenadas do vector $\boldsymbol{\pi}$ usando a conclusão anterior.

ALGORITMO 4.2 – Para determinação as variáveis duais

Passo 0 – *Iniciar*. Seja $T_B = [N, A]$ a árvore com raiz no nó k associada à base B .

Fazer $\pi_k = 0$, $N^L = \{k\}$ e $N^U = N - \{k\}$.

Passo 1 – *Encontrar um arco com nó “para” onde se dirige não pertencente a N^L* .

Seja $e_j \in A$ tal que $p(j) \in N^U$ e $d(j) \in N^L$. Se tal arco não existir, seguir para o passo 3.

Passo 2 – *Calcular o potencial do nó “para” onde se dirige o arco*.

Fazer $\pi_{p(j)} = -c_j + \pi_{d(j)}$, $N^L \leftarrow N^L \cup \{p(j)\}$, $N^U \leftarrow N^U - \{p(j)\}$, seguir para o passo 1.

Passo 3 – *Encontrar um Arco com nó “de” onde provém o arco não pertencente a N^L* .

Seja $e_j \in A$ tal que $d(j) \in N^U$ e $p(j) \in N^L$. Se tal arco não existir, o algoritmo termina.

Passo 4 – *Calcular o potencial do nó “para” onde se dirige o arco*.

Fazer $\pi_{d(j)} = c_j + \pi_{p(j)}$, $N^L \leftarrow N^L \cup \{T(j)\}$, $N^U \leftarrow N^U - \{T(j)\}$, seguir para o passo 1.

Assim, considerando que a variável dual para o nó raiz é nula, todas as variáveis duais ficam determinadas unívoca, visto que, o grafo da árvore com raiz é conexo e não tem ciclos. Os conjuntos dos arcos escolhidos para alteração de fluxo passam a ser descrito em função π como se segue:

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

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

As componentes de $\mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}$ podem ainda ser determinadas de outra modo. Para isso seja \mathbf{B} a base e T_B a correspondente árvore. A componente i do vector $\mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}$ é determinada por $\mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}(i)$. Seja $\mathbf{A}(k) = \mathbf{N}(i)$ e seja $\mathbf{y} = \mathbf{B}^{-1} \mathbf{A}(k)$, note que $e_k \notin T_B$, então $[\mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}]_i$ é igual a $\mathbf{c}^B \mathbf{y}$. Portanto para determinar \mathbf{y} há que resolver o sistema

$$\mathbf{B} \mathbf{y} = \mathbf{A}(k) = \mathbf{e}^{d(k)} - \mathbf{e}^{p(k)}. \quad (4.3)$$

Como \mathbf{B} é triangular, \mathbf{y} pode ser obtido facilmente, usando a árvore T_B para resolver o sistema de equações. Seja $P = \{s_1, e_{j_1}, s_2, e_{j_2}, \dots, s_n, e_{j_n}, s_{n+1}\}$ o único caminho em T_B ligando o nó $F(k)$ ao nó $T(k)$. Então pela Proposição 4.3:

$$\sum_{i=1}^{i=n} O_i(P) \mathbf{A}(j_i) = \mathbf{e}^{d(k)} - \mathbf{e}^{p(k)}$$

Então $\mathbf{c}^B \mathbf{B}^{-1} \mathbf{N}(i) = \mathbf{c}^B \mathbf{y}$ é igual a:

$$\sum_{i=1}^{i=n} c_{j_i} O_i(P) \quad (4.4)$$

Qualquer uma destas metodologias pode ser usada, contudo a primeira tem sido adoptada pela maior parte dos grupos que se dedicam ao desenvolvimento de códigos para problemas lineares em rede.

A equação 4.3, serve para determinar $\mathbf{y} = \mathbf{B}^{-1} \mathbf{A}(k)$ cujas coordenadas são então determinadas pela sequência de orientação dos arcos no caminho P :

$$y_n = \begin{cases} O_i(P) & \text{se } e_{k_n} = e_{k_j} \in P \\ 0 & \text{caso contrário} \end{cases} \quad (4.5)$$

Como $|O_i(P)| = 1$ o cálculo de Δ_1 e Δ_2 passa a ser feito através de:

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

Considere a árvore geradora com raiz da Figura 4.10. O arco $e_k = (2, 4)$ é um arco não pertence à árvore. O caminho $P = \{2, e_2, 3, e_3, \dots, 5, e_4, 4\}$ tem a sequência de orientação $O(P) = \{-1, -1, 1\}$. O ciclo formado para o cálculo das coordenadas de y é ilustrado na Figura 4.12.

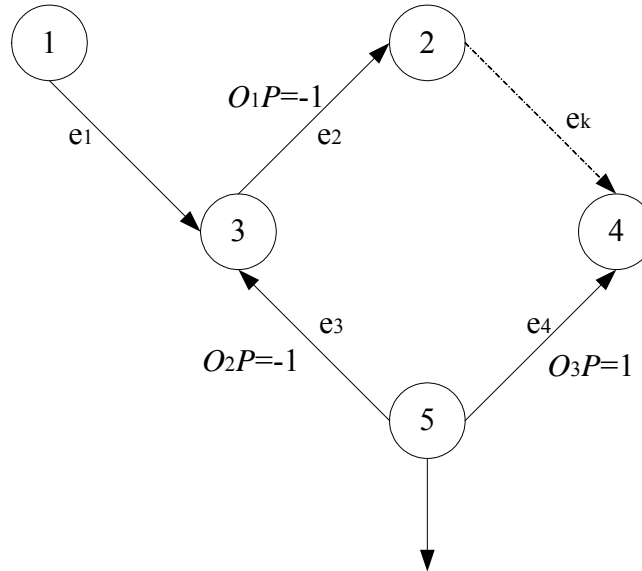


Figura 4.12 – Ciclo formado por (2, 4).

O algoritmo primal simplex num grafo é descrito pelos passos que de seguida são indicados.

ALGORITMO 4.3 – Método Primal Simplex num Grafo

Passo 0 – *Inicialização*. Seja $[x^B \mid x^N]$ uma solução básica possível sendo árvore T_B .

Calcular π usando o Algoritmo 4.2.

Passo 1 – *Avaliação*. Seja ψ_1 o conjunto dos arcos não básicos com fluxo igual a zero e

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

e ψ_2 o conjunto dos arcos não básicos com fluxo igual ao seu máximo e

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

Se $\psi_1 \cup \psi_2 = \emptyset$ o algoritmo termina, sendo $[\mathbf{x}^B \mid \mathbf{x}^N]$ um ponto óptimo; caso contrário seleccionar $e_k \in \psi_1 \cup \psi_2$ e fazer:

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

Passo 2 – *Atribuição do passo*. Seja $P = \{s_1, e_{j_1}, s_2, e_{j_2}, \dots, s_n, e_{j_n}, s_{n+1}\}$ o passo em T_B que liga $F(k)$ a $T(k)$. Fazer:

$$\Delta_1 = \min_{O_i(P)=\delta} \{x_{j_i}, +\infty\}, \Delta_2 = \min_{-O_i(P)=\delta} \{u_{j_i} - x_{j_i}, +\infty\} \text{ e } \Delta = \min\{\Delta_1, \Delta_2, u_k\}$$

Passo 3 – *Actualização dos Fluxos*. Fazer $x_k = x_k + \Delta \delta$. Para todos os arcos e_j pertencentes ao caminho P fazer $x_{ki} = x_{ki} - \Delta \delta O_i(P)$. Se $\Delta = u_k$, seguir para o passo 1.

Passo 4 – *Actualização da Árvore e das Variáveis Duais*. Seja:

$$\psi_3 = \{e_j: x_{j_i} = 0, \text{ onde } O_i(P) = \delta\} \text{ e } \psi_4 = \{e_j: x_{j_i} = u_{j_i}, \text{ onde } -O_i(P) = \delta\}.$$

Seleccionar um qualquer $e_m \in \psi_3 \cup \psi_4$. Substitui e_k por e_m , em T_B e actualizar as variáveis duais pelo Algoritmo 4.2, seguir para o passo 1.

Como exemplo de aplicação deste algoritmo considere de novo o Exemplo 3.1 que será agora resolvido usando o Algoritmo 4.3.

Passo 0 – (*Inicialização*). Seja $[x_1 \ x_2 \ x_3 \ x_4 \ x_5] = [0 \ 0 \ 4 \ 1 \ 0]$, sendo

$$\mathbf{x}^B = [x_1 \ x_4 \ x_5] \text{ e } \mathbf{x}^N = [x_2 \ x_3].$$

Passo 1 – (*Avaliação na Iteração 1*). Para e_2 : $\pi_2 - \pi_3 - c_2 = 0 - (-9) - 1 = 8 \Rightarrow e_2 \in \psi_1$. Para e_3 : $\pi_1 - \pi_3 - c_3 = 1 - (-9) - 3 = 7 \Rightarrow e_3 \notin \psi_1$ e $e_3 \notin \psi_2$. Assim, $\psi_1 = \{e_2\}$, $\psi_2 = \emptyset$.
logo $e_k = e_2$ e $\delta = 1$.

Passo 2 – (Atribuição do Passo). $\Delta_1 = \min\{x_4, +\infty\} = \min\{1, +\infty\} = 1$

$\Delta_2 = \min\{u_1 - x_1, +\infty\} = \min\{4 - 0, +\infty\} = 4$, $\Delta = \min\{\Delta_1, \Delta_2, u_k\} = \min\{1, 4, 2\} = 1$.

Passo 3 – (Atualização dos Fluxos). $x_2 = 0 + 1(1) = 1$

$x_1 = 0 - 1(1)(-1) = 1$, $x_4 = 1 - 1(1)(1) = 0$.

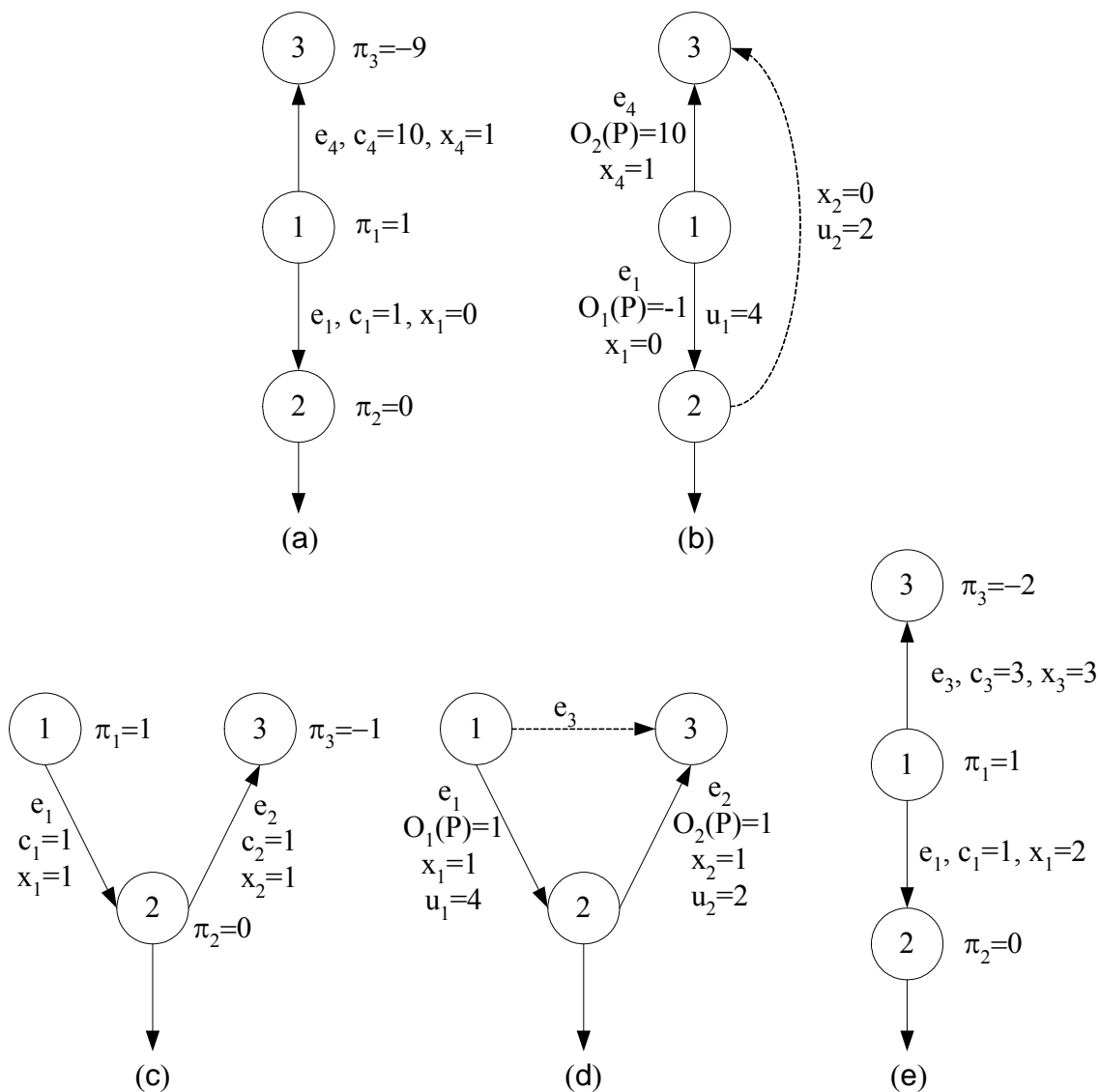


Figura 4.13 – Árvores e ciclos para a resolução do Exemplo 3.1.

- (a) – Árvore para a iteração 1. (b) – Ciclo para a iteração 1.
- (c) – Árvore para a iteração 2. (d) – Ciclo para a iteração 2.
- (e) – Árvore para a iteração 3.

Passo 4 – (*Actualização da Árvore e das Variáveis Duais*). $\psi_3 = \{e_4\}$, $\psi_4 = \phi$. Assim,

$$e_m = e_4, \mathbf{x}^B = [x_1 \quad x_2 \quad x_5], \text{ e } \mathbf{x}^N = [x_3 \quad x_4].$$

Passo 1 – (*Avaliação na Iteração 2 com* $[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5] = [1 \quad 1 \quad 4 \quad 0 \quad 0]$).

$$\text{Para } e_3: \pi_1 - \pi_3 - c_3 = 1 - (-1) - 3 = -1 \Rightarrow e_3 \in \psi_2.$$

$$\text{Para } e_4: \pi_1 - \pi_3 - c_4 = 1 - (-1) - 10 = -8 \Rightarrow e_4 \notin \psi_1 \text{ e } e_4 \notin \psi_2.$$

$$\text{Assim } e_k = e_3 \text{ e } \delta = -1.$$

Passo 2 – (*Atribuição do Passo*). $P = \{1, e_1, 2, e_2, 3\}$. $\Delta_1 = \min\{\infty\} = \infty$

$$\Delta_2 = \min\{4-1, 2-1, \infty\} = 1, \Delta = \min\{\infty, 1, 4\} = 1.$$

Passo 3 – (*Actualização dos Fluxos*). $x_3 = 4 + 1(-1) = 3$

$$x_1 = 1 - 1(-1)(1) = 2, x_2 = 1 - 1(-1)(1) = 2.$$

Passo 4 – (*Actualização da Árvore e das Variáveis Duais*). $\psi_3 = \phi$, $\psi_4 = \{e_2\}$. Assim,

$$e_m = e_2, \mathbf{x}^B = [x_1 \quad x_3 \quad x_5], \text{ e } \mathbf{x}^N = [x_2 \quad x_4].$$

Passo 1 – (*Avaliação na Iteração 3 com* $[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5] = [2 \quad 2 \quad 3 \quad 0 \quad 0]$).

$$\text{Para } e_2: \pi_2 - \pi_3 - c_2 = 0 - (-2) - 1 = 1 \Rightarrow e_2 \notin \psi_1 \text{ ou } \psi_2.$$

$$\text{Para } e_4: \pi_1 - \pi_3 - c_4 = 1 - (-2) - 10 = -7 \Rightarrow e_4 \notin \psi_1 \text{ e } e_4 \notin \psi_2.$$

Concluimos assim que, o óptimo foi atingido.

O método primal simplex num grafo é um método expedito não só para utilização didáctica em cálculo manual, mas também para uma implementação eficiente em computador, recorrendo a técnicas computacionais adequadas quer para armazenamento de dados, quer para processamento eficiente do código ditas de *tecnologia de implementação*. Esta tecnologia envolve o desenvolvimento de estruturas de base de dados, juntamente com algoritmos apropriados à execução dos vários passos do método, que não são abordados no âmbito deste texto. Para iniciar o algoritmo do simplex é

necessário conhecer uma solução básica possível. Esta solução nem sempre é fácil de ser encontrada. Em seguida será descrita uma técnica útil para determinar uma solução básica possível para iniciar o algoritmo do simplex.

4.4- Solução básica possível inicial

O passo inicialização do Algoritmo 4.3 exige encontrar uma solução básica possível com a qual se inicia o algoritmo. O método de “*all-artificial start*” é um método sistemático que permite encontrar uma solução básica possível com a qual se inicia o algoritmo. Neste método, somente os arcos adicionados, chamados arcos artificiais, tem fluxo não nulo. Os arcos da rede determinada pelo problema têm fluxos nulos. Para apresentar o método, considere uma rede, associada a um problema PLR, com \bar{I} nós.

O método “*all-artificial start*” envolve o aumento da rede pelo acréscimo de um novo nó, $\bar{I}+1$ com requisito nulo. Na rede original para cada nó fonte i , com requisito r_i positivo, adicionamos um arco artificial do nó i para $\bar{I}+1$ com custo unitário de transporte nulo e capacidade infinita. O fluxo neste arco é igual a r_i , satisfazendo a lei de conservação do fluxo no nó i . Na rede original para cada nó k com requisito r_k negativo (procura), adicionamos um arco artificial de $\bar{I}+1$ para k com custo unitário de transporte e capacidade infinitos. O fluxo neste arco será igual a $|r_k|$, satisfazendo a lei de conservação do fluxo no nó k . Como os custos unitários de transporte dos últimos arcos artificiais são infinitos, qualquer solução sem fluxo nestes arcos será melhor que uma solução que contemple fluxo não nulo neles. Portanto, fica assegurado que se existir um ponto ótimo o fluxo nos arcos artificiais tem que ser nulo. Esta formulação corresponde ao que tradicionalmente se chama, em literatura de programação linear,

método “*Big-M*”, i.e., método do grande M, visto que, o infinito em computação tem que ser simulado por um número grande.

Um exemplo do método “*all-artificial start*” é apresentado na Figura 4.14. Note-se que o nó 5 serve apenas como um destino fictício com uma procura nula.

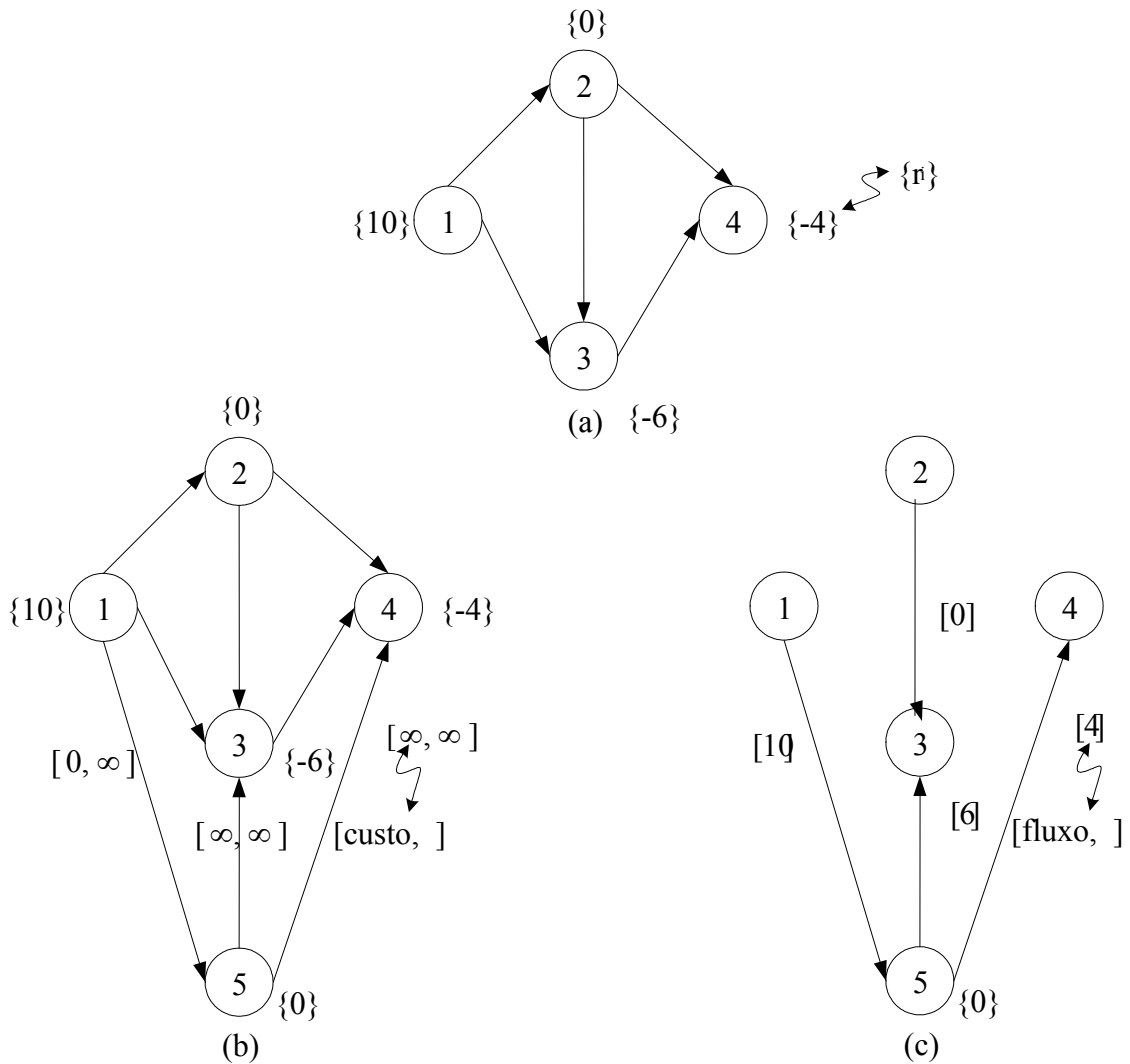


Figura 4.14 – Exemplo de “*all-artificial start*”. (a) – Rede original. (b) – Rede aumentada com o nó e os arcos artificiais. (c) – Árvore geradora.

4.5- Estratégias de avaliação

Um arco e_j pode ter uma alteração de fluxo favorável se $e_j \in \psi_1 \cup \psi_2$ (ver Passo 1 do Algoritmo 4.3). A estratégia de avaliação selecciona entre vários arcos dentro do conjunto $\psi_1 \cup \psi_2$, o arco não básico cujo fluxo será alterado. As estratégias usadas vai desde a selecção do primeiro arco candidato até à selecção, do arco candidato que tenha o maior valor absoluto de lucro reduzido $|\pi_{F(j)} - \pi_{T(j)} - c_j|$.

Na maior parte das estratégias de avaliação, o ficheiro dos arcos é gerido de forma cronológica, isto é, numa pesquisa anterior o último elemento observado foi o localizado na posição k então, a próxima em pesquisa começa na posição $k+1$. Sempre que o fim do ficheiro é alcançado, a pesquisa continua na posição 1. O método simplex termina quando a pesquisa através do ficheiro dos arcos não consegue encontrar um arco candidato que por uma alteração de fluxo melhora o valor corrente da função objectivo

5-ALGORITMO DA CONDIÇÃO PARA A PROGRAMAÇÃO EM REDES

Neste capítulo será apresentado outro método conhecido por *Algoritmo da Condição* para a resolução do problema de PLR. O problema PLR em estudo está escrito na seguinte formulação canónica:

$$\left. \begin{array}{l} \min \quad \mathbf{c}\mathbf{x} \\ \text{s.a} \quad \mathbf{A}\mathbf{x} = \mathbf{r} \\ \quad \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \end{array} \right\} \text{PR}$$

Onde \mathbf{A} é a matriz de incidências dos nós nos arcos e $\mathbf{1}\mathbf{r} = 0$, i.e., a soma dos fluxos que entram nos nós é igual à soma dos fluxos que saem dos nós. Ainda, neste algoritmo assume-se que $\mathbf{u} > \mathbf{0}$. O *Algoritmo da Condição* foi desenvolvido, como se disse no capítulo 1, por Delbert Fulkerson em 1961. Contrariamente ao primal simplex, o algoritmo da condição não é mais uma especialização de um método geral. Este algoritmo foi desenvolvido especificamente para o problema PLR. Alguns estudos computacionais têm sido efectuadas com o objectivo de comparar o desempenho do Algoritmo Simplex para problemas em rede com o do Algoritmo da Condição, mas ainda se mantém dúvidas sobre qual dos algoritmos é superior.

5.1- Condições de óptimo

Este subcapítulo apresenta um conjunto de condições permitem terminar o algoritmo da condição com a conclusão de que o óptimo foi identificado. Estas condições são uma especialização das condições de Karush-Kuhn-Tucker apresentadas no capítulo 3.4. Considere $[N, A]$ o grafo associada ao problema PLR, as condições de Karush-Kuhn-Tucker para o problema podem ser apresentadas pelo seguinte sistema de Karush-Kuhn-Tucker:

$$\mathbf{A} \mathbf{x} = \mathbf{r} \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \quad (5.1)$$

$$\pi_{F(j)} - \pi_{T(j)} - \mathbf{c}_j = \mu_j - \lambda_j \quad \text{para todo } j \in A \quad (5.2)$$

$$(\mathbf{x}_j - \mathbf{u}_j) \mu_j = 0 \quad \text{para todo } j \in A \quad (5.3)$$

$$\mathbf{x}_j \lambda_j = 0 \quad \text{para todo } j \in A \quad (5.4)$$

e

$$\mu_j \geq 0, \lambda_j \geq 0 \quad \text{para todo } j \in A \quad (5.5)$$

Caso seja obtido um conjunto de vectores $(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ que satisfaçam as condições 5.1 a 5.5, então \mathbf{x} é óptimo para o problema PLR, visto que, este problema é um problema de programação matemática convexo.

A $\bar{\mathbf{c}}_j = \pi_{F(j)} - \pi_{T(j)} - \mathbf{c}_j$ é associado o nome de *lucro reduzido*, o seu simétrico é o custo reduzido. Os vectores $\boldsymbol{\mu}$ e $\boldsymbol{\lambda}$ podem ser eliminados do sistema de Karush-Kuhn-Tucker acima apresentado, introduzido em sua substituição o lucro reduzido, visto que, $\lambda_j > 0$ e $\mu_j > 0$ não podem se verificar simultaneamente, atendendo a (5.3) e (5.4). e a que $\mathbf{u} > \mathbf{0}$.

Considere os casos.

Caso 1 – Para um arco j com $\bar{c}_j < 0$ por 5.2 e 5.5 $\lambda_j > 0$. Logo 5.4 não será satisfeito se $x_j \neq 0$, para o arco j 5.2 a 5.5 verificam-se se

$$\lambda_j = -\bar{c}_j, \mu_j = 0 \text{ e } x_j = 0.$$

Caso 2 – Para um arco j com $\bar{c}_j > 0$ por 5.2 e 5.5 $\mu_j > 0$. Logo 5.3 não será satisfeito se $x_j \neq u_j$, para o arco j 5.2 a 5.5 verificam-se se

$$\mu_j = \bar{c}_j, \lambda_j = 0 \text{ e } x_j = u_j.$$

Caso 3 – Para um arco $e_j, \bar{c}_j = 0$ por 5.2 e 5.5 verificam-se se $\mu_j = \lambda_j = 0$, visto que, $\lambda_j > 0$ e $\mu_j > 0$ não podem se verificar simultaneamente.

As condições de óptimo anteriores escrevem-se como:

$$\mathbf{Ax} = \mathbf{r} \tag{5.6}$$

e

$$\text{para o arco } j: \left\{ \begin{array}{ll} \bar{c}_j < 0 & \text{quando } x_j = 0 \\ \bar{c}_j = 0 & \text{quando } 0 \leq x_j \leq u_j \\ \bar{c}_j > 0 & \text{quando } x_j = u_j \end{array} \right\} \tag{5.7}$$

$$\text{com } \bar{c}_j = \pi_{F(j)} - \pi_{T(j)} - c_j$$

Se $(\mathbf{x}, \boldsymbol{\pi})$ satisfazem 5.6 e 5.7, \mathbf{x} é um óptimo para a PLR.

5.2- Metodologia para o algoritmo

O algoritmo da condição começa por iniciar o vector π , vector cujas coordenadas são as *variáveis duais*, também conhecidas por potenciais dos nós, e o vector dos fluxos \mathbf{x} , satisfazendo $\mathbf{A} \mathbf{x} = \mathbf{r}$, $\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$, tal como o Algoritmo do Simplex necessita de uma solução básica admissível inicial. Os fluxos podem ser obtidos de forma sistemática pelo método dos arcos e nó artificiais, como já se disse em 4.4. Se os fluxos satisfazem 5.6 e 5.7 o algoritmo termina com a conclusão que o vector \mathbf{x} corrente é um óptimo, caso contrário os fluxos \mathbf{x} , ou as variáveis duais π ou ambos são alterados.

Se um arco j satisfaz a 5.7, então e_j é dito estar *na condição*, caso contrário, e_j está *fora da condição*. As várias condições que podem ocorrer são ilustradas no Quadro 5.1.

Quadro 5.1 – Condições possíveis de um arco e_j

	$\bar{c}_j < 0$	$\bar{c}_j = 0$	$\bar{c}_j > 0$
$x_j = u_j$	Fora da condição	Na condição	Na condição
$0 < x_j < u_j$	Fora da condição	Na condição	Fora da condição
$x_j = 0$	Na condição	Na condição	Fora da condição

Observe-se que das nove condições possíveis, cinco delas apresentam e_j na condição.

Para cada uma das nove condições é definido um *número de condição*, como se apresenta no Quadro 5.2.

Quadro 5.2 – Condições possíveis de um arco e_j

	$\bar{c}_j < 0$	$\bar{c}_j = 0$	$\bar{c}_j > 0$
$x_j = u_j$	x_j	0	0
$0 < x_j < u_j$	x_j	0	$u_j - x_j$
$x_j = 0$	0	0	$u_j - x_j$

Note-se que os números de condição são todos não negativos e são as alterações de fluxo necessário para que o arco esteja em condição. Na primeira coluna o fluxo no arco tem que decrescer, na segunda é indiferente, enquanto que na terceira tem que ser incrementado. Seja K_j o número de condição do arco e_j . O número de condição da solução corrente, $(\mathbf{x}, \boldsymbol{\pi})$, é definido por

$$\sum_{e_j \in A} K_j.$$

Uma solução cujo número de condição é zero resolve o problema PLR, ou seja, quanto mais próximo o número de condição estiver de zero, mais próximo se está da solução ótima. O método para o algoritmo da condição envolve alterações de $(\mathbf{x}, \boldsymbol{\pi})$ de forma a que as seguintes condições se verifiquem:

- 1 – O número de condição de um arco nunca aumenta.
- 2 – O número da condição é reduzido, necessariamente em algum arco, após um número de iterações finito.

O algoritmo da condição tem duas fases, a fase primal que introduz alterações nos fluxos dos arcos, e a fase dual que introduz alterações nas variáveis duais, ditas também

potenciais dos nós. Ambas as fases tem como objectivo reduzir o número de condição, i.e., algoritmo da condição faz iterações nas duas fases, a primal e a dual, até que se verifique a condição de paragens:

$$\sum_{j \in A} K_j = 0$$

5.3- Fase primal

Durante a fase primal as variáveis duais π mantêm-se fixas. Os fluxos que são alterados pertencem aos arcos de um ciclo que será escolhido convenientemente como será esclarecido mais à frente. Seja $C = \{ s_1, e_{j(1)}, s_2, e_{j(2)}, s_3, e_{j(3)}, \dots, s_n, e_{j(n)}, s_{n+1} \}$, $s_1 = s_{n+1}$, um ciclo simples no grafo $[N, A]$ e seja a sequência de orientação indicada por $O(C)$. $O_i(C)$ é o número da orientação do arco i e $\mathbf{A}(j_i)$ o vector coluna da matriz \mathbf{A} corresponde a esse arco, o somatório dos produtos, do número de orientação de cada arco pelo respectivo vector $\mathbf{A}(j_i)$ é nulo

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

pela Proposição 4.4

Seja \mathbf{y} dado por

$$y_k = \begin{cases} O_i(C) & \text{se } \exists e_{j_i} \in C \text{ tal que } e_{j_i} = e_k \\ 0 & \text{caso contrário} \end{cases}$$

O vector \mathbf{y} tem como coordenadas diferentes de zero os números de orientações dos arcos da rede relativamente a um ciclo C : caso um arco não pertença a este ciclo a coordenada respectiva é nula. Ou seja, se o arco da rede e_k for também arco do ciclo C , então $y_k = O_i(C)$ (orientação do arco no ciclo), caso contrário $y_k = 0$. Logo $\mathbf{A} \mathbf{y} = 0$.

Como ilustração considere a rede na Figura 5.1.

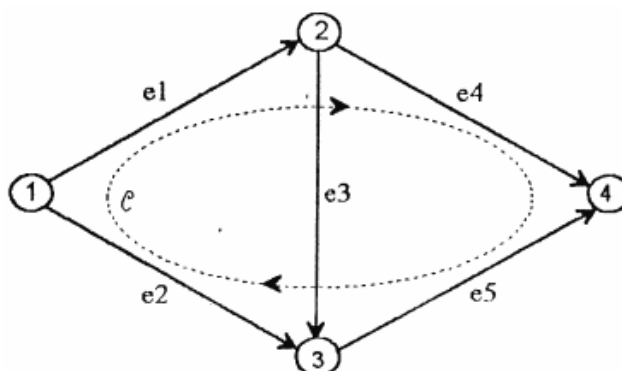


Figura 5.1 – Exemplo de uma rede

Considere o ciclo $C = \{1, e_1, 2, e_4, 4, e_5, 3, e_2, 1\}$ a orientação dos arcos do ciclo é $O(C) = \{1, 1, -1, -1\}$, sendo $y = [1 \ -1 \ 0 \ 1 \ -1]$. A matriz de incidências dos nós nos arcos é dada por:

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

e

$$\mathbf{A} \mathbf{y} = \begin{bmatrix} 1 & 1 & & & \\ -1 & & 1 & 1 & \\ & -1 & -1 & & 1 \\ & & & -1 & -1 \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \\ 0 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Considere o vector dos fluxos \mathbf{x} possível. Seja $\Delta > 0$, dado um ciclo e o correspondente vector das orientações \mathbf{y} , $\hat{\mathbf{x}} = \mathbf{x} + \Delta \mathbf{y}$ satisfaz $\mathbf{A} \hat{\mathbf{x}} = \mathbf{r}$. A transformação de fluxo

$\hat{\mathbf{x}} = \mathbf{x} + \Delta\mathbf{y}$ no ciclo C diz-se que aumenta o fluxo Δ . No entanto, convém salientar que tal aumento corresponde para um arco e_k cuja orientação $y_k = -1$ a uma diminuição de valor Δ no seu fluxo.

Considere K_k o número de condição para o arco e_k com fluxo x_k , e \hat{K}_k o número de condição para o arco e_k com fluxo \hat{x}_k . Dado o vector dos fluxos \mathbf{x} , a fase primal tenta encontrar um ciclo de modo que para $\hat{\mathbf{x}} = \mathbf{x} + \Delta\mathbf{y}$, $\hat{K}_k \leq K_k$ para todo o arco e_k e $\hat{K}_m < K_m$ para pelo menos um arco e_m . Por outras palavras, tentar encontrar um ciclo de modo que um aumento do fluxo nesse ciclo não aumente o número de condições de nenhum arco e diminua o número de condições de pelo menos um. No quadro 5.3, são apresentadas as alterações de fluxo possíveis para os arcos num ciclo.

Quadro 5.3 – Alterações de fluxo possíveis para o arco e_j

	$\bar{c}_j < 0$	$\bar{c}_j = 0$	$\bar{c}_j > 0$
$x_j = u_j$	Reduzir de x_j	Reduzir de x_j	Nenhuma Alteração Admitida
$0 < x_j < u_j$	Reduzir de x_j	Aumentar de $u_j - x_j$	Aumentar de $u_j - x_j$
$x_j = 0$	Nenhuma Alteração Admitida	Aumentar de $u_j - x_j$	Aumentar de $u_j - x_j$

O método usado na fase primal do algoritmo é a seguinte: Dado qualquer arco fora de condição, e_s , encontrar um ciclo que inclua e_s e permita de acordo com o Quadro 5.3 um aumento de fluxo no ciclo. Suponha que para o arco seleccionado fora de condição

$\bar{c}_s > 0$. O método para verificar se existe um ciclo que inclua e_s e permita de acordo com o Quadro 5.3 um aumento de fluxo no ciclo, consiste na obtenção de uma árvore do grafo, designada por T , e com o nó raiz no nó $F(s)$. Esta árvore será de modo que, se o nó $i \in T$, um aumento de fluxo será apenas permitido no único caminho que na árvore T ligue o nó i ao nó raiz. Se $T(s) \in T$, o ciclo é o único caminho na árvore T vai do nó $T(s)$ ao nó $F(s)$ fechando pelo e_s , usando este caminho e retornando-se a $T(s)$ através do arco e_s encontra-se o ciclo pretendido.

O algoritmo para a fase primal é iniciado com a escolha de um arco fora da condição, seja esse arco indicado por e_s .

ALG 5.1 – Fase Primal

Passo 0 – *Iniciar*. Se $\bar{c}_s < 0$, então $\hat{N} = \{T(s)\}$ e $\Delta_{T(s)} = x_s$; caso contrário $\hat{N} = \{F(s)\}$ e $\Delta_{F(s)} = u_s - x_s$. Atribuir ao conjunto dos arcos da árvore o conjunto vazio $\hat{A} = \phi$.

Passo 1 – *Determinar os arcos para a árvore*. Seja

$$\psi_1 = \left\{ e_j : e_j \neq e_s, \bar{c}_j \geq 0, x_j < u_j, F(j) \notin \hat{N}, e T(j) \in \hat{N} \right\}$$

e

$$\psi_2 = \left\{ e_j : e_j \neq e_s, \bar{c}_j \leq 0, x_j > 0, F(j) \in \hat{N}, e T(j) \notin \hat{N} \right\}$$

isto é, ψ_1 é o conjunto dos arcos diferentes do arco e_s , cujo lucro reduzido $\bar{c}_j \geq 0$, tendo fluxo menor que o fluxo máximo e que não partem dos nós pertencentes à árvore, mas terminam num nó pertencente à árvore; e ψ_2 é o

conjunto dos arcos, diferentes do arco e_s cujo lucro reduzido $\bar{c}_j \leq 0$, tendo fluxo maior que zero e que partem dos nós pertencentes à árvore, mas chegam aos nós não pertencentes à árvore. Se $\psi_1 \cup \psi_2 = \phi$, conclui-se que não existe possibilidade de construir o ciclo pretendido, então segue-se para o algoritmo da fase dual.

Passo 2 – *Juntar à árvore um novo arco*. Seleccionar um $e_k \in \psi_1 \cup \psi_2$. Se $e_k \in \psi_1$, fazer $\Delta_{F(k)} = \min\{\Delta_{T(k)}, u_k - x_k\}$. Se $e_k \in \psi_2$, fazer $\Delta_{T(k)} = \min\{\Delta_{F(k)}, x_k\}$. Actualizar o conjunto dos nós e dos arcos, $\hat{N} = \hat{N} \cup \{F(k), T(k)\}$ e $\hat{A} = \hat{A} \cup \{e_k\}$. Se $\{F(s), T(s)\} \subset \hat{N}$, seguir para o passo 3; caso contrário seguir para o passo 1.

Passo 3 – *Calcular os fluxos*. Se $\bar{c}_s < 0$, aumentar o fluxo no ciclo de $\Delta_{F(s)}$; caso contrário aumentar o fluxo no ciclo de $\Delta_{T(s)}$.

5.4- Fase dual

Se a fase primal termina com a conclusão de que não existe ciclo, então alteram-se as variáveis duais de modo a não aumentar o número de condição, isto é, mantendo os fluxos dos arcos e ajustando as variáveis duais de forma conveniente.

Seja a árvore com que terminou a fase primal indicada por $T = [\hat{N}, \hat{A}]$. Na fase dual reduz-se cada uma das variáveis duais associadas com um nó em \hat{N} de um valor θ , onde

θ será determinado pelo algoritmo seguinte. Seja $e_j \in \hat{A}$. Então após a redução das variáveis duais têm-se

$$\hat{\pi}_{F(r)} = \pi_{F(r)} - \theta \text{ e } \hat{\pi}_{T(r)} = \pi_{T(r)} - \theta.$$

Portanto

$$\hat{\pi}_{F(r)} - \hat{\pi}_{T(r)} - c_r = \pi_{F(r)} - \pi_{T(r)} - c_r.$$

Assim, reduzindo o valor de cada variável dual corresponde a nós em T , não afecta o número de condições dos arcos de T . Os únicos arcos cujos números de condição são afectados por esta alteração são aqueles que são incidentes num nó pertencente \hat{N} e num outro nó pertencente $N - \hat{N}$.

O objectivo da fase dual é determinar o valor da redução θ para as variáveis duais, associadas com os nós de T , de tal modo que nenhum número de condição dos arcos sofra aumento e pelo menos um seja reduzido.

A alteração possível para um arco e_j , com $T(j) \in \hat{N}$ e $F(j) \in N - \hat{N}$, é dada no quadro 5.4, enquanto que as alterações possíveis para os arcos e_j , com $F(j) \in \hat{N}$ e $T(j) \in N - \hat{N}$, é dada no quadro 5.5. as zonas em ambos os quadros sem indicação de valores correspondem a situações que não podem ocorrer na fase dual, visto que, correspondem a situações de expansão da árvore da fase primal, i.e., correspondem a ter-se $\Psi_1 \cup \Psi_2 \neq \phi$.

Quadro 5.4 – Redução possível em $\pi_{T(j)}$

	$\bar{c}_j < 0$	$\bar{c}_j = 0$	$\bar{c}_j > 0$
$x_j = u_j$	$-\bar{c}_j$	∞	∞
$0 < x_j < u_j$	$-\bar{c}_j$	■	■
$x_j = 0$	$-\bar{c}_j$	■	■

Quadro 5.5 – Redução possível em $\pi_{F(j)}$

	$\bar{c}_j < 0$	$\bar{c}_j = 0$	$\bar{c}_j > 0$
$x_j = u_j$	■	■	\bar{c}_j
$0 < x_j < u_j$	■	■	\bar{c}_j
$x_j = 0$	∞	∞	\bar{c}_j

O algoritmo da fase dual é iniciado com a árvore $T = [\hat{N}, \hat{A}]$, obtida na fase primal.

ALG 5.2– Fase Dual

Passo 1 – *Determinar os arcos incidentes em T*. Seja

$$\psi_3 = \left\{ e_j : T(j) \in \hat{N}, F(j) \notin \hat{N}, \text{ e } \bar{c}_j < 0 \right\}$$

$$\text{e } \psi_4 = \left\{ e_j : T(j) \notin \hat{N}, F(j) \in \hat{N}, \text{ e } \bar{c}_j > 0 \right\}$$

isto é, ψ_3 é o conjunto dos arcos que terminam nos nós pertencentes a \hat{N} , mas partem de nós não pertencentes a \hat{N} , e têm lucro reduzido $\bar{c}_j < 0$; e ψ_4 é o conjunto dos arcos que chegam a nós não pertencentes a \hat{N} , mas partem de nós pertencentes a \hat{N} , e têm lucro reduzido $\bar{c}_j > 0$.

Passo 2 – *Determinar a alteração máxima possível*. Fazer :

$$\theta = \min_{e_j \in \psi_3 \cup \psi_4} \{ |\bar{c}_j| \}$$

θ é o menor dos módulos dos lucros reduzidos $|\bar{c}_j|$, dos arcos pertencentes à união dos conjuntos ψ_3 e ψ_4 .

Passo 3 – *Reduzir as variáveis duais*: Fazer:

$$\pi_i = \pi_i - \theta \text{ para todos os } i \in \hat{N}.$$

Note que um arco $e_k \in \psi_3 \cup \psi_4$, para o qual no passo 2 da fase dual tinha $|\bar{c}_k| = \theta$, depois do passo 3, terá um lucro reduzido $\bar{c}_k = 0$. Por este motivo, se $|\bar{c}_s| \neq \theta$ e a fase primal for repetida, a árvore anterior é de novo obtida com a adição do arco e_k e de um novo nó que é um ponto extremo. Se $|\bar{c}_s| = \theta$, então e_s estará na condição, tem

consequentemente que determinar-se um outro arco fora de condição antes de voltar à fase primal.

5.5- Algoritmo da condição

O algoritmo da condição pode agora ser apresentado da seguinte forma:

ALG: 5.3 – Algoritmo da Condição

Passo 0 – *Iniciar*. Seja \mathbf{x} um vector dos fluxos que seja uma solução possível. Uma solução possível pode ser sempre obtida usando variáveis artificiais, como se disse em 4.4. Seja π um vector de variáveis duais.

Passo 1 – *Encontrar um arco fora da condição*. Seja e_s um arco fora da condição. Se todos os arcos estão na condição, terminar, concluindo que \mathbf{x} é um ótimo.

Passo 2 – *Chamar a Fase Primal*. Executar o algoritmo ALG 5.1 com o arco e_s (arco fora da condição). Se o algoritmo ALG 5.1 terminar com a conclusão de que não existe ciclo, avançar para o passo 3; caso contrário voltar ao passo 1.

Passo 3 – *Fase Dual*. Executar o algoritmo ALG 5.2 com a árvore desenvolvida no passo 2. Se e_s está fora da condição, voltar para o passo 2; caso contrário voltar para o passo 1.

Considere novamente o exemplo 4.1. Por comodidade este problema é ilustrado na Figura 5.2. A variável x_5 foi omitida da rede uma vez que a soma das três restrições de igualdade implica $x_5 = 0$. Na apresentação feita no capítulo 4.3 a coluna associada com x_5 foi necessária para assegurar que a característica da matriz das incidências dos nós nos arcos fosse igual ao número de linhas desta matriz. O algoritmo da condição não necessita de tal pressuposto, por isso x_5 pode ser retirado da formulação do problema.

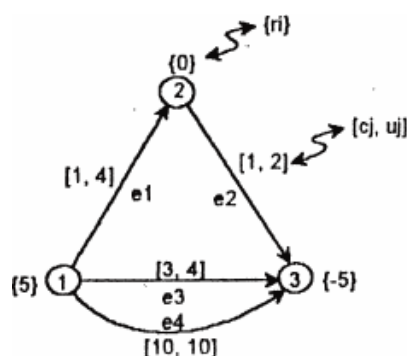


Figura 5.2 – exemplo de rede para o algoritmo da condição

Ainda por comodidade, antes de iniciar a resolução, vamos recordar o enunciado do problema:

$$\begin{aligned} \min \quad & x_1 + x_2 + 3x_3 + 10x_4 \\ \text{s.a} \quad & x_1 + x_3 + x_4 = 5 \\ & -x_1 + x_2 = 0 \\ & -x_2 - x_3 - x_4 = -5 \end{aligned}$$

com as seguintes restrições de não negatividade:

$$0 \leq x_1 \leq 4, 0 \leq x_2 \leq 2, 0 \leq x_3 \leq 4, 0 \leq x_{10} \leq 10.$$

ALG 5.3

Passo 0 – (Iniciar).

$$\text{Fazer } [x_1 \ x_2 \ x_3 \ x_4] = [0 \ 0 \ 4 \ 1]^{(*)} \text{ e } [\pi_1 \ \pi_2 \ \pi_3] = [5 \ 0 \ -1]$$

Passo 1 – (Encontrar um arco fora da Condição).

Determinar os lucros reduzidos de cada arco:

$$\bar{c}_1 = \pi_{F(1)} - \pi_{T(1)} - c_1 = 5 - 0 - 1 = 4$$

$$\bar{c}_2 = \pi_{F(2)} - \pi_{T(2)} - c_2 = 0 - (-1) - 1 = 0$$

$$\bar{c}_3 = \pi_{F(3)} - \pi_{T(3)} - c_3 = 5 - (-1) - 3 = 3$$

$$\bar{c}_4 = \pi_{F(4)} - \pi_{T(4)} - c_4 = 5 - (-1) - 10 = -4$$

Usando o quadro 5.2 determinar o número de condição de cada arco e

usando o quadro 5.1 averiguar se está na condição ou não.

$$x_1 = 0; \bar{c}_1 = 4 \Rightarrow K_1 = 4 \rightarrow \text{fora da condição}$$

$$x_2 = 0; \bar{c}_2 = 0 \Rightarrow K_2 = 0 \rightarrow \text{na condição}$$

$$x_3 = 4; \bar{c}_3 = 3 \Rightarrow K_3 = 0 \rightarrow \text{na condição}$$

$$x_4 = 1; \bar{c}_4 = -4 \Rightarrow K_4 = 1 \rightarrow \text{fora da condição}$$

$$\sum_{j=1}^4 K_j = 5$$

existem dois arcos fora da condição, e_1 e e_4 , escolhendo um qualquer, seja e_4

para o arco escolhido para e_s , passar à fase primal.

ALG 5.1

Passo 0 – (Iniciar).

Como $\bar{c}_4 < 0$ então $\hat{N} = \{T(4)\} = \{3\}$, $\Delta_{T(4)} = x_4 \Rightarrow \Delta_3 = 1$ e o conjunto dos

arcos $\hat{A} = \emptyset$.

Passo 1 – (Determinar arcos da árvore).

Determinar os arcos dos conjuntos ψ_1 e ψ_2 .

$$\psi_1 = \{e_j : e_j \neq e_s, \bar{c}_j \geq 0, x_j < u_j, F(j) \notin \hat{N} \text{ e } T(j) \in \hat{N}\} = \{e_2\} \text{ e}$$

$$\psi_2 = \{e_j : e_j \neq e_s, \bar{c}_j \leq 0, x_j > 0, F(j) \in \hat{N} \text{ e } T(j) \notin \hat{N}\} = \phi$$

como $\psi_1 \cup \psi_2 \neq \phi$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$\psi_1 \cup \psi_2 = \{e_2\}$, como só existe um arco este será o escolhido $e_k = e_2$,

$$e_2 \in \psi_1 \Rightarrow \Delta_{F(k)} = \Delta_{F(2)} = \Delta_2 = \min\{\underbrace{\Delta_{T(2)}}_{\Delta_3}, u_2 - x_2\} = \min\{1, 2 - 0\} = 1,$$

$\hat{N} = \{\hat{N} \cup F(2), T(2)\} = \{3, 2\}$ e $\hat{A} = \hat{A} \cup \{e_2\} = \{e_2\}$, (esta árvore é apresentada

na Figura 5.3a). Como $\{F(4), T(4)\} \not\subset \hat{N}$ voltar ao passo 1.

Passo 1 – (Determinar da árvore).

$\psi_1 = \{e_1\}$ e $\psi_2 = \phi$, como $\psi_1 \cup \psi_2 \neq \phi$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$$e_k = e_1,$$

$e_k = e_1, e_1 \in \psi_1 \Rightarrow \Delta_1 = \min\{1, 4 - 0\} = 1, \hat{N} = \hat{N} \cup \{F(1), T(1)\} = \{3, 2\} \cup \{1, 3, 2\},$

e $\hat{A} = \hat{A} \cup \{e_1\} = \{e_2\} \cup \{e_1\} = \{e_1, e_2\}$. Como $\{F(4), T(4)\} \subset \hat{N}$ seguir para o

passo 3.

Passo 3 – (Calcular os Fluxos).

O ciclo é $\{1, e_1, 2, e_2, 3, e_4, 1\}$, como $\bar{c}_4 < 0$, os fluxos serão alterados de

$\Delta_{F(4)} = \Delta_1 = 1$ (se o sentido do arco é igual ao do ciclo o fluxo será aumentado,

caso contrário será diminuído), o novo vector dos fluxos é $[x_1 \ x_2 \ x_3 \ x_4] = [1 \ 1 \ 4 \ 0]$. Regressar ao passo 1 do ALG 5.3, para encontrar um arco fora da condição.

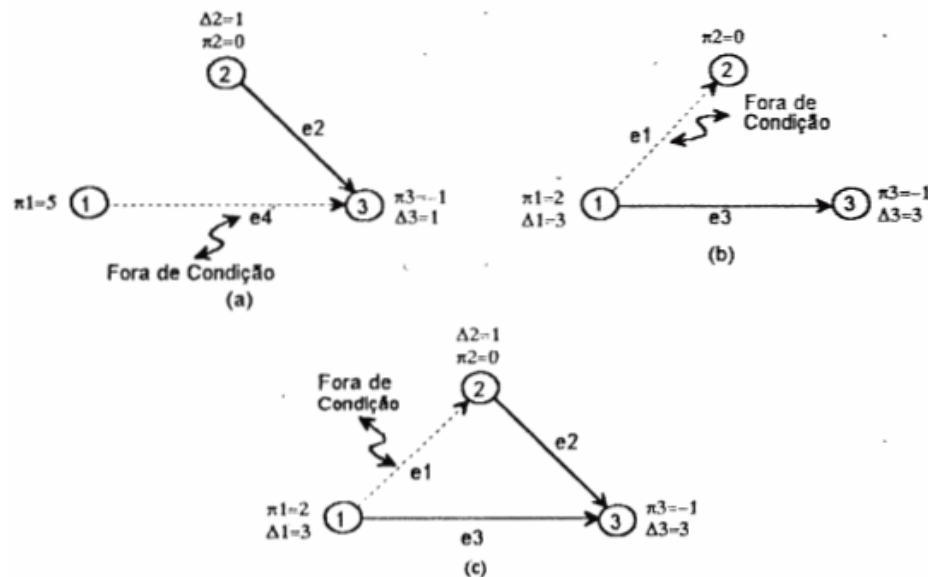


Figura 5.3 – Árvores associadas a fases do exemplo

ALG 5.3

Passo 1 – (Encontrar um arco fora da Condição).

Os lucros reduzidos são os mesmos que foram calculados anteriormente. Os números de condição de cada arco são agora:

$$x_1 = 1; \quad \bar{c}_1 = 4 \quad \Rightarrow \quad K_1 = 3 \rightarrow \text{fora da condição}$$

$$x_2 = 1; \quad \bar{c}_2 = 0 \quad \Rightarrow \quad K_2 = 0 \rightarrow \text{na condição}$$

$$x_3 = 4; \quad \bar{c}_3 = 3 \quad \Rightarrow \quad K_3 = 0 \rightarrow \text{na condição}$$

$$x_4 = 0; \quad \bar{c}_4 = -4 \quad \Rightarrow \quad K_4 = 0 \rightarrow \text{na condição}$$

$$\sum_{j=1}^4 K_j = 3$$

Agora só existe o arco e_1 fora da condição, obviamente $e_s = e_1$, passar à fase primal.

ALG 5.1

Passo 0 – (Iniciar).

Como $\bar{c}_1 > 0$ então $\hat{N} = \{F(1)\} = \{1\}$, $\Delta_{F(1)} = u_1 - x_1 \Rightarrow \Delta_1 = 3$ e o conjunto dos arcos $\hat{A} = \phi$.

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 \cup \psi_2 = \phi$, seguir para a fase dual.

ALG 5.2

Passo 1 – (Determinar os arcos incidentes na árvore).

Determinar ψ_3 e ψ_4 .

$$\psi_3 = \{e_j : T(j) \in \hat{N}, F(j) \notin \hat{N}, e \bar{c}_j < 0\} = \phi, e$$

$$\psi_4 = \{e_j : T(j) \notin \hat{N}, F(j) \in \hat{N}, e \bar{c}_j > 0\} = \{e_1, e_3\}.$$

Passo 2 – (Determinar a alteração máxima possível).

$$\phi = \min\{|\bar{c}_1|, |\bar{c}_3|\} = \min\{4, 3\} = 3.$$

Passo 3 – (Reduzir as variáveis duais).

$\pi_1 = \pi_1 - \theta = 5 - 3 = 2$ (o nó 1 é o único pertencente ao conjunto \hat{N}). As novas

variáveis duais são $[\pi_1 \quad \pi_2 \quad \pi_3] = [2 \quad 0 \quad -1]$. E os lucros reduzidos vêm:

$$\bar{c}_1 = \pi_{F(1)} - \pi_{T(1)} - c_1 = 2 - 0 - 1 = 1$$

$$\bar{c}_2 = \pi_{F(2)} - \pi_{T(2)} - c_2 = 0 - (-1) - 1 = 0$$

$$\bar{c}_3 = \pi_{F(3)} - \pi_{T(3)} - c_3 = 2 - (-1) - 3 = 0$$

$$\bar{c}_4 = \pi_{F(4)} - \pi_{T(4)} - c_4 = 2 - (-1) - 10 = -7$$

Como e_1 está fora da condição, voltar ao passo 0 do ALG 5.1 (fase primal).

ALG 5.1

Passo 0 – (Iniciar).

$$\hat{N} = \{1\}, \Delta_1 = 4 - 1, \hat{A} = \phi$$

Passo 1 – (Determinar candidatos para a árvore).

$$\psi_1 = \phi, \psi_2 = \{3\} \text{ como } \psi_1 \cup \psi_2 \neq \phi, \text{ seguir para o passo 2.}$$

Passo 2 – (Juntar à árvore um novo arco).

$$\psi_1 \cup \psi_2 = \{e_3\}, \text{ logo } e_k = e_3, e_3 \in \psi_2 \Rightarrow \Delta_{T(3)} = \Delta_3 = \min\{\Delta_{F(3)}, x_3\} = \min\{3, 4\} = 3,$$

$\hat{N} = \{1\} \cup \{3\} = \{1, 3\}$ e $\hat{A} = \{e_3\}$, (esta árvore é apresentada na Figura 5.3b). Como

$\{F(1), T(1)\} \not\subset \hat{N}$ seguir para o passo 1.

Passo 1 – (Determinar candidatos para a árvore).

$$\psi_1 = \{e_2\}, \psi_2 = \phi \text{ como } \psi_1 \cup \psi_2 \neq \phi, \text{ seguir para o passo 2.}$$

Passo 2 – (Juntar à árvore um novo arco).

$$e_k = e_2, e_2 \in \psi_1 \Rightarrow \Delta_2 = \min\{1, 3\} = 1,$$

$$\hat{N} = \hat{N} \cup \{F(2), T(2)\} = \{1, 3\} \cup \{2, 3\} = \{1, 2, 3\} \text{ e } \hat{A} = \hat{A} \cup \{e_2\} = \{e_3\} \cup \{e_2\} = \{e_2, e_3\}$$

(esta árvore é apresentada na Figura 5.3c). Como $\{F(1), T(1)\} \subset \hat{N}$ seguir para o passo 3.

Passo 3 –

O ciclo é $\{2, e_2, 3, e_3, 1, e_1, 2\}$, como $\bar{c}_1 > 0$, os fluxos serão alterados de

$\Delta_{T(1)} = \Delta_2 = 1$ (se o sentido do arco é igual ao do ciclo o fluxo será aumentado,

caso contrário será diminuído), e tomarão os valores

$$[x_1 \ x_2 \ x_3 \ x_4] = [2 \ 2 \ 3 \ 0].$$

Realizado o ciclo regressar ao passo 1 do ALG 5.3, verificar se ainda há algum arco fora da condição.

ALG 5.3

Passo 1 – (Encontrar um arco fora da Condição).

De acordo com os lucros reduzidos calculados no passo 3 do ALG 5.2, os números de condição de cada arco são agora:

$$x_1 = 2; \quad \bar{c}_1 = 1 \quad \Rightarrow \quad K_1 = 2 \rightarrow \text{fora da condição}$$

$$x_2 = 2; \quad \bar{c}_2 = 0 \quad \Rightarrow \quad K_2 = 0 \rightarrow \text{na condição}$$

$$x_3 = 3; \quad \bar{c}_3 = 0 \quad \Rightarrow \quad K_3 = 0 \rightarrow \text{na condição}$$

$$x_4 = 0; \quad \bar{c}_4 = -7 \quad \Rightarrow \quad K_4 = 0 \rightarrow \text{na condição}$$

$$\sum_{j=1}^4 K_j = 2$$

O arco e_1 continua fora da condição, portanto $e_s = e_1$ seguir para a fase primal.

ALG 5.1

Passo 0 – (Iniciar).

Como $\bar{c}_1 > 0$ então $\hat{N} = \{F(1)\} = \{1\}$, $\Delta_{F(1)} = u_1 - x_1 \Rightarrow \Delta_1 = 2$ e o conjunto dos arcos $\hat{A} = \phi$.

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \phi, \psi_2 = \{e_3\}$ como $\psi_1 \cup \psi_2 \neq \phi$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$$\psi_1 \cup \psi_2 = \{e_3\}, e_k = e_3, \quad e_3 \in \psi_2 \Rightarrow \Delta_{T(3)} = \Delta_3 = \min[\Delta_{F(3)}, x_3] = \min[3, 3] = 3,$$

$\hat{N} = \{1\} \cup [1, 3] = \{1, 3\}$ e $\hat{A} = \{e_3\}$. Como $\{F(1), T(1)\} \not\subset \hat{N}$ voltar para o passo 1.

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 \cup \psi_2 = \phi$, seguir para a fase dual.

ALG 5.2

Passo 1 – (Determinar os arcos incidentes na árvore).

$$\psi_3 = \phi \text{ e } \psi_4 = \{e_1\}.$$

Passo 2 – (Determinar a alteração máxima possível).

$$\phi = \min[\bar{c}_1] = 1.$$

Passo 3 – (Reduzir as variáveis duais).

$$\pi_1 = \pi_1 - \theta = 2 - 1 = 1$$

$$\pi_3 = \pi_3 - \theta = -1 - 1 = -2$$

As novas variáveis duais são $[\pi_1 \ \pi_2 \ \pi_3] = [1 \ 0 \ -2]$ e os lucros reduzidos:

$$\bar{c}_1 = \pi_{F(1)} - \pi_{T(1)} - c_1 = 1 - 0 - 1 = 0$$

$$\bar{c}_2 = \pi_{F(2)} - \pi_{T(2)} - c_2 = 0 - (-2) - 1 = 1$$

$$\bar{c}_3 = \pi_{F(3)} - \pi_{T(3)} - c_3 = 1 - (-2) - 3 = 0$$

$$\bar{c}_4 = \pi_{F(4)} - \pi_{T(4)} - c_4 = 1 - (-2) - 10 = -7$$

Como $\bar{c}_1 = \theta = 1$ então $e_s = e_1$ fica na condição, voltar ao passo 1 do ALG 5.3.

ALG 5.3

Passo 1 – (Encontrar um arco fora da Condição).

De acordo com os lucros reduzidos calculados no passo anterior, os números de condição de cada arco são agora:

$$x_1 = 2; \bar{c}_1 = 0 \Rightarrow K_1 = 2 \rightarrow \text{na condição}$$

$$x_2 = 2; \bar{c}_2 = 1 \Rightarrow K_2 = 0 \rightarrow \text{na condição}$$

$$x_3 = 3; \bar{c}_3 = 0 \Rightarrow K_3 = 0 \rightarrow \text{na condição}$$

$$x_4 = 0; \bar{c}_4 = -7 \Rightarrow K_4 = 0 \rightarrow \text{na condição}$$

$$\sum_{j=1}^4 K_j = 0$$

Todos os arcos estão na condição. Terminar com a conclusão de que o óptimo foi alcançado.

Para iniciar o algoritmo da condição pelo método do nó e arcos artificiais a que proceder de acordo com:

- 1º. Acrescentar um nó artificial à árvore cujo requisito é igual à soma dos requisitos dos outros nós.
- 2º. Criar arcos artificiais dos nós com requisito positivo para o nó artificial e criar arcos artificiais dos nós com requisito negativo para o nó artificial.
- 3º. Considerar os arcos criados em 2º que chegam ao nó artificial com um custo zero e capacidade de transporte infinita, $[0, \infty]$, e os outros arcos, criados igualmente em 2º, que partem do nó artificial com custo infinito e capacidade de transporte infinita, $[\infty, \infty]$.
- 4º. Transferir os fluxos pelos arcos fictícios, atribuindo aos fluxos dos arcos do problema original o valor zero.

Esta iniciação corresponde a uma expansão da rede do problema original. A rede expandida é representada na Figura 5.4, e os cálculos são indicados de seguida para os dois primeiros passos do algoritmo da condição.

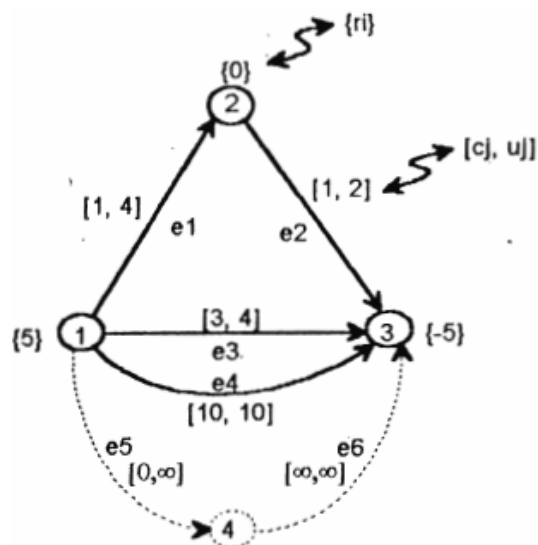


Figura 5.4 – Árvore para o exemplo, nó 4 artificial.

Seja M um número positivo, mas muito grande, “Big M method”, ou considere no que se segue que $M \rightarrow +\infty$.

ALG 5.3

Passo 0 – (Iniciar).

$$\text{Fluxos } [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6] = [0 \quad 0 \quad 0 \quad 0 \quad 5 \quad 5].$$

$$\text{Potenciais } [\pi_1 \quad \pi_2 \quad \pi_3 \quad \pi_4] = [0 \quad 0 \quad 0 \quad 0].$$

Passo 1 – (Encontrar um arco fora da Condição)

Lucros reduzidos de cada arco:

$$\begin{aligned} \bar{c}_1 &= \pi_{F(1)} - \pi_{T(1)} - c_1 = 0 - 0 - 1 = -1 \\ \bar{c}_2 &= \pi_{F(2)} - \pi_{T(2)} - c_2 = 0 - 0 - 1 = -1 \\ \bar{c}_3 &= \pi_{F(3)} - \pi_{T(3)} - c_3 = 0 - 0 - 3 = -3 \\ \bar{c}_4 &= \pi_{F(4)} - \pi_{T(4)} - c_4 = 0 - 0 - 10 = -10 \\ \bar{c}_5 &= \pi_{F(5)} - \pi_{T(5)} - c_5 = 0 - 0 - 0 = 0 \\ \bar{c}_6 &= \pi_{F(6)} - \pi_{T(6)} - c_6 = 0 - 0 - M = -M \end{aligned}$$

Os números de condição são

$$\begin{aligned} x_1 = 0; \quad \bar{c}_1 = -1 &\Rightarrow K_1 = 0 \rightarrow \text{na condição} \\ x_2 = 0; \quad \bar{c}_2 = -1 &\Rightarrow K_2 = 0 \rightarrow \text{na condição} \\ x_3 = 0; \quad \bar{c}_3 = -3 &\Rightarrow K_3 = 0 \rightarrow \text{na condição} \\ x_4 = 0; \quad \bar{c}_4 = -10 &\Rightarrow K_4 = 0 \rightarrow \text{na condição} \\ x_5 = 5; \quad \bar{c}_5 = 0 &\Rightarrow K_5 = 0 \rightarrow \text{na condição} \\ x_6 = 5; \quad \bar{c}_6 = -M &\Rightarrow K_6 = 5 \rightarrow \text{fora da condição} \end{aligned}$$

$$\sum_{j=1}^6 K_j = 5$$

O arco e_6 fora da condição e o algoritmo continua. Observe que pelo facto dos potenciais dos nós serem nulos os lucros reduzidos são o simétrico dos custos unitários de transporte nos arcos, sendo estes custos não negativos só o arco com custo unitário infinito está fora de condição. Seguidamente, será feita de novo a resolução do problema anterior alterando os requisitos dos nós e usando o método de iniciação por nó e arcos artificiais, pelo que fica ao cuidado do leitor a continuação desta resolução.

5.6- Exemplo de iniciar o algoritmo da condição por nó e arcos artificiais

Uma empresa pretende proceder ao planeamento de uma subcontratação para transporte de 10 geradores construídos na fábrica de Lisboa nó 1, para um cliente localizado em Luanda nó 3, existindo a alternativa de enviar os transformadores para a África do Sul nó 2 e daí para Luanda. O grafo da rede para o problema é representado na Figura 5.5, sendo o nó 2 o nó de transbordo situado na África do Sul e o nó 4, um nó fictício para iniciar a solução do problema com uma solução básica admissível pelo método dos arcos e nó artificial para o algoritmo da condição.

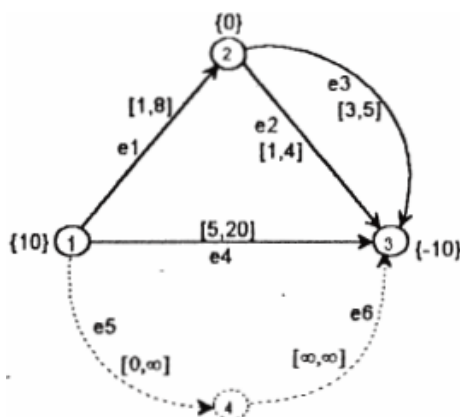


Figura 5.5 – Grafo do problema

Problema na forma canónica:

$$\begin{aligned} \min & x_1 + x_2 + 3x_3 + 5x_4 + \\ \text{s.a} & \\ & x_1 + x_4 - x_5 = 10 \\ & x_2 + x_3 = 0 \\ & -x_2 - x_3 - x_4 - x_5 = -10 \\ & -x_5 + x_6 = 0 \\ & 0 \leq x_1 \leq 8, 0 \leq x_2 \leq 4, 0 \leq x_3 \leq 5, 0 \leq x_4 \leq 20. \end{aligned}$$

ALG 5.3

Passo 0 – (Iniciar).

$$\text{Potenciais } [\pi_1 \ \pi_2 \ \pi_3 \ \pi_4] = [0 \ 0 \ 0 \ 0].$$

$$\text{Fluxos } [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6] = [0 \ 0 \ 0 \ 0 \ 10 \ 10].$$

Passo 1 – (Encontrar um arco fora da Condição).

Lucros reduzidos de cada arco:

$$\begin{aligned} \bar{c}_1 &= \pi_{F(1)} - \pi_{T(1)} - c_1 = 0 - 0 - 1 = -1 \\ \bar{c}_2 &= \pi_{F(2)} - \pi_{T(2)} - c_2 = 0 - 0 - 1 = -1 \\ \bar{c}_3 &= \pi_{F(3)} - \pi_{T(3)} - c_3 = 0 - 0 - 3 = -3 \\ \bar{c}_4 &= \pi_{F(4)} - \pi_{T(4)} - c_4 = 0 - 0 - 5 = -5 \\ \bar{c}_5 &= \pi_{F(5)} - \pi_{T(5)} - c_5 = 0 - 0 - 0 = 0 \\ \bar{c}_6 &= \pi_{F(6)} - \pi_{T(6)} - c_6 = 0 - 0 - M = -M \end{aligned}$$

E os números de condição

$$\begin{aligned} x_1 = 0; \quad \bar{c}_1 = -1 & \Rightarrow K_1 = 0 \rightarrow \text{na condição} \\ x_2 = 0; \quad \bar{c}_2 = -1 & \Rightarrow K_2 = 0 \rightarrow \text{na condição} \\ x_3 = 0; \quad \bar{c}_3 = -3 & \Rightarrow K_3 = 0 \rightarrow \text{na condição} \\ x_4 = 0; \quad \bar{c}_4 = -5 & \Rightarrow K_4 = 0 \rightarrow \text{na condição} \\ x_5 = 10; \quad \bar{c}_5 = 0 & \Rightarrow K_5 = 0 \rightarrow \text{na condição} \\ x_6 = 10; \quad \bar{c}_6 = -M & \Rightarrow K_6 = 10 \rightarrow \text{fora da condição} \end{aligned}$$

$$\sum_{j=1}^6 K_j = 10$$

Arco e_6 está fora da condição, seguir para a fase primal.

ALG 5.1

Passo 0 – (Iniciar).

Como $\bar{c}_6 < 0$ então $\hat{N} = \{T(6)\} = \{3\}$, $\Delta_{T(6)} = \Delta_3 = x_6 = 10$ e $\hat{A} = \phi$.

Passo 1 – (Determinar candidatos para a árvore).

Como $\psi_1 \cup \psi_2 = \phi$, seguir para a fase dual.

ALG 5.2

Passo 1 – (Determinar os arcos incidentes na árvore).

$\psi_3 = \{e_2, e_3, e_4, e_6\}$ e $\psi_4 = \phi$.

Passo 2 – (Determinar a alteração máxima possível).

$\phi = \min[|-1|, |-3|, |-5|, |-M|] = 1$.

Passo 3 – (Reduzir as variáveis duais).

$$\pi_3 = 0 - 1 = -1$$

o vector dos potenciais é $[\pi_1 \ \pi_2 \ \pi_3 \ \pi_4] = [0 \ 0 \ -1 \ 0]$, calcular os lucros

reduzidos:

$$\bar{c}_1 = \pi_{F(1)} - \pi_{T(1)} - c_1 = 0 - 0 - 1 = -1$$

$$\bar{c}_2 = \pi_{F(2)} - \pi_{T(2)} - c_2 = 0 - (-1) - 1 = 0$$

$$\bar{c}_3 = \pi_{F(3)} - \pi_{T(3)} - c_3 = 0 - (-1) - 3 = -2$$

$$\bar{c}_4 = \pi_{F(4)} - \pi_{T(4)} - c_4 = 0 - (-1) - 5 = -4$$

$$\bar{c}_5 = \pi_{F(5)} - \pi_{T(5)} - c_5 = 0 - 0 - 0 = 0$$

$$\bar{c}_6 = \pi_{F(6)} - \pi_{T(6)} - c_6 = 0 - (-1) - M = -M + 1$$

Voltar ao passo 1 do ALG 5.1.

ALG 5.1

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \{e_2\}$, e $\psi_2 = \phi$. Como $\psi_1 \cup \psi_2 \neq \phi$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$e_k = e_2$. Como $e_2 \in \psi_1 \rightarrow \Delta_{F(k)} = \Delta_2 = \min[\Delta_3, u_2 - x_2] = \min[10, 4] = 4$,

$\hat{N} = \{2, 3\}$ e $\hat{A} = \{e_2\}$. Como $\{F(6), T(6)\} \not\subset \hat{N}$ voltar para o passo 1.

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \psi_2 = \phi$. Como $\psi_1 \cup \psi_2 = \phi$, seguir para a fase dual.

ALG 5.2

Passo 1 – (Determinar os arcos incidentes na árvore).

$\psi_3 = \{e_1, e_4, e_6\}$ e $\psi_4 = \phi$.

Passo 2 – (Determinar a alteração máxima possível) $\phi = \min[|-1|, |-4|, |-M+1|] = 1$.

Passo 3 – (Reduzir as variáveis duais).

$\pi_3 = -1 - 1 = -2$ e $\pi_2 = 0 - 1 = -1$. E para os potenciais vem

$[\pi_1 \ \pi_2 \ \pi_3 \ \pi_4] = [0 \ -1 \ -2 \ 0]$, tomando os lucros reduzidos os valores:

$$\bar{c}_1 = \pi_{F(1)} - \pi_{T(1)} - c_1 = 0 - (-1) - 1 = 0 \quad K_1 = 0$$

$$\bar{c}_2 = \pi_{F(2)} - \pi_{T(2)} - c_2 = -1 - (-2) - 1 = 0 \quad K_2 = 0$$

$$\bar{c}_3 = \pi_{F(3)} - \pi_{T(3)} - c_3 = -1 - (-2) - 3 = -2 \quad K_3 = 0$$

$$\bar{c}_4 = \pi_{F(4)} - \pi_{T(4)} - c_4 = 0 - (-2) - 5 = -3 \quad K_4 = 0$$

$$\bar{c}_5 = \pi_{F(5)} - \pi_{T(5)} - c_5 = 0 - 0 - 0 = 0 \quad K_5 = 0$$

$$\bar{c}_6 = \pi_{F(6)} - \pi_{T(6)} - c_6 = 0 - (-2) - M = -M + 2 \quad K_6 = 10$$

$$\overline{\sum_{j=1}^6 K_j} = 10$$

Voltar ao passo 1 do ALG 5.1.

ALG 5.1

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \{e_1\}$, e $\psi_2 = \phi$. Como $\psi_1 \cup \psi_2 \neq \phi$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$e_k = e_1$. Como $e_1 \in \psi_1 \rightarrow \Delta_{F(k)} = \Delta_1 = \min[\Delta_2, u_1 - x_1] = \min[4, 8] = 4$,

$\hat{N} = \{1, 2, 3\}$ e $\hat{A} = \{e_1, e_2\}$. Como $\{F(6), T(6)\} \not\subset \hat{N}$ voltar para o passo 1.

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \phi$ e $\psi_2 = \{e_5\}$. Como $\psi_1 \cup \psi_2 \neq \phi$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$e_k = e_5$. Como $e_5 \in \psi_2 \rightarrow \Delta_{T(k)} = \Delta_4 = \min[\Delta_1, x_5] = \min[4, 10] = 4$,

$\hat{N} = \{1, 2, 3, 4\}$ e $\hat{A} = \{e_1, e_2, e_5\}$. Como $\{F(6), T(6)\} \subset \hat{N}$ seguir para o passo 3.

Passo 3 – (Calcular os Fluxos).

O ciclo $C = \{4, e_5, 1, e_1, 2, e_2, 3, e_6, 4\}$ e está representado na Figura 5.6, como

$\bar{c}_6 < 0$, $\Delta_4 = 4$, $[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6] = [4 \ 4 \ 0 \ 0 \ 6 \ 6]$. Voltar ao passo 1 do ALG 5.3.

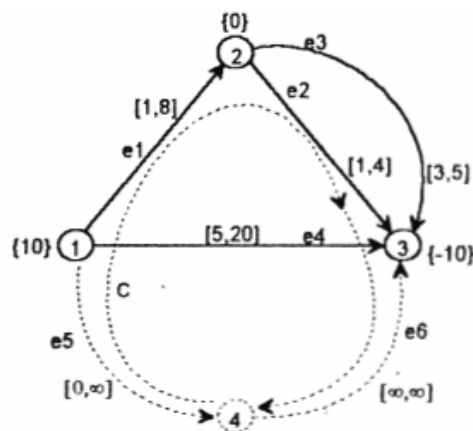


Figura 5.6 – Ciclo C

ALG 5.3

Passo 1 – (Encontrar um arco fora da Condição).

De acordo com os novos fluxos e os últimos lucros reduzidos os números de condição são os seguintes:

$$\begin{aligned}
 x_1 = 4; \quad \bar{c}_1 = 0 & \Rightarrow K_1 = 0 \rightarrow \text{na condição} \\
 x_2 = 4; \quad \bar{c}_2 = 0 & \Rightarrow K_2 = 0 \rightarrow \text{na condição} \\
 x_3 = 0; \quad \bar{c}_3 = -2 & \Rightarrow K_3 = 0 \rightarrow \text{na condição} \\
 x_4 = 0; \quad \bar{c}_4 = -3 & \Rightarrow K_4 = 0 \rightarrow \text{na condição} \\
 x_5 = 6; \quad \bar{c}_5 = 0 & \Rightarrow K_5 = 0 \rightarrow \text{na condição} \\
 x_6 = 6; \quad \bar{c}_6 = -M + 2 & \Rightarrow K_6 = 6 \rightarrow \text{fora da condição} \\
 \sum_{j=1}^6 K_j = 6 &
 \end{aligned}$$

Arco e_6 está fora da condição, seguir para a fase primal.

ALG 5.1

Passo 0 – (Iniciar). Como $\bar{c}_6 < 0$ então $\hat{N} = \{T(6)\} = \{3\}$, $\Delta_{T(6)} = \Delta_3 = x_6 = 6$ e o conjunto dos arcos $\hat{A} = \phi$.

Passo 1 – (Determinar candidatos para a árvore), $\psi_1 = \psi_2 = \phi$. Como $\psi_1 \cup \psi_2 = \phi$, seguir para a fase dual.

ALG 5.2

Passo 1 – (Determinar os arcos incidentes na árvore).

$$\psi_3 = \{e_3, e_4, e_6\} \text{ e } \psi_4 = \phi.$$

Passo 2 – (Determinar a alteração máxima possível).

$$\phi = \min[|-2|, |-3|, |-M + 2|] = 2.$$

Passo 3 – (Reduzir as variáveis duais).

$$\pi_3 = -2 - 2 = -4 .$$

E para os potenciais vem $[\pi_1 \ \pi_2 \ \pi_3 \ \pi_4] = [0 \ -1 \ -4 \ 0]$, tomando os

lucros reduzidos os valores:

$$\begin{aligned} \bar{c}_1 &= \pi_{F(1)} - \pi_{T(1)} - c_1 = 0 - (-1) - 1 = 0 & K_1 &= 0 \\ \bar{c}_2 &= \pi_{F(2)} - \pi_{T(2)} - c_2 = -1 - (-4) - 1 = 2 & K_2 &= 0 \\ \bar{c}_3 &= \pi_{F(3)} - \pi_{T(3)} - c_3 = -1 - (-4) - 3 = 0 & K_3 &= 0 \\ \bar{c}_4 &= \pi_{F(4)} - \pi_{T(4)} - c_4 = 0 - (-4) - 5 = -1 & K_4 &= 0 \\ \bar{c}_5 &= \pi_{F(5)} - \pi_{T(5)} - c_5 = 0 - 0 - 0 = 0 & K_5 &= 0 \\ \bar{c}_6 &= \pi_{F(6)} - \pi_{T(6)} - c_6 = 0 - (-4) - M = -M + 4 & K_6 &= 6 \end{aligned}$$

$$\overline{\sum_{j=1}^6 K_j} = 6$$

Voltar ao passo 1 do ALG 5.1.

ALG 5.1

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \{e_3\}$, e $\psi_2 = \phi$. Como $\psi_1 \cup \psi_2 \neq \phi$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$e_k = e_3$. Como $e_3 \in \psi_1 \rightarrow \Delta_{F(k)} = \Delta_2 = \min[\Delta_3, u_3 - x_3] = \min[6, 5] = 5$,

$\hat{N} = \{2, 3\}$ e $\hat{A} = \{e_3\}$. Como $\{F(6), T(6)\} \not\subset \hat{N}$ voltar para o passo 1.

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \{e_1\}$, e $\psi_2 = \phi$. Como $\psi_1 \cup \psi_2 \neq \phi$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$e_k = e_1$. Como $e_1 \in \psi_1 \rightarrow \Delta_{F(k)} = \Delta_1 = \min[\Delta_2, u_1 - x_1] = \min[5, 8 - 4] = 4$,

$\hat{N} = \{1, 2, 3\}$ e $\hat{A} = \{e_1, e_3\}$. Como $\{F(6), T(6)\} \not\subset \hat{N}$ seguir para o passo 1.

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \emptyset$, e $\psi_2 = \{e_5\}$. Como $\psi_1 \cup \psi_2 \neq \emptyset$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$e_k = e_5$. Como $e_5 \in \psi_2 \rightarrow \Delta_{T(k)} = \Delta_4 = \min[\Delta_1, x_5] = \min[4, 6] = 4$,

$\hat{N} = \{1, 2, 3, 4\}$ e $\hat{A} = \{e_1, e_3, e_5\}$. Como $\{F(6), T(6)\} \subset \hat{N}$ seguir para o passo 3.

Passo 3 – (Calcular os Fluxos).

O ciclo é

$$C = \{4, e_5, 1, e_1, 2, e_3, 3, e_6, 4\}$$

e está representado na Figura 5.7, como $\bar{c}_6 < 0$, os fluxos dos arcos do ciclo virão alterados de $\Delta_4 = 4$ e tomam os valores $[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6] = [8 \ 4 \ 4 \ 0 \ 2 \ 2]$. Voltar ao passo 1 do ALG 5.3.

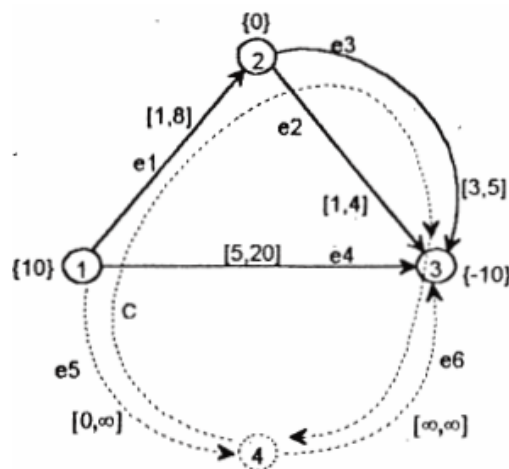


Figura 5.7 – Ciclo C

ALG 5.3

Passo 1 – (Encontrar um arco fora da Condição).

De acordo com os novos fluxos e os últimos lucros reduzidos os números de condição são os seguintes:

$$\begin{aligned}
 x_1 = 8; \quad \bar{c}_1 = 8 & \quad \Rightarrow \quad K_1 = 0 \rightarrow \text{na condição} \\
 x_2 = 4; \quad \bar{c}_2 = 2 & \quad \Rightarrow \quad K_2 = 0 \rightarrow \text{na condição} \\
 x_3 = 4; \quad \bar{c}_3 = 0 & \quad \Rightarrow \quad K_3 = 0 \rightarrow \text{na condição} \\
 x_4 = 0; \quad \bar{c}_4 = -1 & \quad \Rightarrow \quad K_4 = 0 \rightarrow \text{na condição} \\
 x_5 = 2; \quad \bar{c}_5 = 0 & \quad \Rightarrow \quad K_5 = 0 \rightarrow \text{na condição} \\
 x_6 = 2; \quad \bar{c}_6 = -M + 4 & \quad \Rightarrow \quad K_6 = 2 \rightarrow \text{fora da condição} \\
 \sum_{j=1}^6 K_j = 2 &
 \end{aligned}$$

Arco e_6 está fora da condição, seguir para a fase primal.

ALG 5.1

Passo 0 – (Iniciar).

Como $\bar{c}_6 < 0$ então $\hat{N} = \{T(6)\} = \{3\}$, $\Delta_{T(6)} = \Delta_3 = x_6 = 2$ e o conjunto dos arcos

$$\hat{A} = \phi.$$

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \{e_3\}$, e $\psi_2 = \phi$ como $\psi_1 \cup \psi_2 \neq \phi$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$e_k = e_3$. Como $e_3 \in \psi_1 \Rightarrow \Delta_{F(k)} = \Delta_2 = \min[\Delta_3, u_3 - x_3] = \min[2, 1] = 1$,

$\hat{N} = \{2, 3\}$ e $\hat{A} = \{e_3\}$. Como $\{F(6), T(6)\} \not\subset \hat{N}$ voltar para o passo 1.

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \psi_2 = \phi$ como $\psi_1 \cup \psi_2 \neq \phi$, seguir para a fase dual.

ALG 5.2

Passo 1 – (Determinar os arcos incidentes na árvore).

$$\psi_3 = \{e_4, e_6\} \text{ e } \psi_4 = \phi.$$

Passo 2 – (Determinar a alteração máxima possível).

$$\phi = \min[|-1|, |-M + 4|] = 1.$$

Passo 3 – (Reduzir as variáveis duais).

$$\pi_3 = -4 - 1 = -5.$$

E para os potenciais vem $[\pi_1 \ \pi_2 \ \pi_3 \ \pi_4] = [0 \ -1 \ -5 \ 0]$, tomando os

lucros reduzidos os valores:

$$\begin{aligned} \bar{c}_1 &= \pi_{F(1)} - \pi_{T(1)} - c_1 = 0 - (-1) - 1 = 0 & K_1 &= 0 \\ \bar{c}_2 &= \pi_{F(2)} - \pi_{T(2)} - c_2 = -1 - (-5) - 1 = 3 & K_2 &= 0 \\ \bar{c}_3 &= \pi_{F(3)} - \pi_{T(3)} - c_3 = -1 - (-5) - 3 = 1 & K_3 &= 0 \\ \bar{c}_4 &= \pi_{F(4)} - \pi_{T(4)} - c_4 = 0 - (-5) - 5 = 0 & K_4 &= 0 \\ \bar{c}_5 &= \pi_{F(5)} - \pi_{T(5)} - c_5 = 0 - 0 - 0 = 0 & K_5 &= 0 \\ \bar{c}_6 &= \pi_{F(6)} - \pi_{T(6)} - c_6 = 0 - (-5) - M = -M + 5 & K_6 &= 2 \end{aligned}$$

$$\overline{\sum_{j=1}^6 K_j} = 2$$

Voltar ao passo 1 do ALG 5.1.

ALG 5.1

Passo 1 – (Determinar candidatos para a árvore).

$$\psi_1 = \{e_4\}, \text{ e } \psi_2 = \phi. \text{ Como } \psi_1 \cup \psi_2 \neq \phi, \text{ seguir para o passo 2.}$$

Passo 2 – (Juntar à árvore um novo arco).

$$e_k = e_4. \text{ Como } e_4 \in \psi_1 \rightarrow \Delta_{F(k)} = \Delta_1 = \min[\Delta_3, u_4 - x_4] = \min[2, 20] = 2,$$

$$\hat{N} = \{1, 3\} \text{ e } \hat{A} = \{e_4\}. \text{ Como } \{F(6), T(6)\} \not\subset \hat{N} \text{ seguir para o passo 1.}$$

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \phi$, e $\psi_2 = \{e_5\}$. Como $\psi_1 \cup \psi_2 \neq \phi$, seguir para o passo 2.

Passo 2 – (Juntar à árvore um novo arco).

$e_k = e_5$. Como $e_1 \in \psi_2 \rightarrow \Delta_{T(k)} = \Delta_4 = \min[\Delta_1, x_5] = \min[2, 2] = 2$,

$\hat{N} = \{1, 3, 4\}$ e $\hat{A} = \{e_4, e_5\}$. Como $\{F(6), T(6)\} \subset \hat{N}$ seguir para o passo 3.

Passo 3 – (Calcular os Fluxos).

O ciclo é $C = \{4, e_5, 1, e_4, 3, e_6, 4\}$ e está representado na Figura 5.8, como

$\bar{c}_6 < 0$, os fluxos dos arcos do ciclo virão alterados de $\Delta_4 = 2$ e tomam os valores $[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6] = [8 \ 4 \ 4 \ 2 \ 0 \ 0]$. Voltar ao passo 1 do ALG 5.3.

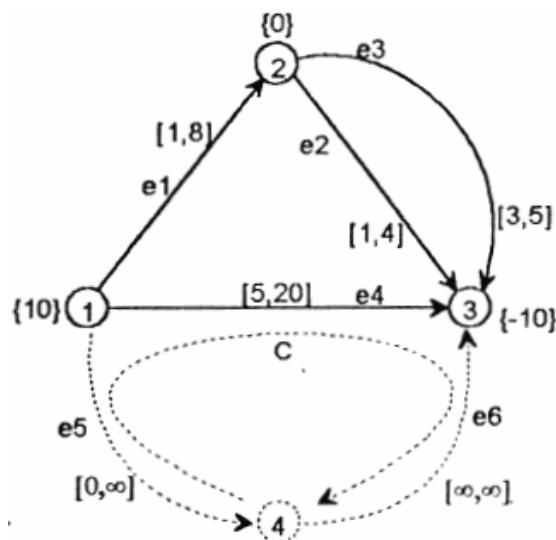


Figura 5.8 – Ciclo C

ALG 5.3

Passo 1 – (Encontrar um arco fora da Condição).

De acordo com os novos fluxos e os últimos lucros reduzidos os números de condição são os seguintes:

$$\begin{aligned} x_1 = 8; \quad \bar{c}_1 = 0 & \quad \Rightarrow \quad K_1 = 0 \rightarrow \text{na condição} \\ x_2 = 4; \quad \bar{c}_2 = 3 & \quad \Rightarrow \quad K_2 = 0 \rightarrow \text{na condição} \\ x_3 = 4; \quad \bar{c}_3 = 1 & \quad \Rightarrow \quad K_3 = 1 \rightarrow \text{fora da condição} \\ x_4 = 2; \quad \bar{c}_4 = 0 & \quad \Rightarrow \quad K_4 = 0 \rightarrow \text{na condição} \\ x_5 = 0; \quad \bar{c}_5 = 0 & \quad \Rightarrow \quad K_5 = 0 \rightarrow \text{na condição} \\ x_6 = 0; \quad \bar{c}_6 = -M + 4 & \quad \Rightarrow \quad K_6 = 0 \rightarrow \text{na condição} \end{aligned}$$

$$\sum_{j=1}^6 K_j = 1$$

Arco e_3 está fora da condição, seguir para a fase primal.

ALG 5.1

Passo 0 – (Iniciar).

Como $\bar{c}_3 > 0$ então $\hat{N} = \{F(3)\} = \{2\}$, $\Delta_{F(3)} = \Delta_2 = u_3 - x_3 = 1$ e o conjunto dos arcos $\hat{A} = \phi$.

Passo 1 – (Determinar candidatos para a árvore).

$\psi_1 = \psi_2 = \phi$ como $\psi_1 \cup \psi_2 \neq \phi$, seguir para a fase dual.

ALG 5.2

Passo 1 – (Determinar os arcos incidentes na árvore).

$\psi_3 = \phi$ e $\psi_4 = \{e_2, e_3\}$.

Passo 2 – (Determinar a alteração máxima possível).

$\phi = \min[|-3|, |-1|] = 1$.

Passo 3 – (Reduzir as variáveis duais).

$\pi_2 = -1 - 1 = -2$.

E para os potenciais vem $[\pi_1 \ \pi_2 \ \pi_3 \ \pi_4] = [0 \ -2 \ -5 \ 0]$, tomando os lucros reduzidos os valores:

$$\begin{aligned} \bar{c}_1 &= \pi_{F(1)} - \pi_{T(1)} - c_1 = 0 - (-2) - 1 = 1 & K_1 &= 0 \\ \bar{c}_2 &= \pi_{F(2)} - \pi_{T(2)} - c_2 = -2 - (-5) - 1 = 2 & K_2 &= 0 \\ \bar{c}_3 &= \pi_{F(3)} - \pi_{T(3)} - c_3 = -2 - (-5) - 3 = 0 & K_3 &= 1 \\ \bar{c}_4 &= \pi_{F(4)} - \pi_{T(4)} - c_4 = 0 - (-5) - 5 = 0 & K_4 &= 0 \\ \bar{c}_5 &= \pi_{F(5)} - \pi_{T(5)} - c_5 = 0 - 0 - 0 = 0 & K_5 &= 0 \\ \bar{c}_6 &= \pi_{F(6)} - \pi_{T(6)} - c_6 = 0 - (-5) - M = -M + 5 & K_6 &= 0 \end{aligned}$$

$$\sum_{j=1}^6 K_j = 1$$

Como $\bar{c}_3 = 0$, e_3 fica na condição, voltar ao passo 1 do ALG 5.3.

ALG 5.3

Passo 1 – (Encontrar um arco fora da Condição).

De acordo com os novos fluxos e os últimos lucros reduzidos os números de condição são os seguintes:

$$\begin{aligned} x_1 = 8; \quad \bar{c}_1 = 1 & \Rightarrow K_1 = 0 \rightarrow \text{na condição} \\ x_2 = 4; \quad \bar{c}_2 = 2 & \Rightarrow K_2 = 0 \rightarrow \text{na condição} \\ x_3 = 4; \quad \bar{c}_3 = 0 & \Rightarrow K_3 = 0 \rightarrow \text{na condição} \\ x_4 = 2; \quad \bar{c}_4 = 0 & \Rightarrow K_4 = 0 \rightarrow \text{na condição} \\ x_5 = 0; \quad \bar{c}_5 = 0 & \Rightarrow K_5 = 0 \rightarrow \text{na condição} \\ x_6 = 0; \quad \bar{c}_6 = -M & \Rightarrow K_6 = 0 \rightarrow \text{na condição} \end{aligned}$$

$$\sum_{j=1}^6 K_j = 0$$

Todos os arcos estão na condição de optimalidade, sendo os fluxos e os potenciais finais os seguintes:

$$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6] = [8 \ 4 \ 4 \ 2 \ 0 \ 0]$$

$$[\pi_1 \ \pi_2 \ \pi_3 \ \pi_4] = [0 \ -2 \ -5 \ 0].$$

Assim o problema dado é resolvido utilizando o algoritmo da condição. Em anexos a este capítulo, apresenta-se, em anexo 1 a listagem do programa de computador desenvolvido no âmbito deste trabalho, em anexo 2 os resultados dados pelo programa da resolução do problema dado e em anexo 3 a resolução de um segundo problema para teste do programa.

5.7- Convergência do Algoritmo da Condição

Para provar que este algoritmo converge num número finito de iterações assume-se que \mathbf{r} , \mathbf{u} , \mathbf{c} , \mathbf{x} , e $\boldsymbol{\pi}$ são vectores cujas coordenadas são inteiros. Assim, cada vez que se obtém um avanço e uma alteração do fluxo na fase primal, o número da condição de e_s é reduzido pelo menos de 1. Cada vez que se alteram as variáveis duais na fase dual, ou a árvore desenvolvida na fase primal seguinte tem pelo menos mais um nó (e um arco) ou o número da condição do sistema é reduzido de pelo menos 1. Por aplicação repetida das fases primal e dual sobre um arco fora da condição, não será possível desenvolver a árvore indefinidamente. Uma árvore geradora para o grafo pode ser produzida na fase primal. A árvore geradora juntamente com e_s permite construir um ciclo. Então, após um número finito de passos, $\sum_{e_j \in A} K_j$ será reduzido a zero.

6- CÓDIGO LSNNO, CASCATA HÍDRICA

Neste capítulo no âmbito da programação matemática em redes é abordado o problema da simulação computacional da gestão racional de albufeiras em cascata num horizonte de curto prazo e para suporte computacional é usada a aplicação informática LSNNO, “Large Scale Nonlinear Network Optimization”. O LSNNO foi desenvolvido no departamento de matemática “ND de la Paix” (Bélgica) e que está disponível em linguagem Fortran na Internet juntamente com um manual do utilizador. No entanto, o manual do utilizador não está em “word” pelo que se anexa uma versão deste manual no fim deste capítulo.

6.1- Simulação da Gestão de cascatas hídricas

Uma empresa de sistemas de energia eléctrica deve ter a preocupação de decidir com racionalidade a gestão dos recursos que gere, i.e., o parque de recursos que é o seu sistema a gerir, com o qual obtém os proveitos que justificam e asseguram a sua existência. O objectivo a atingir na produção da energia eléctrica e a maximização do lucro obtido com a venda da energia eléctrica dentro das condicionantes quer

regulamentadas quer técnicas. A venda de energia eléctrica pode ser realizada quer directamente aos consumidores por contratos bilaterais, quer num mercado de energia eléctrica onde a empresa tem que caracterizar a sua oferta também de acordo com a regulamentação para o mercado.

O objectivo de atingir, a maximização do lucro obtido com a venda da energia eléctrica, salvaguarda a viabilidade da empresa e a sua competitividade num mercado competitivo de energia eléctrica. Num aproveitamento hidroeléctrico a optimização da gestão dos recursos consiste em decidir a gestão da água que está disponível. Esta optimização sujeita a condicionantes que são impostas quer pelo equipamento, condicionantes técnicas, quer por razões ecológicas e de segurança, condicionantes políticas. Envolve a determinação do volume de água a turbinar mais proveitoso em cada período, tendo em conta o valor pago pela energia produzida nesses períodos, que é variável durante o período. A dificuldade em tomar as decisões mais adequadas cresce quando a empresa de produção de energia eléctrica tem que gerir mais de que um aproveitamento hídrico e esses aproveitamentos são em cascata. As variáveis ligadas aos trânsitos de volume de águas ficam relacionadas pelas equações de balanço das águas, complicando o problema e o gestor tem que recorrer a sistemas de suporte à tomada de decisão para conseguir ter uma gestão adequada, que atinja o melhor proveito.

O recurso à simulação computacional do modelo matemático do sistema a gerir e a técnicas de optimização, permitem gerar um sistema de informação para o suporte das decisões racionais do sistema a gerir. Estes sistemas de informação de gestão são obviamente de grande interesse para a viabilidade da empresa e a sua competitividade no mercado de energia eléctrica. Actualmente, não se pode conceber a exploração dos recursos de forma eficiente sem o uso deste tipo de simulação computacional.

6.2- Gestão de uma cascata hídrica no curto prazo

O estudo que é apresentado aborda um aproveitamento hídrico com albufeiras em cascata, que é um problema bem adaptado para ser tratado por programação matemática em redes cuja formulação é do tipo

$$\left. \begin{array}{l} \max f(\mathbf{x}) \\ \text{s.a } \mathbf{Ax} = \mathbf{r} \\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \end{array} \right\} \text{problema para a gestão da cascata no curto prazo}$$

Onde a função objectivo pode ser uma função não linear e os fluxos são os volumes de água quer em trânsito quer armazenados nas albufeiras.

Seja a configuração para o problema a do seguinte aproveitamento hídrico em cascata

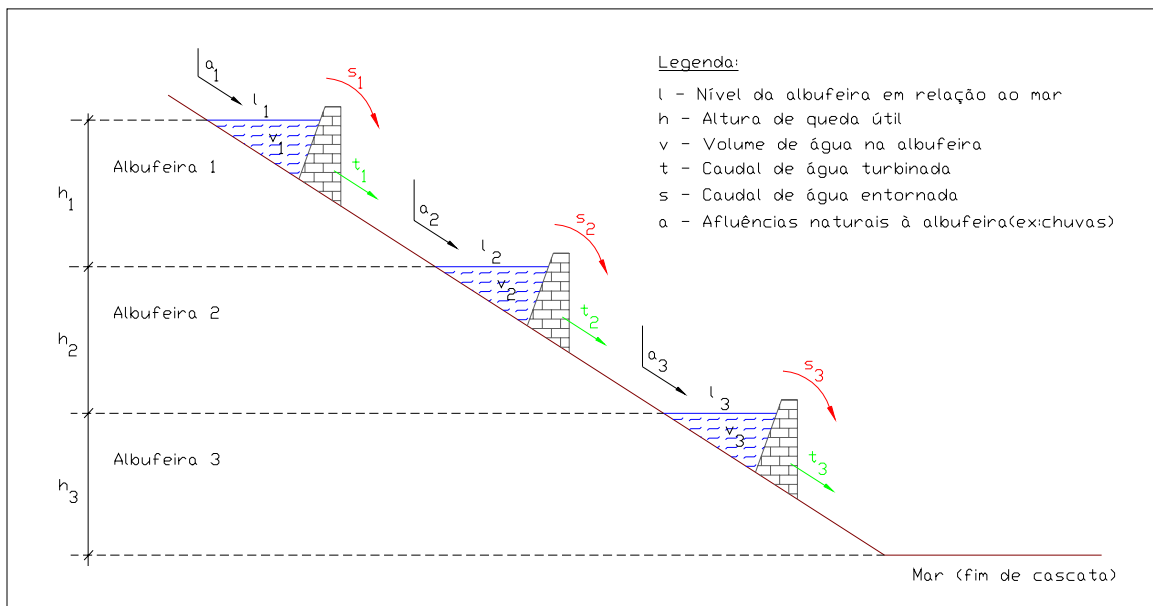


Figura 6.1 – Configuração do aproveitamento hídrico em cascata.

Tipicamente, o problema da gestão racional de albufeiras em cascata num horizonte de curto prazo considera as decisões tomadas em períodos horários durante um horizonte

temporal de alguns dias até uma semana. Portanto, é o horizonte de decisão do ciclo diária até ao semanal do uso de energia eléctrica por parte dos consumidores que é considerado neste problema. No entanto, é justificável em certas situações um período inferior, quer para obter uma melhor gestão do aproveitamento quer para considerar tempos de transito da água entre albufeiras que necessitem de reduzir a amplitude do período. Neste problema existem condicionantes determinadas pelas limitações físicas das albufeiras, das turbinas e condicionantes determinadas quer por razões de segurança das albufeiras quer por questões ambientais. Os dados que caracterizam a capacidade produtiva dos recursos ao longo do horizonte temporal são o volume inicial da água armazenado nas albufeiras e as séries de afluências aos reservatórios ao longo dos períodos do horizonte temporal.

6.3- Limites para as variáveis

Os volumes máximos e mínimos das albufeiras são determinados por razões quer ecológicas quer de segurança, cujo o estudo não é objectivo desta simulação.

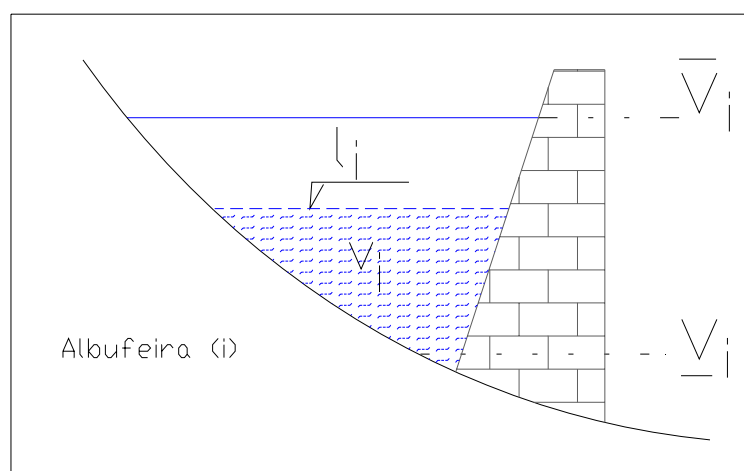


Figura 6.2 – Volume máximo e mínimo de uma albufeira nível da água numa albufeira.

Consequentemente, os volumes máximos e os volumes mínimos para as albufeiras são considerados dados do problema. Nas expressões indicadas de seguida o índice superior corresponde ao período e o inferior corresponde à albufeira. Quando o índice superior for nulo, período zero, corresponde a valores de iniciais das variáveis.

Dados

v_i^0	-	Volume inicial na albufeira i
a_i^k	-	Afluência à albufeira i no intervalo de tempo k
$\bar{v}_i, \underline{v}_i$	-	Volume mínimo e volume máximo permitido na albufeira i
$\bar{s}_i, \underline{s}_i$	-	Caudal mínimo e caudal máximo que é possível entornar na albufeira (i);
$\bar{t}_i, \underline{t}_i$	-	Caudal mínimo e caudal máximo que é possível turbinar na albufeira (i);

As limitações consideradas são:

$\underline{v}_i \leq v_i^k \leq \bar{v}_i$ - intervalo de variação do volume admissível na albufeira i

$\underline{s}_i \leq s_i^k \leq \bar{s}_i$ - intervalo de variação do caudal descarregado na albufeira i

$\underline{t}_i \leq t_i^k \leq \bar{t}_i$ - intervalo de variação dos caudal turbinado na albufeira i.

Na página seguinte na Figura 6.3 está representada a rede para o aproveitamento hídrico em cascata em estudo. A rede descreve o aproveitamento hidroelétrico em cascata composto por três albufeiras. O horizonte de exploração é uma semana com períodos de uma hora, isto é, 168 horas de exploração.

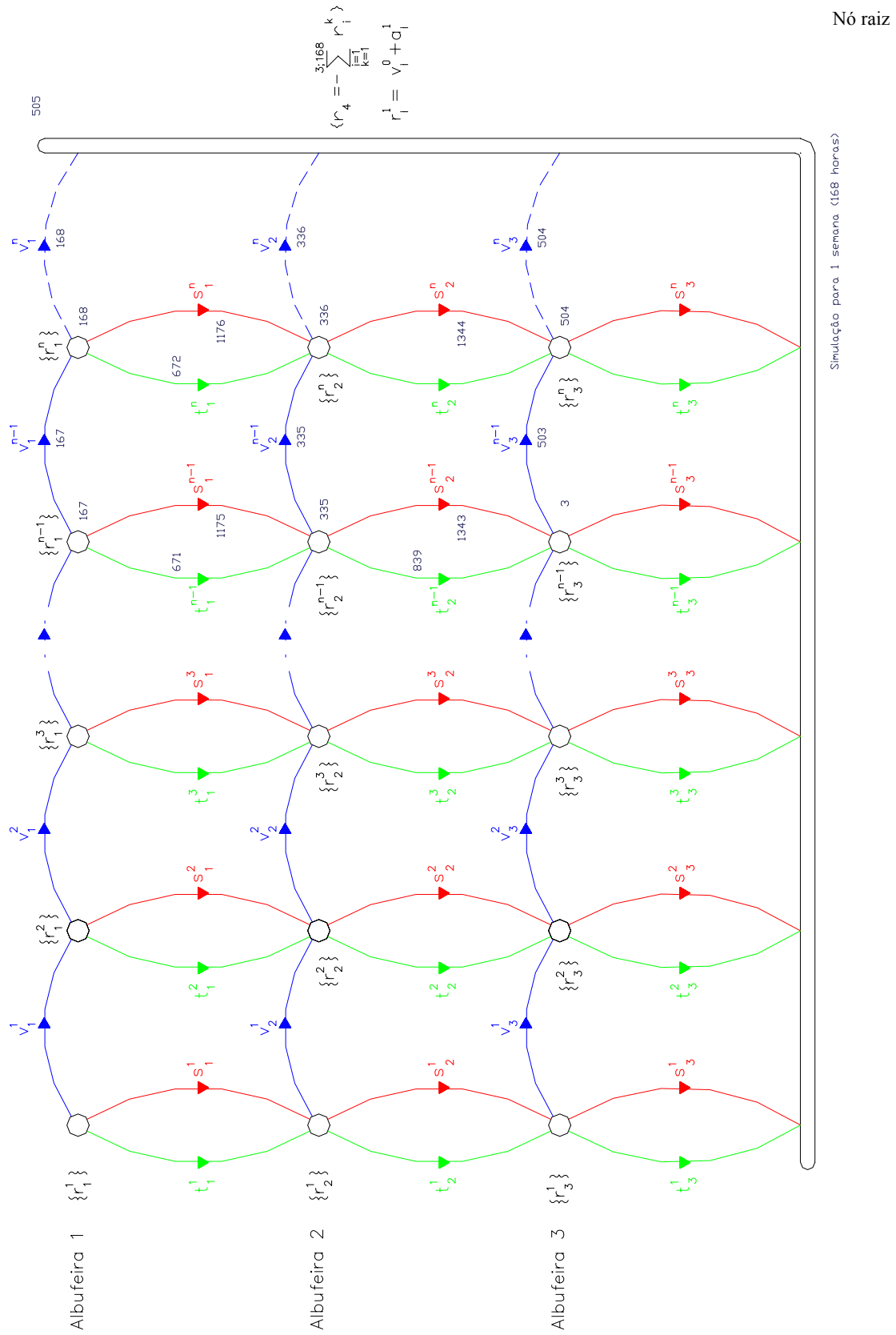


Figura 6.3 – Rede para o problema com três albufeiras em cascata.

Com a possível excepção da central de cabeça cada central recebe a água que provém da albufeira de montante e envia a água para jusante. Caso se justifique a água pode ser descarregada directamente para jusante, i.e., sem originar produção de energia eléctrica na central.

6.4- Função objectivo

A função objectivo que mede o mérito das decisões tomadas é a soma dos lucros obtidos na exploração das centrais hidroeléctricas, sendo a expressão do lucro para uma central i dada por

$$\text{lucro}_i^k = \lambda_i^k p_i^k(h_i^k, t_i^k) \quad (6.1)$$

em que λ_i^k o preço unitário da energia eléctrica para a albufeira i no período k . No que respeita à produção de energia eléctrica $p_i^k(h_i^k, t_i^k)$, esta depende não só do caudal de água turbinada como também da altura de queda útil existente entre as duas albufeiras de montante e de jusante. A figura seguinte representa algumas curvas que são aproximações lineares das curvas típicas para $p(h, t)$.

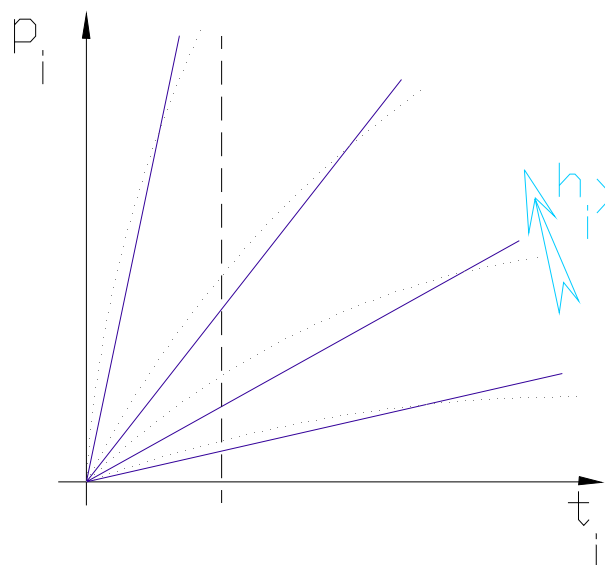


Figura 6.3 – Aproximação linear das curvas de $p(h, t)$.

Assim, resulta que são consideradas as expressões do tipo seguinte para a determinação da produção

$$p_i^k = t_i^k (\eta_{i0} + \eta_i h_i^k) \quad \begin{array}{l} i = 1, 2, 3 \\ \eta_{i0}, \eta_i \in \mathbb{R} \end{array} \quad (6.2)$$

Considerando todos os aproveitamentos hídricos, ou sejam todas as três albufeiras, obtemos a seguinte função objectivo

$$\sum_{k=1}^n (\lambda_1^k p_1^k + \lambda_2^k p_2^k + \lambda_3^k p_3^k) \quad (6.3)$$

caso o valor unitário da energia eléctrica nas centrais produtoras seja o mesmo a função objectivo é a seguinte

$$\sum_{k=1}^n \lambda^k (p_1^k + p_2^k + p_3^k)$$

Na equação (6.2), a produção p será relacionada com o caudal de água turbinada t_i^k e a altura de queda h_i^k , pela seguinte relação

$$p_i^k = t_i^k (\eta_{i0} + \eta_i h_i^k)$$

onde o valor de h_i^k depende dos níveis relativos das albufeiras em relação ao mar. Estes níveis serão calculados em função dos volumes de água armazenados nas albufeiras, sendo consideradas respectivamente as seguintes expressões, para a altura de queda e nível da água nos reservatórios

$$h_i^k = l_i^k - l_{i+1}^k \quad \text{sendo} \quad l_i^k = l_{i0} + m_i v_i^k \quad \text{e} \quad l_{i+1}^k = l_{i+1} + m_{i+1} v_{i+1}^k \quad (6.4)$$

com $l_{i0}, m_{i0} \in \mathbb{R}$

A figura seguinte ilustra a convenção utilizada para os índices e as variáveis que determinam a produção das centrais hidroeléctricas.

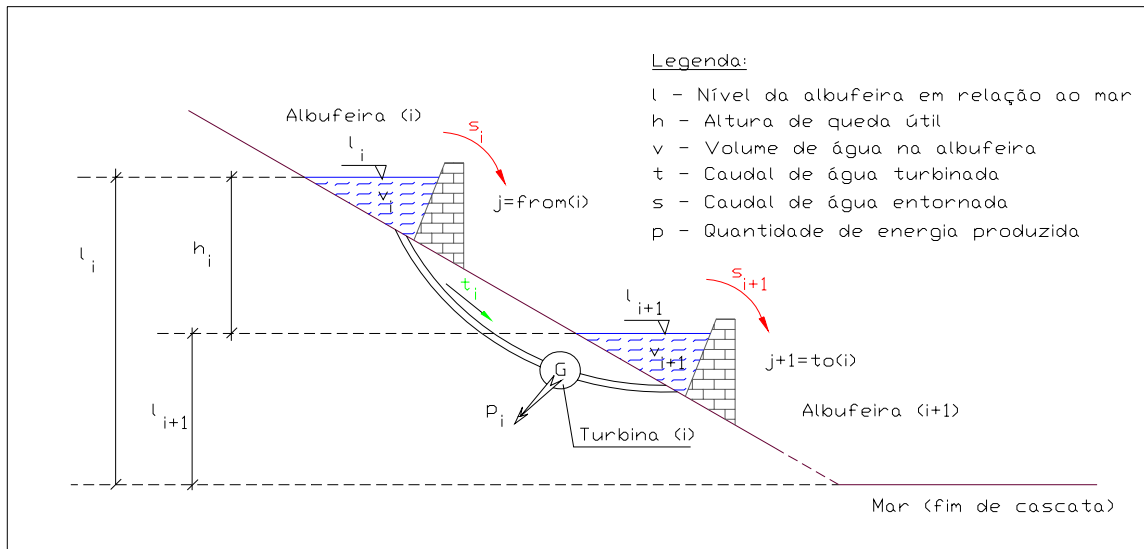


Figura 6.4 – Representação da altura de queda útil para uma albufeira.

6.5- Equações de balanço

Outras restrições do problema são as equações de balanço que impõe a condição de os nós não terem capacidade de armazenamento. A soma de todos os fluxos dos arcos sobre uma superfície fechada que encerra um nó tem que ser igual a zero ou seja

$$v_i^{n-1} + r_i^n + t_{i-1}^n + s_{i-1}^n = v_i^n + t_i^n + s_i^n \quad (6.5)$$

6.6- Construção dos ficheiros para a aplicação LSNNO

Como aplicação informática para suporte da solução deste problema decisional de gestão de curto prazo de um aproveitamento hidroeléctrico em cascata composto por três albufeiras, sugere-se o Software designado por LSNNO “Large Scale Nonlinear

Network Optimization”, desenvolvido no departamento de matemática “ND de la Paix” (Bélgica) e que está disponível no “site” da Internet com o seguinte endereço:

<http://www-fp.mes.anl.gov/otc/Guide/SoftwareGuide/Blurbs/lsnno.html>

o software pode ser obtido em

<ftp://ftp.bilkent.edu.tr/pub/IEOR/Opt/Network>

LSNNO é uma rotina para resolver problemas de optimização não linear, com restrições lineares em rede, desenvolvida em Standart ANSI Fortran.

Para utilizar este código LSNNO é necessário alterar em conformidade com os dados do problema as seguintes sub-rotinas que estão dentro do pacote fornecido pelo autor:

```

exmain.f,
    elfnct.f,
        elpr.f,
            els.f,
                rhs.f,
                    xlower.f,
                        xupper.f,
                            range.f.

```

O manual do utilizador para este código é anexado no fim deste do capítulo, sendo conveniente uma primeira leitura do manual LSNNO, A Fortran Subroutine for Large Scale Nonlinear Network Optimization Problems (User’s Guide) antes da apresentação que se segue do problema

Para compreender a introdução dos dados neste código, considere-se apenas três períodos no horizonte temporal. A rede para a cascata hídrico é então a apresentada na figura seguinte.

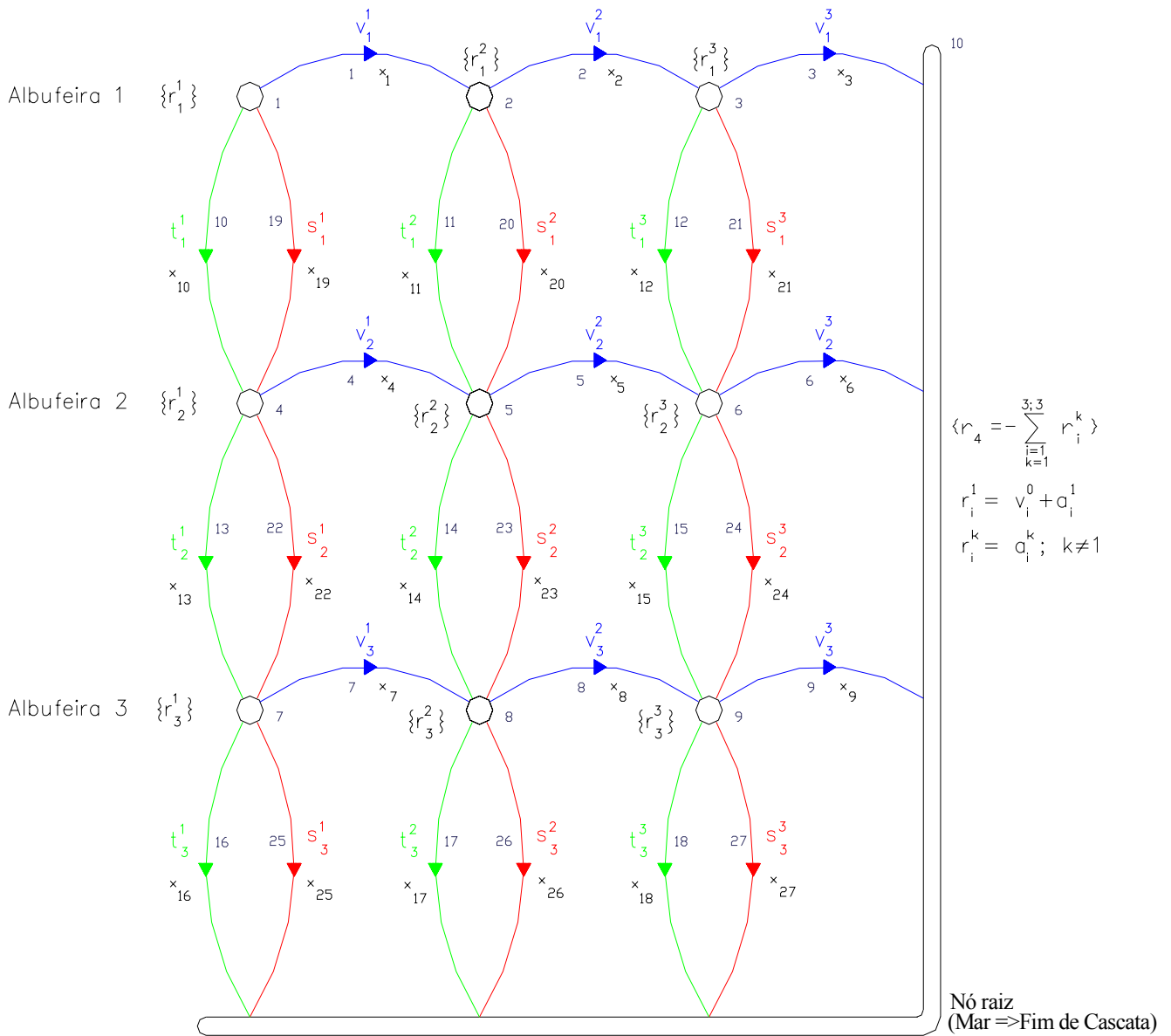


Figura 6.5 – Rede para o problema com três períodos no horizonte.

As expressões matemáticas que definem a função objectivo para as diferentes horas de funcionamento e para as três albufeiras, são obtidas usando as expressões do lucro, produção de energia eléctrica e altura de queda, sendo obtidas as equações seguintes.

	1ªhora	2ªhora	3ªhora
Albufeira 1	$\lambda^1 p_1^1$ $p_1^1 = t_1^1(\eta_{10} + \eta_1 h_1^1)$ $h_1^1 = (l_{10} + m_1 v_1^1 - l_{20} - m_2 v_2^1)$	$\lambda^2 p_1^2$ $p_1^2 = t_1^2(\eta_{10} + \eta_1 h_1^2)$ $h_1^2 = (l_{10} + m_1 v_1^2 - l_{20} - m_2 v_2^2)$	$\lambda^3 p_1^3$ $p_1^3 = t_1^3(\eta_{10} + \eta_1 h_1^3)$ $h_1^3 = (l_{10} + m_1 v_1^3 - l_{20} - m_2 v_2^3)$
Albufeira 2	$\lambda^1 p_2^1$ $p_2^1 = t_2^1(\eta_{20} + \eta_2 h_2^1)$ $h_2^1 = (l_{20} + m_2 v_2^1 - l_{30} - m_3 v_3^1)$	$\lambda^2 p_2^2$ $p_2^2 = t_2^2(\eta_{20} + \eta_2 h_2^2)$ $h_2^2 = (l_{20} + m_2 v_2^2 - l_{30} - m_3 v_3^2)$	$\lambda^3 p_2^3$ $p_2^3 = t_2^3(\eta_{20} + \eta_2 h_2^3)$ $h_2^3 = (l_{20} + m_2 v_2^3 - l_{30} - m_3 v_3^3)$
Albufeira 3	$\lambda^1 p_3^1$ $p_3^1 = t_3^1(\eta_{30} + \eta_3 h_3^1)$ $h_3^1 = (l_{30} + m_3 v_3^1 - l_{40} - m_4 v_4^1)$	$\lambda^2 p_3^2$ $p_3^2 = t_3^2(\eta_{30} + \eta_3 h_3^2)$ $h_3^2 = (l_{30} + m_3 v_3^2 - l_{40} - m_4 v_4^2)$	$\lambda^3 p_3^3$ $p_3^3 = t_3^3(\eta_{30} + \eta_3 h_3^3)$ $h_3^3 = (l_{30} + m_3 v_3^3 - l_{40} - m_4 v_4^3)$

Na última albufeira 3 o nível é constante, i.e, tem-se $m_4 = 0$. Após substituição obtemos os seguintes elementos pertencentes à função objectivo

	1ªhora	2ªhora	3ªhora
Albufeira 1			$\lambda^3 t_1^3 [\eta_{10} + \eta_1 (l_{10} + m_1 v_1^3 - l_{20} - m_2 v_2^3)]$
		$\lambda^2 t_1^2 [\eta_{10} + \eta_1 (l_{10} + m_1 v_1^2 - l_{20} - m_2 v_2^2)]$	
Albufeira 2			$\lambda^3 t_2^3 [\eta_{20} + \eta_2 (l_{20} + m_2 v_2^3 - l_{30} - m_3 v_3^3)]$
		$\lambda^2 t_2^2 [\eta_{20} + \eta_2 (l_{20} + m_2 v_2^2 - l_{30} - m_3 v_3^2)]$	
Albufeira 3			$\lambda^3 t_3^3 [\eta_{30} + \eta_3 (l_{30} + m_3 v_3^3 - l_{40})]$
		$\lambda^2 t_3^2 [\eta_{30} + \eta_3 (l_{30} + m_3 v_3^2 - l_{40})]$	

Substituindo nestas expressões os caudais de água turbinada t , descarregados s e os volumes das albufeiras v por variáveis x de acordo com a Figura 6.5, obtemos as seguintes expressões:

Quadro 6.1- Expressões para a função objectivo

	1ª hora	2ª hora	3ª hora
Albufeira 1	$\lambda^1 x_{10} [\eta_{10} + \eta_1 (l_{10} + m_1 x_1 - l_{20} - m_2 x_4)]$	$\lambda^2 x_{11} [\eta_{10} + \eta_1 (l_{10} + m_1 x_2 - l_{20} - m_2 x_5)]$	$\lambda^3 x_{12} [\eta_{10} + \eta_1 (l_{10} + m_1 x_3 - l_{20} - m_2 x_6)]$
Albufeira 2	$\lambda^1 x_{13} [\eta_{20} + \eta_2 (l_{20} + m_2 x_4 - l_{30} - m_3 x_7)]$	$\lambda^2 x_{14} [\eta_{20} + \eta_2 (l_{20} + m_2 x_5 - l_{30} - m_3 x_8)]$	$\lambda^3 x_{15} [\eta_{20} + \eta_2 (l_{20} + m_2 x_6 - l_{30} - m_3 x_9)]$
Albufeira 3	$\lambda^1 x_{16} [\eta_{30} + \eta_3 (l_{30} + m_3 x_7 - l_{40})]$	$\lambda^2 x_{17} [\eta_{30} + \eta_3 (l_{30} + m_3 x_8 - l_{40})]$	$\lambda^3 x_{18} [\eta_{30} + \eta_3 (l_{30} + m_3 x_9 - l_{40})]$

Nesta fase da leitura é conveniente ler o manual LSNNO, A Fortran Subroutine for Large Scale Nonlinear Network Optimization Problems (User's Guide), no subcapítulo 6.7, e simular o exemplo que está descrito nele, visto que, ajuda a compreender a introdução de dados no código. O função objectivo é uma forma quadrática indefinida como facilmente se prova pelo facto do traço da matriz hessiana ser nulo, pelo que este problema deveria ser tratado como uma problema que necessita dos métodos de optimização global. No entanto, pode usar-se este código se for iniciado convenientemente e averiguado os resultados obtidos. Para a introdução da função objectivo no código é necessário decompor estas em termos aditivos ditos elementos com um número reduzido de variáveis para que o código tenha um desempenho mais eficaz. Para obter as variáveis dos elementos da função objectivo a introduzir no

ficheiro elpr.f, vai se proceder à análise da distribuição dos índices das variáveis no grafo seguinte.

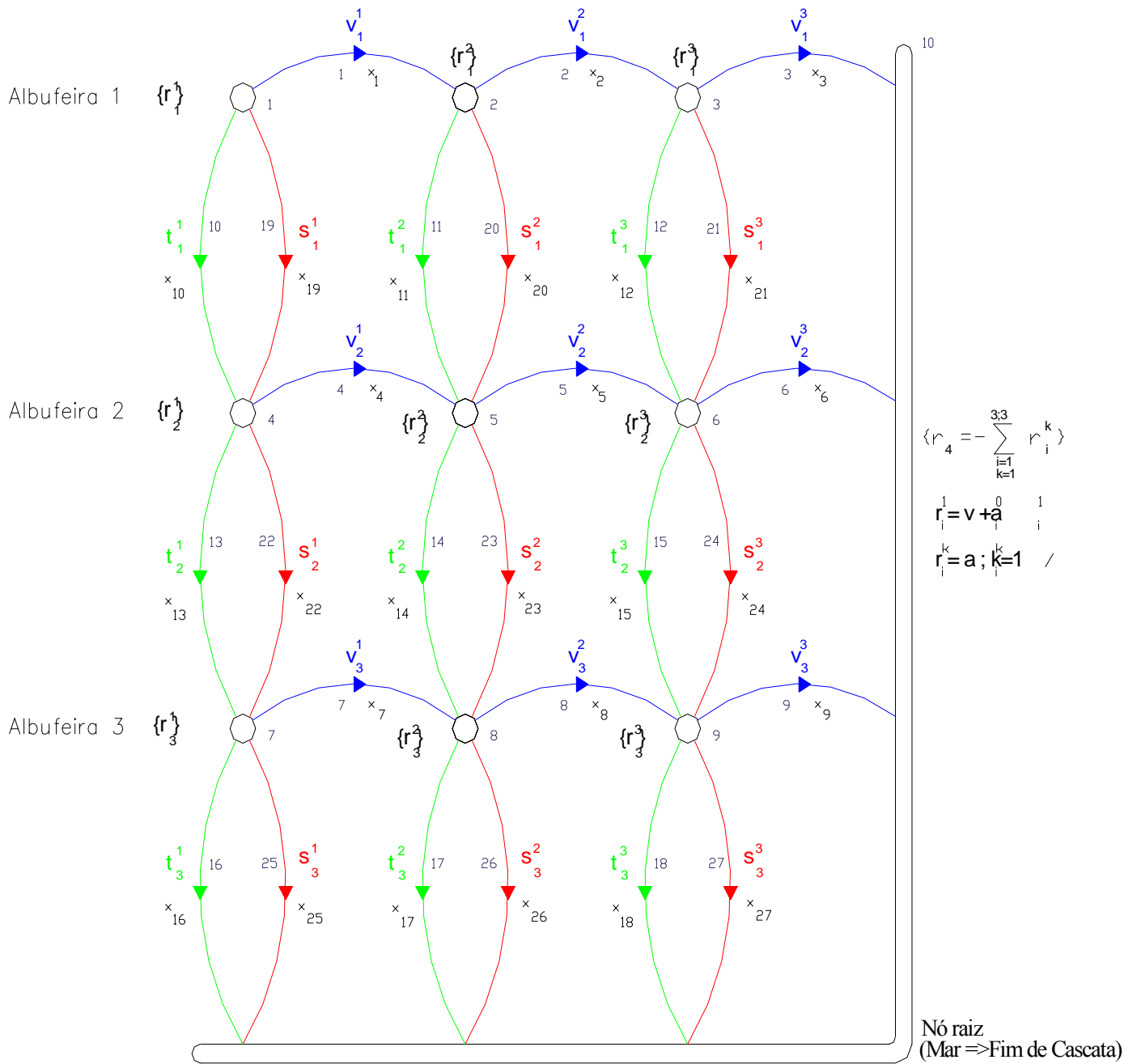


Figura 6.6 – Variáveis pertencentes a cada elemento da função objectivo.

Agrupando as variáveis de cada elemento da função objectivo obtemos o seguinte:

$$\begin{array}{lll}
 \text{elm}_1(x_1, x_4, x_{10}) & \text{elm}_2(x_2, x_5, x_{11}) & \text{elm}_3(x_3, x_6, x_{12}) \\
 \text{elm}_4(x_4, x_7, x_{13}) & \text{elm}_5(x_5, x_8, x_{14}) & \text{elm}_6(x_6, x_9, x_{15}) \\
 \text{elm}_7(x_7, x_{16}) & \text{elm}_8(x_8, x_{17}) & \text{elm}_9(x_9, x_{18})
 \end{array}$$

Em conclusão: os elementos associados às duas primeiras albufeiras, em geral os elementos associados a todas as albufeiras menos a última, são compostos por três variáveis, com a seguinte determinação dos índices:

A primeira variável tem o índice igual ao do elemento;

A segunda tem o índice dado pela soma do índice do elemento mais o número de períodos do horizonte temporal em estudo.

A última variável tem o índice igual ao do elemento somado com o produto entre o número de períodos de funcionamento e o número de albufeiras do sistema hídrico em estudo;

Para os elementos associados à última albufeira a lei a determinação dos índices é idêntica, mas são compostos por duas variáveis, com a seguinte determinação dos índices:

A primeira variável tem o índice igual ao do elemento;

A última variável tem o índice igual ao do elemento somado com o produto entre o número de períodos de funcionamento e o número de albufeiras do sistema hídrico em estudo;

Além da função objectivo é conveniente introduzir no código o gradiente desta função. Ainda, a matriz hessiana da função objectivo pode ser introduzida ou caso contrário será estimada pelo código, sendo dadas opções ao utilizador para a metodologia de estimação, que sugerimos seja testado pelo leitor o uso dessa estimativa com o fim de tira conclusões. As formulas para o cálculo do gradiente e da matriz hessiana são

seguidamente indicados a título de exemplo para o primeiro elemento e para o último do problema em estudo. A expressão do primeiro elemento da função objectivo é dada por

$$elm_1 = \lambda^1 x_{10} [\eta_{10} + \eta_1 (l_{10} + m_1 x_1 - l_{20} - m_2 x_4)]$$

Gradiente do primeiro elemento da função objectivo

$$\nabla ELM_1 = \left(\frac{\partial elm_1}{\partial x_{10}}, \frac{\partial elm_1}{\partial x_1}, \frac{\partial elm_1}{\partial x_4} \right)$$

Sendo as derivadas parciais

$$\frac{\partial elm_1}{\partial x_{10}} = \lambda^1 [\eta_{10} + \eta_1 (l_{10} + m_1 x_1 - l_{20} - m_2 x_4)]$$

$$\frac{\partial elm_1}{\partial x_1} = \lambda^1 x_{10} \eta_1 m_1$$

$$\frac{\partial elm_1}{\partial x_4} = -\lambda^1 x_{10} \eta_1 m_2$$

A matriz hessiana do primeiro elemento da função objectivo

$$H ELM_1 = \begin{bmatrix} \frac{\partial^2 elm_1}{\partial x_{10}^2} & \frac{\partial^2 elm_1}{\partial x_{10} \partial x_1} & \frac{\partial^2 elm_1}{\partial x_{10} \partial x_4} \\ \frac{\partial^2 elm_1}{\partial x_1 \partial x_{10}} & \frac{\partial^2 elm_1}{\partial x_1^2} & \frac{\partial^2 elm_1}{\partial x_1 \partial x_4} \\ \frac{\partial^2 elm_1}{\partial x_4 \partial x_{10}} & \frac{\partial^2 elm_1}{\partial x_4 \partial x_1} & \frac{\partial^2 elm_1}{\partial x_4^2} \end{bmatrix}$$

Os elementos da matriz hessiana não nulos são dados por.

$$\frac{\partial^2 elm_1}{\partial x_1 \partial x_{10}} = \lambda^1 \eta_1 m_1 \quad \frac{\partial^2 elm_1}{\partial x_4 \partial x_{10}} = -\lambda^1 \eta_1 m_2$$

Para o último elemento será:

$$elm_9 = \lambda^3 x_{18} [\eta_{30} + \eta_3 (l_{30} + m_3 x_9 - l_{40})]$$

O gradiente será

$$\nabla ELM_1 = \left(\frac{\partial elm_9}{\partial x_{18}}, \frac{\partial elm_9}{\partial x_9} \right)$$

as derivadas parciais são

$$\frac{\partial elm_9}{\partial x_{18}} = \lambda^3 [\eta_{30} + \eta_3 (l_{30} + m_3 x_9 - l_{40})]$$

$$\frac{\partial elm_9}{\partial x_9} = \lambda^3 x_{18} \eta_3 m_3$$

A matriz hessiana do último elemento é

$$H ELM_1 = \begin{bmatrix} \frac{\partial^2 elm_9}{\partial x_{18}^2} & \frac{\partial^2 elm_9}{\partial x_{18} \partial x_9} \\ \frac{\partial^2 elm_9}{\partial x_9 \partial x_{18}} & \frac{\partial^2 elm_9}{\partial x_9^2} \end{bmatrix}$$

O elemento da matriz hessiana não nulo é dado por

$$\frac{\partial^2 elm_1}{\partial x_{18} \partial x_9} = \lambda^3 \eta_3 m_3.$$

Nota sobre o código LSNNO

Da leitura do manual de utilizar do código LSNNO, verifica-se que o mesmo foi desenvolvido com vista à obtenção de soluções óptimas para problemas em que o operador de optimização é a minimização do tipo

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.a} \quad & \mathbf{Ax} = \mathbf{r} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \end{aligned}$$

logo há necessidade de recorrer da seguinte igualdade

$$\begin{array}{ccc} \max f(\mathbf{x}) & = & - \min - f(\mathbf{x}) \\ \text{sa} & & \text{sa} \\ \mathbf{x} \in F & & \mathbf{x} \in F \end{array}$$

a função objectivo não é o lucro, mas sim o simétrico do lucro, idem para o vector gradiente e para a matriz hessiana. Além disto, como o volume de água armazenado numa albufeira tem um limite inferior não necessariamente nulo as restrições do nosso problema são do tipo

$$\begin{array}{l} \mathbf{A}\mathbf{x} = \mathbf{r} \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{L} \end{array}$$

o que obriga a uma translação do vector das variáveis do tipo

$$\mathbf{x} = \mathbf{x}' + \mathbf{l}$$

para que as restrições se possam escrever como

$$\begin{array}{l} \mathbf{A}\mathbf{x}' = \mathbf{r}' \\ \mathbf{0} \leq \mathbf{x}' \leq \mathbf{u}' \end{array}$$

sendo

$$\begin{array}{l} \mathbf{r}' = \mathbf{r} - \mathbf{A}\mathbf{l} \\ \mathbf{u}' = \mathbf{L} - \mathbf{l} \end{array}$$

e a função objectivo será escrita como $f(\mathbf{x}' + \mathbf{l})$.

6.7- Dados para um caso exemplo

Na simulação computacional usaram-se os seguintes dados para as três albufeiras consideradas que são uma aproximação dos dados para as albufeiras de Miranda, Picote e Bemposta. Volumes mínimos e máximos permitidos nas albufeiras em $[10^6 \text{m}^3]$:

$$\begin{array}{l} \text{Miranda:} \\ \bar{V} = 9,2 \\ \underline{V} = 2,8 \\ \bar{V} = 11,9 \end{array}$$

Picote:

$$\underline{V} = 1,4$$

Bemposta:

$$\bar{V} = 23,3$$

$$\underline{V} = 0$$

- Caudais mínimos e máximos que é possível entornar nas albufeiras em [$10^6 m^3$]:

Miranda = Picote = Bemposta:

$$\bar{S} = 10^{14}$$

$$\underline{S} = 0 [m^3]$$

- Caudais máximos que é possível turbinar nas albufeiras em [m^3]:

Miranda: $\bar{T} = 23\ 400$

Picote: $\bar{T} = 19\ 800$

Bemposta: $\bar{T} = 25\ 200$

- Requisitos / Afluências verificadas nas albufeiras em cada hora em [m^3]:

Albufeira	1 ^a h	2 ^a h	3 ^a h	...	167 ^a h	168 ^a h
Miranda	$9,2 \times 10^6$	0	0	...	0	0
Picote	4×10^6	0	0	...	0	0
Bemposta	5×10^6	0	0	...	0	0

6.8- Manual do utilizador do LSNNO

O “Fortran Subroutine for Large Scale Nonlinear Network Optimization Problems User’s Guide” estão só disponíveis em LaTeX no ficheiro "lsnno.tex", com o objectivo de permitir uma leitura o “Users' Guide of LSNNO” é anexado nas folhas seguintes.