Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Biológicas
Departamento de Computação

Apostila de CIC373 –Fluxo em Redes

Aluno: Diego Faria Lemes

Professor: Gustavo Peixoto Silva

Programa PRÓ-ATIVA da UFOP

Junho de 2006

Sumário

1. Introdução

- Programação Linear : minimização de uma função que depende do fluxo (custo/lucro) em uma rede.
- Modelos de fluxo em redes em varias aplicações. Isso permite o desenvolvimento de algorítimos especializados com grandes vantagens computacionais.
- A geometria das redes, em relação entre as entidades, pode ser desenhada no plano, permitindo fácil compreensão do problema.

2. Alguns sistemas abordados como redes:

- Sistemas de produção/distribuição.
- Sistemas logísticos militar.
- Sistemas de trafego urbano.
- Sistemas de rodovias (transporte).
- Sistemas de comunicação.
- Rede de dutos/tubulações.
- Sistema de localização.
- Sistema de roteamento e programação.
- Redes elétricas, ...

3. Notação e Definições

Um grafo direcionado $G = (N,A)$ consiste de um conjunto N de nós e outro A de arcos.

Os arcos são pares ordenados de nós distintos.

Arcos – ligações unidirecionais de transporte de produtos.
Nós – locais de produção /consumo, terminais de conexão.

Uma rede direcionada é um grafo direcionado com valores associados aos seus nós e/ou arcos.

Arcos – custos, capacidade, fluxo.
Nós – oferta, demanda, custo/lucro marginal.

Normalmente se utiliza n para denotar o numero de nós e m para denotar o numero de arcos em G, ou seja, $|N| = n$ e $|A| = m$.

A estrutura de uma rede pode ser armazenada em uma matriz nxm que tem uma linha para cada nó, e uma coluna para cada arco, dita matriz de incidência nó-arco.

A coluna correspondente ao arco (i,j) tem apenas dois elementos não nulos, $+1$ na linha do nó i e -1 na linha do nó j.

Geralmente um Problema de Fluxo em Redes é:

Min Cx (1)
Sujeito a Ax = b (2)
$0 \leq x \leq u$, (3)

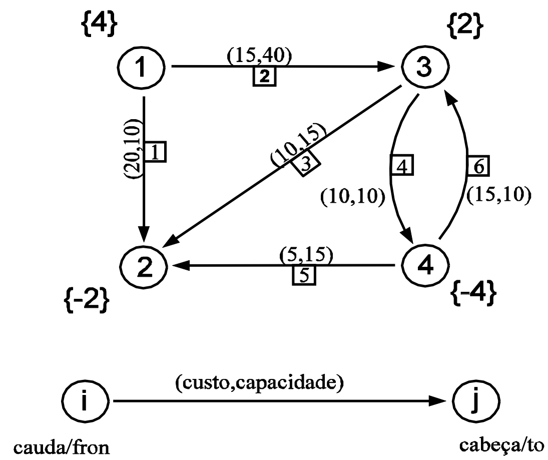
Onde:
C = vetor de custos nos arcos.
x = vetor de fluxo nos arcos (incógnita).
A = matriz de incidência nó-arco.
b = vetor de demanda nos arcos.

u = vetor de capacidade nos arcos.
A equação (1) minimizar os custos, na (2) é garantido o equilíbrio de fluxo em cada nó da rede e a restrição (3) assegura que o fluxo não ultrapasse o limite de cada arco.

Para cada nó i temos que:

Se $b_i < 0$ – i é nó produtor.
$b_i > 0$ – i é no consumidor.
$b_i = 0$ – i é nó de transbordo.

Figura 1



O custo é dado por unidade de fluxo que passa pelo arco. Assim temos o modelo de programação linear equivalente.

$$
\textbf{Matriz A} = \begin{array}{c} \\ \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{c}
\begin{array}{cccccc} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 \\ (1,2) & (1,3) & (3,2) & (3,4) & (4,2) & (4,3) \end{array} \\
\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 & 1 \end{bmatrix}
\end{array}, \quad \textbf{b} = \begin{bmatrix} 4 \\ -2 \\ 2 \\ -4 \end{bmatrix}
$$

Assim o modelo que otimiza o fluxo nesta rede (figura1) é:

$$Min \quad 20X_1 \quad +15X_2 \quad +10X_3 \quad +10X_4 \quad +5X_5 \quad +15X_6$$

Sujeito a

$$
\begin{array}{l} nó1 \\ nó2 \\ nó3 \\ nó4 \end{array}
\left\{ \begin{array}{ccccccccc}
X_1 & +X_2 & & & & & = & 4 \\
-X_1 & & -X_3 & & -X_5 & & = & -2 \\
 & -X_2 & +X_3 & +X_4 & & -X_6 & = & 2 \\
 & & & -X_4 & +X_5 & +X_6 & = & 1
\end{array} \right.
$$

$$
\begin{array}{lll}
X_1 \leq 10 & X_2 \leq 40 & X_3 \leq 15 \\
X_4 \leq 10 & X_5 \leq 15 & X_6 \leq 10
\end{array}
$$

Observação: A matriz tem apenas *2xm* elementos não nulos entre as *nxm* posições.
Esta matriz, que tem apenas *+1* e *-1* é dita uni modular, de importância teórica pois garante a existência de algorítimos mais eficientes na resolução de problemas de otimização associados a ela.

Seja O = vetor origem, D = vetor destino e C = custos nos arcos, então uma outra representação para a rede é:

arco	θ	*D*	*C*	*U*
1	1	2	20	10
2	1	3	15	40
3	3	2	10	15
4	3	4	10	10
5	4	2	5	15
6	4	3	15	10

O problema de Fluxo com custo mínimo (PFCM) pode ser formulado como:

$$Min \quad \sum_{(i,j)\in A} CijXij$$

Sujeito a

$$\underset{\text{(fluxo sai)}}{\sum_{j:(i,j)\in A} Xij} \quad - \underset{\text{(fluxo entra)}}{\sum_{i:(j,i)\in A} Xji} \quad = \underset{\text{(fluxo disponível)}}{bi} \qquad \forall\, i \in N$$

$$0 \le Xij \le Uij \qquad \forall\,(i,j) \in A$$

$$\sum_{i=1}^{n} bi = 0$$

Lei de conservação de fluxo nos nós: ∑ fluxo entra = ∑ fluxo sai.

4. Problemas Clássicos

a. **Problema do Caminho Mínimo(PCM):** Qual é a melhor forma de percorrer uma rede indo de um dado ponto a outro, com o menos custo possível?
b. **Problema de Fluxo Máximo(PFM):** É uma rede com arcos capacitados, como e qual o máximo de fluxo o que possível de se enviar de um dado ponto a outro da rede, respeitando a capacidade dos arcos?
c. **Problema de Fluxo com Custo Mínimo(PFCM):** Considerando um dado custo por unidade de fluxo em uma rede (nos seus arcos) com arcos capacitados, e que precisamos enviar unidades de fluxo alocados em determinados nós (oferta/produção) para outros nós (demanda/consumo), como faze-lo com o menor custo possível?

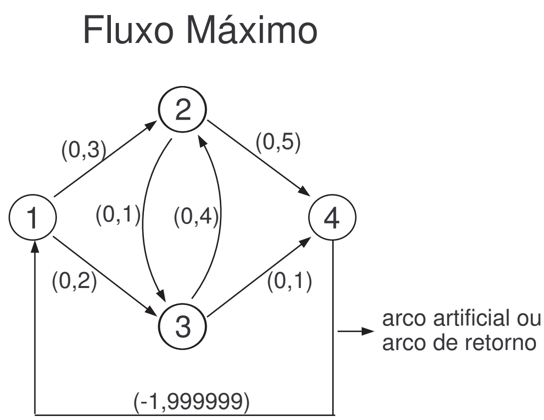
4.1 Problema do caminho mínimo

- De um nó *s* dito origem para um dado nó destino *t*. Fazemos $b_S = 1$, $b_t = -1$ e $b_i = 0 \;\; \forall\, i \neq s \text{ e } i \neq t$, temos o PCM representado como um problema de fluxo a custo mínimo.
- De um nó *S* para todos os demais nós. Fazendo $b_S = n\text{-}1$ e $b_i = 0 \;\; \forall\, i \neq s$, o problema se transforma e num PFCM.

4.2 Problema de Fluxo Máximo (PFM)

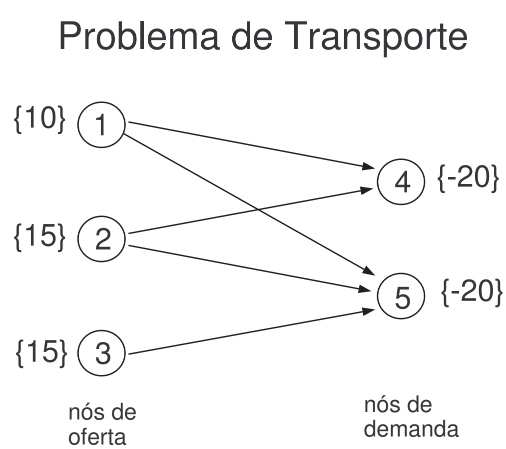
Consiste em enviar a quantidade máxima de fluxo de uma dada origem *s* para um certo destino *t* em uma rede capacitada, ou seja, $C_i < \infty \;\; \forall\, i = 1,..,m$.

Criando, ou seja, adicionando um arco de *t* para *s* com custo igual a -1 e U= ∞, e fazendo $C_i = 0$, para os demais arcos, o PFM é transformado em PFCM.



4.3 Problema de Transporte

Rede bipartida onde uma parte contém nós de oferta e a outra contém os nós de demanda. Os arcos ligam os nós de oferta diretamente aos nós de demanda.

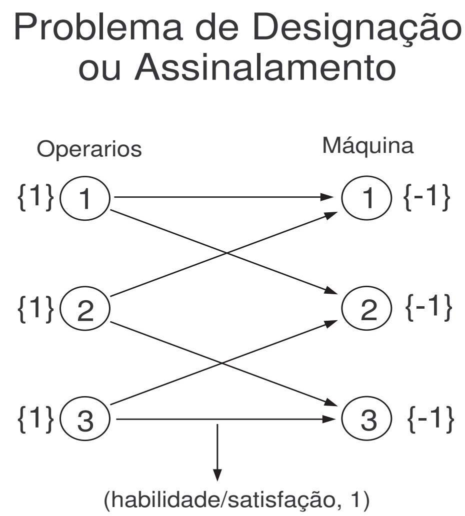


4.4 Problema de transbordo

É um problema de transporte com nós intermediários com b = 0 ou seja a demanda dos nós intermediários é igual a 0.

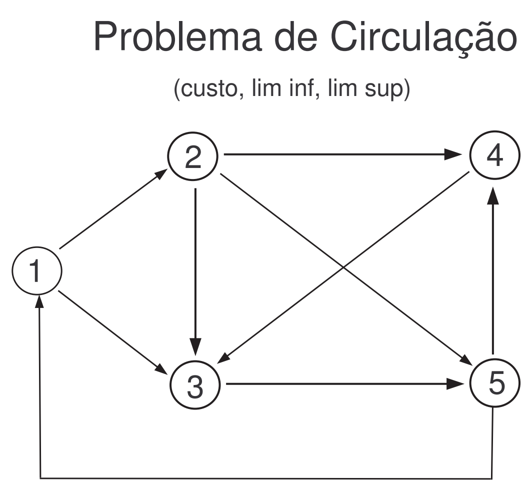
4.5 Problema de Designação ou assinalamento (casamento)

Rede bipartida onde o número de nós de oferta é igual ao número de nós de demanda. Para que haja o assinalamento 1 a 1, fazemos b_k=1 onde k é nó de oferta e b_L=-1 onde L é nó de demanda.



4.6 Problema de Circulação

Não existe oferta nem demanda nos nós. Neste caso a rede deve ter alguma característica que gere a movimentação de unidades produzindo um fluxo na rede. Assim temos b = 0, e ($C_k < 0$ ou l_M>0) para algum k ou M.



Exercício 1: Uma secretaria de educação esta colhendo propostas de 4 empresas de transporte escolar para realizar as 4 rotas pré-determinadas. Os dados são:

	*rota*1	*rota*2	*rota*3	*rota*4
*empresa*1	4000	5000		
*empresa*2		4000		4000
*empresa*3	3000		2000	
*empresa*4			4000	5000

a) Supondo que cada empresa só pode ficar com uma rota, montar a rede que minimiza o custo total da secretaria (usando assinalamento).
b) E se cada empresa puder operar em duas rotas, como fica o modelo de assinalamento?

Exercício 2: Uma companhia produz óleo em 2 poços. É possível enviar petróleo diretamente para seus clientes C1 e C2, ou enviar o petróleo a 2 portos distintos P1 e P2 e então despachar para os clientes. Os dados são:

	*poço*1	*poço*2	*porto*1	*porto*2	*cliente*1	*cliente*2
*poço*1			10	13	25	28
*poço*2			15	12	26	25
*porto*1				6	16	17
*porto*2			6		14	16
*cliente*1						15
*cliente*2					15	
oferta / demanda	150.000	200.000			160.000	140.000

A oferta e a demanda é dada em barris/dia, enquanto que os custos são para transportar 1.000 barris.

Exercício 3: Uma empresa produz e vende um único artigo. O artigo é produzido em três fabricas localizadas em cidades diferentes, com as seguintes capacidades mensais, em unidades:
Fabrica A: 20 unidades
Fabrica B: 35 unidades
Fabrica C: 40 unidades

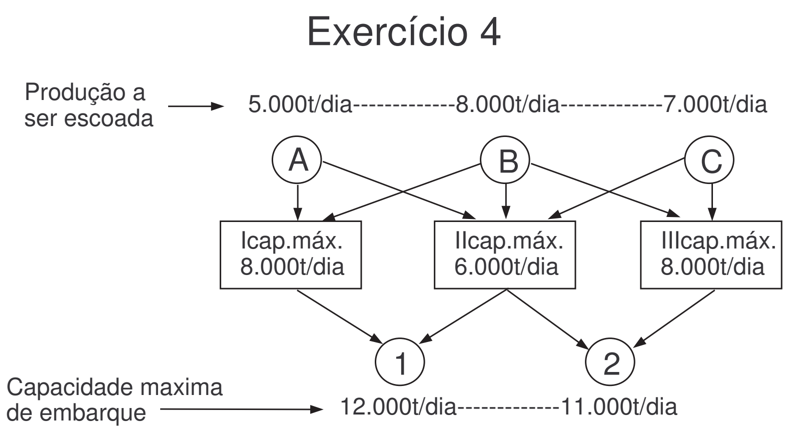
O artigo é vendido a três clientes exclusivos. O cliente F quer comprar 30 unidades por mês, o cliente G quer pelo menos 25 unidades por mês e o cliente H estão dispostos a comprar quaisquer quantidades em excesso dos respectivos mínimos estipulados.

Como os clientes são de cidades diferentes e, alem disso, os preços pagos por eles diferem entre si, o lucro por unidade auferido pela empresa depende tato da fabrica de origem como do cliente de destino. Esses lucros, em milhares de cruzados, são dados na tabela a seguir:

		clientes		
		F	*G*	*H*
	A	\$7	\$3	\$5
fábricas	*B*	\$2	\$1	\$4
	C	\$3	\$6	\$2

Suponha que a empresa queira estabelecer um plano de distribuição das produções das fabricas para os clientes que maximize o seu lucro total. Mostre como este problema pode ser formulado como o "modelo de transporte" de programação linear.

Exercício 4: Um sistema de escoamento de produção precisa levar 20.000 t/dia de três regiões para dois pontos de embarque. O material precisa ser acondicionado em ter entrepostos intermediários. Monte o problema da determinação de fluxo de transporte, a um mínimo custo global, como de programação linear. Indique claramente quais são as variáveis de decisão.



Custo Unitário de transporte \$/t

	I	II	III	1	2
A	8	5	–	–	–
B	7	6	4	–	–
C	–	5	7	–	–
I	–	–	–	6	–
II	–	–	–	7	7
III	–	–	–	–	6

4.7 Fluxo Generalizado

No PFCM existe conservação de fluxo no arco, o fluxo que entra é igual ao fluxo que sai do arco.

No problema de fluxo generalizado o arco pode consumir ou produzir fluxo. Exemplos onde isso ocorre:

(a) transmissão de energia por linhas elétricas
(b) água transportada por dutos ou canais
(c) investimentos financeiros onde os arcos representam oportunidade de investimento.

4.8 Problema de multifluxo ou múltiplos produtos

Este problema surge quando vários produtos usam a mesma rede, ou tem diferentes pontos de origem e destino. Exemplos:
(a) Transporte de passageiros de diferentes pontos
(b) O roteamento de veículos não homogêneos
(c) Transporte de grãos
(d) Envio de mensagens

5 Definições

5.1 Grafos e redes direcionados

G=(N,A) ou *R=(N,A,c,l,u,b)* onde *N* é um conjunto de nós e A é um conjunto de arcos que são pares ordenados de nós distintos.

5.2 Graus

Graus de entrada: o grau de entrada de um nó é igual ao número de arcos que chagam no nó.

Grau de saída: o grau de saída de um nó, é igual ao número de arcos que saem do nó.

5.3 Lista de adjacência

Lista de arcos adjacentes *A(i)* de um nó *i* é o conjunto de arcos que saem de *i*.
$A(i)=\{(i,j) \in A : j \in N\}$

Propriedade $\sum_{i \in N} |A(i)| = m$

5.4 Subgrafo

G'=(N',A') é um subgrafo de *G* se $N' \leq N$ e $A' \leq A$. G' é subgrafo gerador de *G* se $N'=N$ e $A' \leq A$.

5.5 Caminho

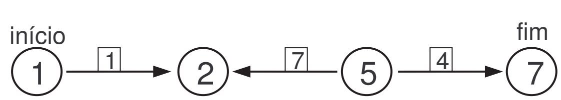
Caminho é uma seqüência alternada contínua de nós e arcos, sem repetição de nós. Pode ter arcos no sentido direto ou no sentido oposto do caminho.

5.6 Caminho direcionado ou cadeia

É um caminho com todos os arcos orientados no sentido direto.

Observação: Podemos armazenar um caminho definindo um índice (vetor) predecessor *pred(i)* para todo nó *i* do caminho. Se *i* e *j* são nós consecutivos segundo a orientação do caminho, então *pred(j)=i*. Na figura a seguir temos: *Pred(7)=5, pred(5)=2, pred(2)=1* e *pred(1)= 0*

Caminho



5.7 Ciclo
É um caminho fechado.

5.8 Circuito
É uma cadeia fechada.

5.9 Grafo Conectado
Existe um caminho conectando qualquer par de nós.

5.10 Grafo Fortemente Conectado
Existe uma cadeia (caminho direcionado) conectando qualquer par de nós.

5.11 Árvore
É um grafo conectado sem ciclos.

5.12 Árvore Geradora
É uma árvore e um subgrafo gerador.

5.13 Corte
É uma partição dos nós de N em duas partes, S e $\bar{S}$ = N – S. Cada corte define um conjunto de arcos a um extremo em S e outro em $\bar{S}$. Estes arcos são denotados por [S, $\bar{S}$]. A capacidade de um corte é dada pela soma da capacidade dos arcos que ligam nós se S a nós de $\bar{S}$ menos a soma do limite inferior dos arcos que ligam nós de $\bar{S}$ a nós de S.

5.14 Árvore Enraizada
É uma árvore com um dado nó denominado raiz.

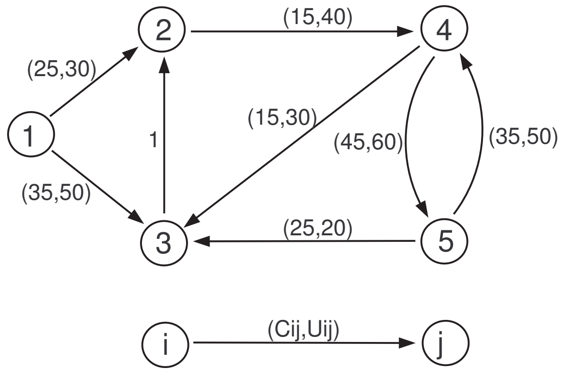
5.15 Fluxo Factível
Seja X = (X_{ij}) vetor de fluxo em cada arco da rede. X é um vetor de fluxo factível se $l_{ij} \le X_{ij} \le u_{ij} \quad \forall (i,j) \in A$ e,

Equação de conservação de fluxo
$$\begin{cases} \sum_{j \in Ai} X_{ij} - \sum_{i \in Bj} X_{ji} = v & se \ i \ é \ \ oferta, \\ \sum_{j \in Ai} X_{ij} - \sum_{i \in Bj} X_{ji} = -v & se \ i \ é \ \ demanda \\ \sum_{j \in Ai} X_{ij} - \sum_{i \in Bj} X_{ji} = 0 & se \ i \ é \ \ tranbordo \end{cases}$$

6 Armazenamento de uma rede
Lista de nós adjacentes: Para cada nó *i* temos uma lista ligada contendo os seus nós adjacentes *j* e as informações do arco *(i,j)*.

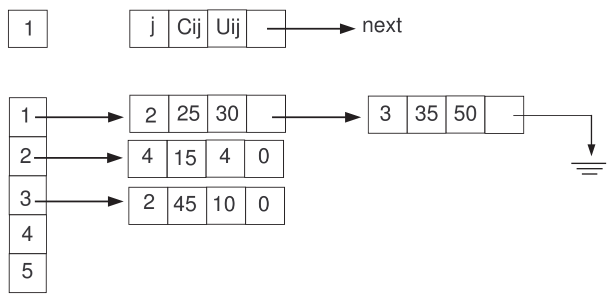
Para acessar as *n* listas temos um vetor de ponteiros para a primeira célula de cada lista. Essa primeira célula é dita "*first*" . Se a lista de adjacentes do nó i é vazia *first(i)=0* .

Armazenamento de Uma Rede



A representação por lista de adjacência da rede acima é dada, de forma resumida a seguir.

Representação forword star



7 Transformação da rede

- Transformação da rede
- Mostrar equivalência
- Inicialização método/representação padrão requerido

Considere a formulação do PFCM

$$Min \quad \sum_{(i,j)\in A} CijXij$$

Sujeito a

$$\sum_{j:(i,j)\in A} Xij \quad - \sum_{i:(j,i)\in A} Xji \quad = bi \qquad \forall\, i \in N$$

$$0 \le Xij \le Uij \qquad \forall\,(i,j) \in A$$

$$\sum_{i=1}^{n} bi = 0$$

7.1 Arcos não direcionados para arcos direcionados

Se $\{i,j\} \in A$ com $C_{ij} \geq 0$ e capacidade U_{ij}, permite fluxo de i para j e de j para i e $X_{ij} + X_{ji} \leq U_{ij}$ e na Função objetivo temos $C_{ij}X_{ij} + C_{ij}X_{ji}$.

Se $C_{ij} \geq 0$ então $X_{ij}^* = 0$ *ou* $X_{ji}^* = 0$ e considere $l_{ij} = 0$.

Transformação: trocar $\{i,j\}$ pelos arcos (i,j) e (j,i), ambos com custo C_{ij} e capacidade U_{ij}.

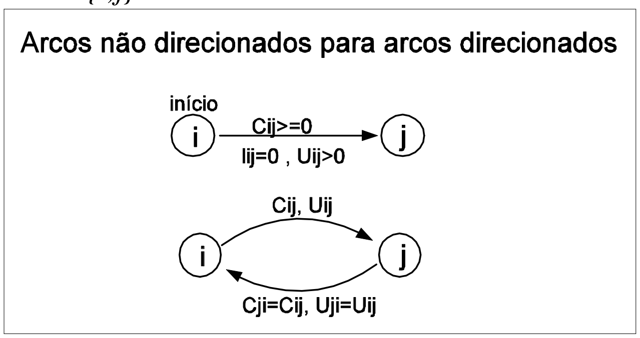
Assim, se $\{i,j\}$ tem fluxo α de i para j na rede transformada teremos $X_{ij} = \alpha$ e $X_{ji} = 0$ e vice-versa.

Analogamente, se $X_{ij} \neq 0$ e $X_{ji} \neq 0$ na rede direcionada $X_{ij} - X_{ji} > 0$ então o fluxo em $\{i,j\}$ será de $X_{ij} - X_{ji}$ unidades no sentido de i para j.

Se $X_{ji} - X_{ij} > 0$ então $\{i,j\}$ terá $X_{ji} - X_{ij}$ unidades de fluxo no sentido de j para i.

Em ambos os casos o fluxo contrario é igual a 0.

Se $X_{ij} = X_{ji}$ o fluxo $\{i,j\}$ é zero.



7.2 Removendo Limitante Inferior não Zero

Se arco (i,j) tem limitante inferior $l_{ij} > 0$ sobre X_{ij}, trocar X_{ij} por $X'_{ij} + l_{ij}$ na formulação. Então teremos $l_{ij} \leq X'_{ij} + l_{ij} \leq U_{ij}$ e depois $0 \leq X'_{ij} \leq U_{ij} - l_{ij}$.

Segundo a equação de equilíbrio de fluxo teremos: $b(i)$ passa para $b(i) - l_{ij}$ e $b(j)$ passa para $b(j) - l_{ij}$.

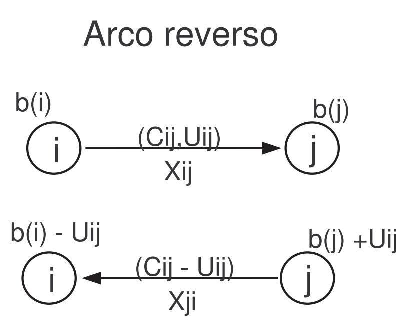
Há o acréscimo de uma constante na Função Objetivo que pode ser ignorada no processo de otimização.

7.3 Arco Reverso

Usado para eliminar arcos com custo negativo.

Seja U_{ij} capacidade de (i,j), trocar X_{ij} por $U_{ij} - X_{ij}$ então $C_{ij}\, X_{ij} = C_{ij}\, U_{ij} - C_{ij}\, U_{ji}$, trocar o arco (i,j) pelo (j,i) com custo $- C_{ij}$.

Onde X_{ji} é a quantidade de fluxo a ser "retirada" do "fluxo completo" U_{ij}.



7.4 Removendo Arcos Capacitados

Seja (i,j) com capacidade U_{ij}, então introduzir um nó adicional tal que a restrição de capacidade se transfira para a restrição de equilíbrio de fluxo.

Suponha que introduzimos uma variável de folga $S_{ij} \geq 0$, e escrevemos a restrição de capacidade na forma de igualdade.

$$X_{ij} \leq U_{ij} \text{ então } X_{ij} + S_{ij} = U_{ij}\ (-1) \text{ assim } -X_{ij} - S_{ij} = -U_{ij}.$$

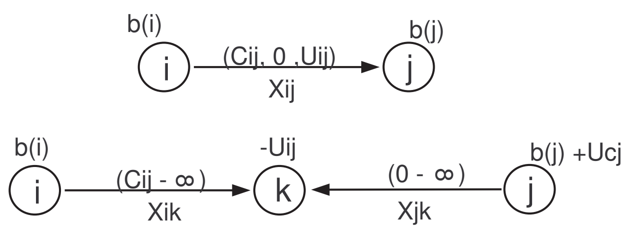
Subtraindo $-X_{ij} - S_{ij} = -U_{ij}$ da equação de fluxo do nó j temos X_{ij} e S_{ij} somente em duas equações. Tais operações corresponde a:

Exemplo:

Na rede bipartida os nós da esquerda são iguais aos nós da rede original com oferta igual $b(i) + \sum_{\{k:(k,i)\in A\}} U_{ki}$ e os nós da direita são iguais aos arcos $(i+j) \in A$ na rede original com demanda igual a $-U_{ij}$. Cada nó $(i\text{-}j)$ tem exatamente dois arcos chegando neste com origens nos nós i e j da esquerda.

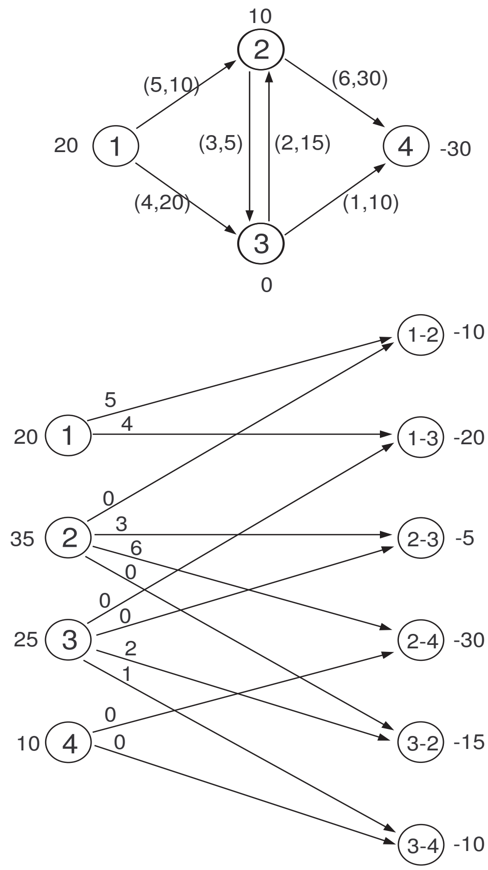
A rede transformada tem $n+m$ nós e $2m$ arcos.

Removendo arcos capacitados



Um exemplo de remoção de arcos capacitados é apresentado a seguir.

Exemplo Removendo arcos capacitados



7.5 Divisão de Nó

Divide cada nó *i* nos nós *i'* e *i"* com funções de nó de saída e entrada.

Cada arco *(i,j)* se transforma em *(i', i")* de mesmo custo e capacidade. Adiciona o arco *(i",i')* a custo zero, e a capacidade infinita, ou de acordo com a necessidade.

A oferta/demanda da rede transformada é:

Se $b(i)>0$ então $b(i")=b(i)$ e $b(i')=0$.

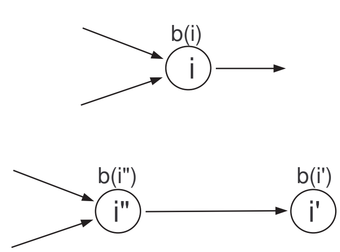
Se b(i)<0 então $b(i')=b(i)$ e $b(i")=0$.

Caso contrario $b(i')=b(i")=0$.

Exemplo

Esta transformação pode ser usada para representar situações onde os nós tem capacidade de custo para unidade que passa por ele.

Divisão de nó



8 Algoritmo de Busca

- Serve para encontrar nós de uma rede com determinadas características.
- Muitas variantes estão nos algoritmo de fluxo máximo e fluxo com caminho mínimo.

8.1 Aplicações

- Encontrar todos os nós que estão ligados a um dado nó origem através de um caminho direcionado e vice-versa.
- Identificar todos os componentes conexos de uma dada rede.
- Determinar se uma dada rede é bipartida.
- Encontrar um ciclo direcionado numa rede.

Exemplo: $G=(N,A)$, encontrar os nós eu são alcançados a partir de um nó origem S, por um caminho direcionado.

1. marcar o nó S. Os demais são nós não marcados.
2. buscar, a partir de S, os nós que satisfazem uma dada condição e marca-los.
3. suponha i marcado e j não marcado. Então, se existe o arco (i,j), este arco é dito como admissível. Caso contrario, ele é inadmissível.
4. Examinando arcos admissíveis, o algoritmo irá marcar novos nós. Se o procedimento marcar um nó j examinando um arco admissível (i,j), dizemos que o nó i é predecessor do nó j, isto é, $pred(j)=i$.
5. o algoritmo termina quando a rede não possui arcos admissíveis.

A busca percorre os nós marcados em uma dada ordem. Gravamos esta ordem num vetor $ordem(i)=k$. Neste caso, o nó i esta na k-ésima posição do caminho.

Lista é o conjunto de nós marcados ainda por examinar se existir arco admissível saindo dele.

Ao final, estão marcados todos os nós alcançáveis, a partir de S via caminho direcionado.

O vetor predecessor define uma árvore com todos os caminhos marcados.

Complexidade: $\Theta(m+n)=\Theta(m)$.

9 Caminho Mínimo

$G=(N,A)$, $c_{ij} =$ comprimento de arco (i,j). O comprimento de um caminho direcionado P é dado por:

$$\sum_{(i,j)\in P} C_{ij}$$

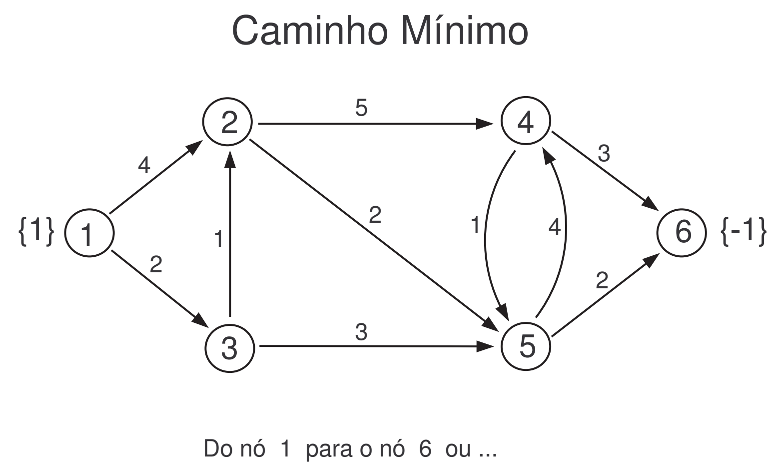
Considerando nó origem S e o nó destino A, o modelo matemático fica:

$$\min \sum_{(i,j)\in A} C_{ij} X_{ij}$$

Sujeito a

$$\sum_{(i,j)\in A} Xij - \sum_{(j.i)\in A} Xji = \begin{cases} 1 & se \quad i=s \\ -1 & se \quad i=t \\ 0 & caso \quad contrario \end{cases}$$

No caso, G não pode conter ciclos negativos.



9.1 Classificação dos Algoritmos de Caminho Mínimo

- Label setting (rotulamento), não admite custo negativo nos arcos.
- Label correcting (rotulamento modificado) admite custo negativo nos arcos.

Esses algoritmos atribui estimativas de distâncias aos nós, a cada passo. Variações estão na maneira como atualizam a distância de um passo para o outro, e como convergem para a solução.

Rotulamento: atribui uma distância como permanente (ótima) a cada iteração.

Rotulamento modificado: considera todos os rotulamentos como temporários, até o final, quando então elas se tornam permanentes.

Rotulamento: aplicáveis quando *G* é acíclicos e com custos não negativos nos arcos.

Rotulamento modificado: *G* não pode ter ciclos negativos, mas pode ter arcos com custo negativo.

9.2 Rotulamento

Algoritmo de Dijkstra: encontra o caminho mínimo a partir de uma dada origem para todos os demais nós de *G*.

Características:

- mantém $d(i) \forall i \in N$ com limite superior para a distancia de *S* até *i*.

-divide os nós: 1. rotulados permanente
2. rotulados temporariamente

-o rótulo permanente é a menos distancia de *S* até o nó.

- $d(j)=d(i) + C_{ij}$ ocorre quando o nó *j* é rótulo permanente de tem o arco *(i,j)* no caminho mínimo.
- $d(j) \leq d(j) + C_{ij} \;\; \forall (i,j) \in A$

Idéia básica:

Partir de *S* e rotular permanentemente os nós na ordem de suas distancias de *S*.

Inicialização: d(s) := 0

d(i):= $\infty \;\; \forall i \in N- \{S\}$

A cada iteração, o rótulo de um nó *i* que é a menos distancia desde a origem por um caminho cujos nós intermediários são todos rotulados permanentemente.

O algoritmo seleciona o nó com o menor rótulo temporário, o rótulo como permanente e pesquisa os arcos que saem de *i,* atualizando a distancia dos seus nós adjacentes.

O algoritmo termina quando todos os nós estiverem rotulados.

10 Padrão Dimac de armazenamento de redes

rede.txt

n_1	b_1			
n_2	b_2			
.	.			
.	.			
.	.			
n_n	b_n			
t_1	h_1	L_1	U_1	C_1
t_2	h_2	L_2	U_2	C_2
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
t_n	h_n	L_n	U_n	C_n
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
calda	cabeça	limite inferior	limite superior	custo

Para maiores detalhes, visitar o site do *Center for Discrete Mathematics and Theoretical Computer Science* em <http://dimacs.rutgers.edu/>

11 Complexidade do algoritmo de Dijkstra

- seleção dos nós: são realizadas n seleções pra encontrar o mínimo entre os nós não rotulados, com um "tempo" total de $m+(m-1)+(m-2)+...+1=\Theta(n^2)$.
- atualização da distância: são realizadas |A(i)|. No total temos $\sum_{i\in N} |A(i)| = m$.

Cada operação requer $\Theta(1)$ então temos $\Theta(m)$ para a atualização da distância .

Como $\Theta(n^2) \geq \Theta(m)$, temos que a complexidade $\Theta(n^2)$ para o algoritmo.

Observação: para rede densa m ≈ n^2 , para rede espaça m ≈ n.

Rotulamento Permanente: trabalha com redes acíclicas e com custos não negativos.
Rotulamento modificado: trabalha com redes acíclicas e com custos que podem ser negativos.

Ambos reduzem a distancia de um nó a cada iteração considerando uma busca local do tipo: o comprimento do arco e a distancia corrente dos seus nós adjacentes.

12 Condições de Otimalidade

Seja $d(j)\ j\neq s$ o comprimento do caminho mínimo da origem até j, então temos que:

$$d(j) \leq d(i) + Cij \quad \forall (i,j) \in A$$

Ou seja, para cada $(i,j) \in A$, o comprimento do caminho mínimo até j não pode ser maior do que o caminho mínimo até i mais o comprimento do arco (i,j).

12.1 Definição

A distancia reduzida do arco (i,j) em relação ao rótulo distancia d(i) é $c^{a}_{ij}=c_{ij}+d(i)-d(j)$

12.2 Propriedades

- para qualquer ciclo direcionado W, temos que
$$\sum_{(i,j)\in W} c^{d}_{ij} = \sum_{(i,j)\in W} c_{ij}$$
- Para qualquer caminho direcionado P do nó K ao nó L, temos que
$$\sum_{(i,j)\in P} c^{d}_{ij} = \sum_{(i,j)\in P} c_{ij} + d(k) - d(l)$$
- se d(i) representa a distancia de caminhos mínimos, então $c^{d}_{ij} \geq 0 \ \forall (i,j)\in A$

Exercício

Aplicar Dijkstra à rede acima escrevendo a cada iteração o conteúdo de S, $\bar{S}$, d e pred(i).

13 Caminho Mínimo de *S* para *t* pelo Dijkstra

Ao final o vetor d contém a distancia mínima de S até o referido nó.

Para encontrar o caminho mínimo de S para t, basta interromper o processo quando o nó t for escolhido para ser rotulado permanentemente, ou seja, quando

$d(t)=min\{d(j):j\in \bar{S}\}$.

14 Algoritmo de rotulamento modificado para o caminho mínimo.

Suponha que a rede não tenho ciclos negativos. No algoritmo do rotulamento modificado, a lista é examinada pelo método FIFO (first-in,first-out), ou seja, a lista é uma fila.

Teorema: o algoritmo do rotulamento modificado FIFO para o problema co caminho mínimo tem complexidade $\Theta(nm)$.

Implementação com uma dequene: é mais rápida para redes esparsas.

Dequene: lista onde podemos incluir ou retirar elementos do inicio ou do final.

Retira sempre da frente da fila. Para adicionar um nó à fila observe: se já estiver anteriormente à fila, adicionar na frente. Caso contrario, adicionar o nó no final da fila.

Representamos uma fila por um vetor do tamanha $n=1$ para armazenar até n elementos.

1	2	3	4	5	6	7	8
		5	8	2	4		

fila

↑ *prieiro* ↑ *último*

Temos um 'ponteiro' *prim*, que mantém o índice da primeira posição na fila menos um, e mantemos o outro ponteiro *ult* com o índice do ultimo elemento da fila.

14.1 Operação na Fila

- A fila estará vazia se *prim=ult*
- o primeiro elemento da fila é acessado por *fila(prim+1),* ou seja, para retirar faça *prim:=prim+1* e *I:=fila(prim)*
- para inserir um elemento *i* na lista, faça *ult:=ult+1* e *fila(ult):=i.*

Exercício

Executar o algoritmo nas redes do capítulo do livro de Ahuja, Magnanti e Orlin.

15 Fluxo Máximo

15.1 Cortes: Separando a origem e o destino

G rede direcionada. *G(N,A,l,u,s,t).* Sejam

X subconjuntos de nós de G contendo a origem, $\overline{X}$ subconjunto de nós de G contendo o destino tal que $X \cup \overline{X} = G$ e $X \cap \overline{X} = \phi$.

A partição $(X, \overline{X})$ define um corte separando os nós de origem e destino.

Os arcos diretos do corte $(X, \overline{X})$ são {(i,j):(i,j), $i \in X$ e $j \in \overline{X}$ }.

Os arcos reversos do corte $(X, \overline{X})$ são {(i,j):(i,j), $i \in \overline{X}$ e $j \in X$ }.

A capacidade do corte $(X, \overline{X})$ é dada por $\sum_{(i,j)} U_{ij} - \sum_{(i,j)} l_{ij}$

15.2 Formulação

Considere a rede capacitada *G=(N,A)* com capacidade não negativa U_{ij} para cada arco $(i,j) \in A$. A formulação matemática para o problema de fluxo máximo do nó origem *s* para o nó destino *t* fica:

Max v

s.a

$$\sum_{\{i:(i,j) \in A} X_{ij} - \sum_{\{j:(j,i) \in A} X_{ji} = \begin{matrix} v \; se \; i = s \\ 0 \;\; \forall \, i \in N - \{s,t\} \\ -v \; se \; t = t \end{matrix}$$

$0 \leq X_{ij} \leq U_{ij} \;\; \forall \, (i, j) \in A\,]$

Consideramos o problema de fluxo máximo nas seguintes condições:

1.a rede é direcionada
2.todas as capacidades são inteiras não negativas
3.a rede contém um caminho direcionado de *s* para *t* com capacidade infinita
4.a rede não contém arcos paralelos.

15.3 Caminho de Aumento de Fluxo (CAP)

Seja $X=(X_{ij})$ um vetor de fluxo factível de valor *v* numa rede *G=(N,A,l,u,s,t).* Um caminho *P* de *s* para *t* é dito um caminho de aumento de fluxo em relação a *X* se:

$X_{ij}<U_{ij}$ para os arcos diretos de *P*

$X_{ij}>l_{ij}$ ara os arcos reversos de *P*

Seja $\delta = min\{\delta_1, \; \delta_2\}$ onde

δ_1*=min{(U_{ij}-X_{ij}):(i,j) é um arco direto em P}*
δ_2*=min{(X_{ij}-l_{ij}):(i,j) é um arco reverso em P}*

Define-se um novo fluxo a partir de $\overline{X}$ dado por:

$\overline{X}_{ij} = X_{ij}$ *se (i,j)*$\notin P$

$\overline{X}_{ij} = X_{ij} + \delta$ *se (i,j) é arco direto em P*

$\overline{X}_{ij} = X_{ij} - \delta$ *se (i,j) é arco reverso em P.*

Assim $\overline{X}$ é um fluxo factível e dado um fluxo factível e um *CAF*, é possível construir um outro fluxo factível com valor maior que o anterior.

15.4 Problema do fluxo factível (aplicação do fluxo máximo)

Identificar um fluxo X na rede G=9N,A) tal que:

$$\sum_{\{j:(i,j)\in A\}} X_{ij} - \sum_{\{j:(j,i)\in A\}} X_{ji} = b_{ij} \quad \forall\, i \in N$$

$$0 \le X_{ij} \le U_{ij} \quad \forall\ (i,j) \in A \qquad (1)$$

$$\sum_{I\in N} b_i = 0$$

Encontramos um fluxo factível resolvendo um problema de fluxo máximo numa rede aumentada como segue.
Introduzimos dois nós s-super origem e t-super destino. Para cada i com bi>0 acrescentamos um arco (s,i) com capacidade bi e para cada nó i com bi<0, acrescentamos um arco (i,t) com capacidade –bi.
Esta é a rede transformada, sobre a qual calculamos o fluxo máximo de s pata t.
Se o fluxo máximo saturar todos os arcs ligados a s e a t, entãoo problema (1) tem um fluxo factível caso contrario ele será infactível.
Caso exista, o fluxo factível será aquele definido de fluxo máximo na rede original da rede transformada.

15.5 Teorema do fluxo máximo/corte mínimo

O valor do fluxo máximo é igual a capacidade mínima dentre os cortes que separam a origem e o destino.

15.6 Problema do fluxo máximo

Dados dois nós s origem e destino t numa rede capacitada, enviar o máximo de fluxo de s pata t.
Tipos básicos de algoritmo:

1. algoritmos de caminhos aumentantes que mantêm o equilíbrio de fluxo nos nós com exceção de s para t.
2. Algoritmos Preflow-push inundam a rede onde os nós podem ter excesso de fluxo pré fluxo. Os excessos são empurrados para t ou de volta para s.

16 Algoritmo genérico de caminhos aumentantes

Algoritmo caminho aumentante
Begin
  X:=0;
  While G(x) contém um caminho direcionado de s para t faça

Begin

Identifique um caminho aumentante P de s pata t;

Δ=min{rij:(i,j)∈ P}

Aumente Δ unidades de fluxo de s para t em P

Atualize G(x),

End;

End;

17 Aplicações do problema de fluxo máximo

17.1 Problema dos representantes

Uma cidade tem *r* residentes *R1, R2, …,Rr; q* clubes *C1, C2, …,Cq*; e *p* partidos políticos *P1, P2,…, Pp.* Cada residente é membro pelo menos de um clube e pode pertencer a exatamente um partido político. Cada clube deve nomear um de seus membros para representá-lo na comissão que irá governar a cidade de modo que o número dos membros da comissão que pertencem ao partido político *Pk* seja de no máximo *uk*. É possível encontrar uma comissão que satisfaça esta restrição de equilíbrio?

Ilustramos esta formulação com um exemplo.

Consideramos um problema com *r=7, q=4, p=3,* e formulamo-lo como um problema de fluxo máximo. *Figura 1*. Os nós *R1, R2...,R7* representam os residentes, os nós *C1, C2... C4* representam os clubes, e os nós *P1, P2...,P3,* representam os partidos políticos.

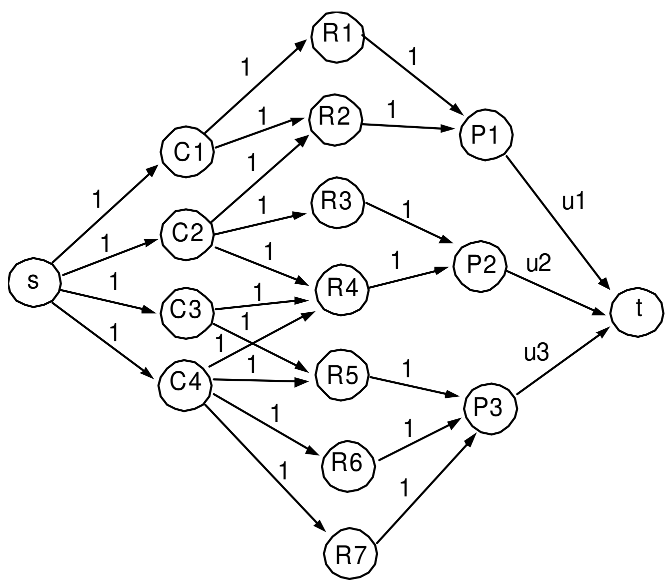


Figura 1

A rede contém também um nó *s* de origem e o nó *t* de destino. Existe um arco *(s,Ci)* para cada nó *Ci* que denota um clube, um arco *(Ci, Pk)* se o residente *Rj* pertencer ao partido político *Pk.* Finalmente, adicionamos um arco (*Pk, t)* para cada *k=1....3* com a capacidade *Uk;* todos arcos restantes têm a capacidade igual a *1*

Encontramos em seguida um fluxo máximo nesta rede. Se o valor máximo do fluxo igualar q, a cidade tem uma comissão equilibrada; caso contrário, não. A prova desta afirmação é fácil de estabelecer mostrando que:

(1) o fluxo de valor q na rede corresponde a uma comissão equilibrada, e que

(2) a comissão equilibrada implica um fluxo de valor q na rede.

Este tipo de modelo tem aplicações em diversos problemas de distribuição de recurso. Para o exemplo, suponha que os residentes são pessoas especialistas, o clube *Ci* é o conjunto de pessoas com uma habilidade particular, e o partido político *Pk* corresponde a uma experiência de cada pessoa.

Neste exemplo, um conselho equilibrado da cidade corresponde a uma atribuição de pessoas à comissão de diretores para governar a cidade, de modo que cada classe de especialidade tenha representantes na comissão e que nenhum tipo de experiência tenha um número dominante de representantes.

17.2 Programação de Maquinas Paralelas Uniformes

Nesta aplicação consideramos o problema de escalonar um conjunto de J job's em M máquinas uniformes paralelas. Cada job $j \in J$ tem um processamento requerido pj (significando a quantidade de maquina-dia requerido para completar o job), uma data de liberação rj (representando a data em que o job j se tornou disponível para processamento) é uma data de término do job $dj = rj + pj$ (representando a data em que o job deve ser finalizado). Assume-se que uma máquina pode trabalhar somente em um job por vez e que cada job pode ser processado por mais de uma máquina por vez. Todavia, é permitida a preempção, ou seja, pode-se interromper um job e processá-lo em diferentes máquinas e em dias distintos. O objetivo do problema é determinar um escalonamento viável que complete todos os jobs antes da data limite para o seus respectivos términos, ou mostrar que tal escalonamento não existe.

Este tipo de problema de programação aparece em sistemas produtivos em massa, ou seja, em lotes com grandes números de unidades. O problema de escalonamento da produção, descrito no parágrafo acima, é um problema fundamental neste contexto e pode ser usado como uma sub-rotina para uns problemas de programação mais gerais, tais como o problema do tempo de entrega mínimo, e o problema da máxima da utilização.

O problema de escalonamento foi formulado como um problema de fluxo máximo, conforme ilustra a Tabela 1. No exemplo da Figura 2 tem-se que $M = 3$ máquinas. Em primeiro lugar, todas as datas de liberação rj e as datas finais dj são ordenadas ascendentemente e determina-se $P \leq 2|J|-1$ intervalos mutuamente disjuntos de datas entre instantes consecutivos. Seja $Tk;l$ um intervalo que inicia na data k e termina no início do dia $l + 1$. No exemplo em questão, a ordem das datas de liberação e finais é a seguinte: 1; 3; 4; 5; 7; 9. Existem 5 intervalos representados por $T_{1,2}$, $T_{3,3}$, $T_{4,4}$, $T_{5,6}$ e $T_{7,8}$. Note que dentro de cada intervalo $Tk_{,l}$, o conjunto de jobs disponíveis não se altera. Podem-se processar todos os jobs j com $rj \leq k$ e $dj \leq l+1$ no intervalo.

O problema de escalonamento foi formulado como um problema de fluxo máximo em uma rede bipartida G. Foi introduzido um nó origem s, um nó destino t, um nó correspondente a cada job j e um nó correspondente a cada intervalo $Tk,1$ conforme pode ser visualizado na Figura 2. Conecta-se o nó origem com todos os nós job's j com um arco de capacidade pj, indicando que são necessários pj dias de tempo de máquina para o job j. Conecta-se cada intervalo de nó $Tk,1$ ao nó destino t com um arco de capacidade $(l - k + 1)M$, representando o número total de máquinas-dia disponíveis no intervalo de dias de k a l. Finalmente, conecta-se um nó job j a todos intervalos de nos

Tk,l se $rj \leq k$ e $dj \leq l+1$ por um arco de capacidade (l-k+1) que representa o número máximo de máquinas que podem ser destinadas ao job *j* no intervalo de dias de *k* a *l*. O problema de fluxo máximo nesta rede foi resolvido. O problema possui um escalonamento viável se, somente se, o valor do fluxo máximo for igua $\sum_{j \in J} p_i$.

$Job(j)$	1	2	3	4
$\Pr ocessamento(pj)$	1,5	1,25	2,1	3,6
$Liberação(rj)$	3	1	3	5
$Entrega$	5	4	7	9

Tabela 1

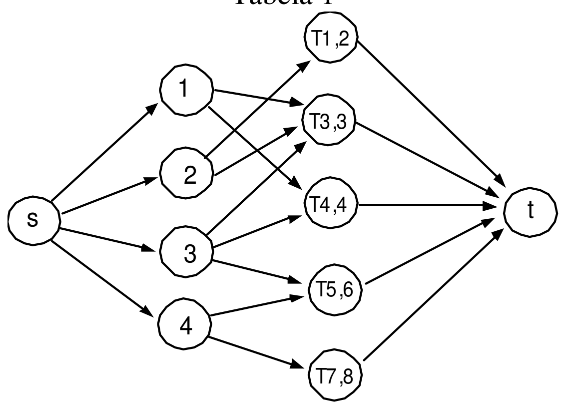


Figura 2

17.3 Arredondamento de matriz

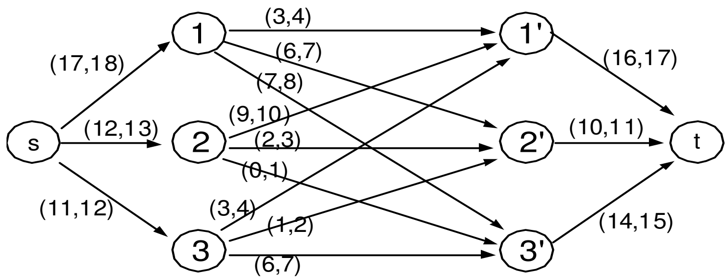
Dada uma matriz pxq de reais D={dij} cuja soma das linhas é αi e a soma das colunas é βj.

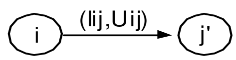
Podemos arredondar qualquer número a para $\lfloor a \rfloor$ ou $\lceil a \rceil$.

O problema de arredondamento requer que a soma dos arredondantes seja igual ao arredondamento das somas.

Pode-se encontrar um arredondante consistente resolvendo o problema do fluxo máximo para a rede correspondente.

$$\begin{array}{cccc} \left[\begin{array}{ccc} 3,1 & 6,8 & 7,3 \\ 9,6 & 2,4 & 0,7 \\ 3,6 & 1,2 & 6,5 \end{array}\right] & \begin{array}{c} 17,2 \\ 12,7 \\ 11,3 \end{array} \\ \begin{array}{ccc} 16,3 & 10,4 & 14,5 \end{array} & \end{array}$$





18 Problema de Fluxo com Custo Mínimo

18.1 Algoritmos Básicos

- Algoritmo de cancelamento de fluxo negativo
  Usa caminho mínimo para encontrar ciclos aumentantes com custo negativo (ciclo negativo), então aumenta o fluxo neste ciclo. Ele repete a operação até que nenhum ciclo negativo exista na rede residual

- Algoritmo de sucessivos caminhos mínimos
  Introduz gradualmente fluxo na rede de algum nó origem para algum nó destino, cada vez selecionando um caminho mínimo devidamente construído.
- Primal-dual e out-of-kilter
  Usam estratégias similares: a cada iteração resolve um problema de caminho mínimo e aumentam o fluxo em um ou mais caminhos mínimos.

18.1.1 Primal-Dual

Usam o algoritmo do fluxo máximo para aumentar o fluxo simultaneamente em vários caminhos mínimos.

18.1.2 Out-of-Kilter

Permite que o fluxo viole a capacidade nos arcos. Usa o caminho mínimo ou de aumento de fluxo para encontrar fluxos que satisfazem a capacidade e as condições de otimalidades.

18.1.3 Simplex para rede

Utiliza como idéias central árvore geradora mínima que esta associada a uma base do sistema linear. As arvores geradoras mínimas são soluções obtidas atribuindo aos arcos fora da árvore um fluxo igual a zero ou igual à capacidade do arco.

O problema de fluxo com custo mínimo tem pelo menos uma solução arvore geradora mínima ótima. A cada iteração um arco fora da árvore entra nela e um arco fora dela é retirado.

Hipóteses

- Todos os dados são inteiros
- A rede é direcionada
- $\sum_{i \in N} b_i = 0$
- Todos os custos são não-negativos

- Rede residual segundo um fluxo *X, G(X).*

18.2 Condições de Otimalidade do PFCM

18.2.1 Ciclo Negativo

Teorema 1: Uma solução factível X^* é ótima para o problema de fluxo com custo mínimo se e somente se a rede residual $G(X^*)$ não contém ciclo (direcionado) negativo.

18.2.2 Custo reduzido

$C_{ij}^{d} = -C_{ij} + d_{(i)} - d_{(j)} \geq 0 \quad \forall (i,j) \in A$

Definições: $\pi_{(i)} \in R \quad \forall i \in N$ é dito potencial do nó i.

$\pi_{(I)}$ é a variável dual correspondente à restrição de equilíbrio no nó i

Seja

$C_{ij}^{\pi} = C_{ij} - \pi_{(i)} + \pi_{(j)} \quad \forall (i,j) \in A$ o custo reduzido do arco (i,j). Observe que $\pi_{(I)}$ = -*d(i).*

Propriedades:

a) Para qualquer caminho direcionado P de k a l temos $\sum_{(i,j)\in P} C_{ij}^{\pi} = \sum_{(i,j)\in P} C_{ij} - \pi(k) + \pi(l)$

b) Para qualquer ciclo direcionado W $\sum_{(i,j)\in W} C_{ij}^{\pi} = \sum_{(i,j)\in W} C_{ij}$

a) Potenciais não alteram o caminho mínimo entre k e l, pois aumentam em $\pi_{(l)} - \pi_{(k)}$.

b) **Ciclo negativo para C também o é para C^{π}.**
Teorema 2:**Uma solução factível X* é ótima para o problema de fluxo com custo mínimo se e somente se algum conjunto de potenciais π satisfaz:**

$C_{ij}^{\pi} \geq 0 \forall (i,j) \in G(x^*)$

$C_{ij}^{\pi} = C_{ij} - \pi_{(i)} + \pi_{(j)} \geq 0 \quad \forall (i,j) \in A$

Podemos também definir 'potencial ótimo' como o conjunto que satisfaz a condição $C_{ij}^{\pi} \geq 0 \forall (i,j) \in G(x)$ para algum fluxo factível *X.*

18.3 Interpretação Econômica

Cij= custo de transportar uma unidade do produto de *i* para *j* e

μ(i) = -π(i) custo para produzir uma unidade do produto em *i*

Cij+ μ(i) é o custo do produto obtido em *i* e transportado até *j.*

$C_{ij} - \pi_{(i)} + \pi_{(j)} \geq 0$ ou $\mu_{(j)} \leq C_{ij} + \mu_{(i)}$

Significa que no ótimo, o custo para obter uma unidade em *j* não pode ser maior que obtê-la em *i* e transporta-la até *j* pelo arco *(i,j).*

18.4 Folgas Complementares

Os dois teoremas anteriores impõem condições sobre a rede residual. Estas condições são agora colocadas em termos da rede residual.

Teorema 3 – Uma solução factível X^* é ótima para o problema de fluxo com custo mínimo, se e somente se, para algum conjunto de potenciais π. O fluxo e o custo reduzido satisfazem as condições:

Para $\forall (i,j) \in A,$

Se $C_{ij}^{\pi} > 0,$ então $X_{ij}^{*} = l_{ij}$

Se $C_{ij}^{\pi} = 0$, então $l_{ij} < X_{ij}^{*} < U_{ij}$

Se $C_{ij}^{\pi} < 0$, então $X_{ij}^{*} = U_{ij}$.

18.5 Relacionando Fluxo Ótimo com Potencial Ótimo

18.5.1 Calculando o potencial ótimo conhecido o fluxo ótimo (dado $X^{*} \Rightarrow$ obter π^{*})

Seja *G(X*)* a rede residual em relação ao fluxo ótimo X^*. *G(X*)* não contém ciclos negativos pois X^* é ótima. Seja *d(i)* o vetor de distancias mínimas a partir do nó 1 para todos nós na rede residual, considerando *Cij* o tamanho do arco *(i,j)*.

A condição de otimalidade para o caminho mínimo implica que:

$d(j) \le C_{ij} + d(i) \quad \forall (i,j) \in G(X^{*})$ fazendo π = *-d* temos que:

$C_{ij}^{\pi} = C_{ij} - \pi(i) + \pi(j) \ge 0 \quad \forall (i,j) \in G(X^{*})$.

18.5.2 Obtendo o fluxo ótimo, conhecido o potencial ótimo(dado $\pi^{*} \Rightarrow$ obter X^{*})

Calcular o custo reduzido C_{ij}^{π} $\forall (i,j) \in A$

Se $C_{ij}^{\pi} > 0 \Rightarrow X_{ij}^{*} := 0$ e deletar (i, j) de A

Se $C_{ij}^{\pi} < 0 \Rightarrow X_{ij}^{*} := U_{ij}$, deletar (i, j) de A e $b(i) := b(i) - U_{ij}$ e $b(j) := b(j) + U_{ij}$.

Se $C_{ij}^{\pi} = 0 \Rightarrow 0 < X_{ij}^{*} < U_{ij}$ (Que valor atribuir a *X** neste caso?)

Seja *G'(N,A') e b'* o vetor oferta/demanda modificado. Agora o problema se resume em encontrar um fluxo factível e G' que atende à oferta/demanda modificada b' nos nós. Usar a aplicação do problema de fluxo máximo para encontrar um fluxo factível.

18.6 Solução básica factível para o problema de fluxo com custo mínimo

Qualquer solução básica factível contém três tipos de variáveis.

- Variáveis Básicas sem degeneração, cada variável básica X_{ij} satisfaz $l_{ij} < X_{ij} < U_{ij}$. Com degeneração, é possível que uma variável básica assuma um valor dos limitantes do arco.
- Variáveis não básicas X_{ij}: Assumem o valor do limite superior do arco U_{ij}.
- Variáveis não básicas X_{ij}: Assume o valor o limite inferior do arco l_{ij}.

18.7 Resolvendo uma rede com *n* nós.

Vamos considerar as *n* restrições de conservação de fluxo e ignorar as restrições de capacidade.

Qualquer solução satisfazendo *n-1* restrições automaticamente satisfaz a ultima restrição.

Uma solução básica factível para uma rede com *n* nós terá *n-1* variáveis básicas.

Como determinar uma solução básica factível com *n-1* variáveis?

Um conjunto A de *n-1* variáveis será uma solução básica factível se e somente se seus arcos formam uma árvore geradora para a rede.

18.8 Computando os coeficientes da F.O. para uma solução básica factível.

Para uma dada solução básica factível, seja $C_{ij}.B^{-1} = [\pi_1, ..., \pi_n]$ os potenciais dos nós.

Se definirmos $\pi_1 = 0$, então os coeficientes de X_{ij} na linha da função objetivo podem ser escritos como $C_{ij}^{\pi} = C_{ij} - \pi_{(i)} + \pi_{(j)}$.

Como cada variável básica deve ter $C_{ij}^{\pi} = 0$, podemos encontrar $\pi_1, \ldots, \pi_n$ resolvendo o seguinte sistema linear:

$$\begin{cases} \pi_{(1)} = 0 \\ C_{ij} = \pi_{(i)} - \pi_{(j)} \\ \pi_{(i)} = C_{ij} + \pi_{(j)} \end{cases}$$

18.9 Determinando de uma solução básica factível é ótima.

Basta que todas as variáveis não básicas satisfaçam as condições de otimalidade.

Uma solução básica factível é ótima se e somente se as seguintes condições são satisfeitas:

1. Se uma variável não básica $X_{ij}=l_{ij}$ se e somente se $C_{ij}^{\pi} \geq 0$.
2. Se uma variável não básica $X_{ij}=U_{ij}$ se e somente se $C_{ij}^{\pi} \leq 0$.

Se alguma das condições acima não for satisfeita, a função objetivo pode ser melhorada.

18.10 Pivoteamente no simplex para rede (melhoramento da F.O.)

Uma vez detectada uma variável a entrar na base, o seu arco deve fazer parte da nova arvore geradora, algum arco da arvore deve sair. Para tanto basta aplicar a lei de conservação de fluxo aos nós que estão no ciclo formado com a inclusão deste novo arco.

Aumentando o fluxo do arco que entra na base aquele arco do ciclo que chegar primeiro em um dos seus limites será o arco a sair da arvore.

Com a nova base, voltar para o calculo dos coeficientes da F.O.

Exemplo:

Calculando Os coeficientes da F.O.

$$\pi_{(1)} = 0$$
$$C_{13} = \pi_{(1)} - \pi_{(3)},$$
$$C_{35} = \pi_{(3)} - \pi_{(5)},$$
$$C_{25} = \pi_{(2)} - \pi_{(5)},$$
$$C_{45} = \pi_{(4)} - \pi_{(5)}$$

$$\pi_{(1)} = 0$$
$$\pi_{(3)} = -12$$
$$\pi_{(3)} - \pi_{(5)} = 7 \Rightarrow \pi_{(5)} = -19$$
$$6 = \pi_{(2)} + 19 \Rightarrow \pi_{(2)} = -13$$
$$3 = \pi_{(4)} + 19 \Rightarrow \pi_{(4)} = -16$$

$C_{12}^{\pi} = C_{12} - \pi_{(1)} + \pi_{(2)} \Rightarrow 0 - 10 + (-13) = -3$ in-kilter

$C_{14}^{\pi} = C_{14} - \pi_{(1)} + \pi_{(4)} \Rightarrow 6 - 0 + (-16) = -10$ in-kilter

$C_{32}^{\pi} = C_{32} - \pi_{(3)} + \pi_{(2)} \Rightarrow 2 - (-12) + (-13) = 1$ in-kilter

$C_{34}^{\pi} = C_{34} - \pi_{(3)} + \pi_{(4)} \Rightarrow 3 - (-12) + (-16) = -1$ out-of-kilter

Portanto como $X_{34}=0$ e $C_{34}^{\pi} < 0$ então X_{34} deve crescer tanto quanto possível. Para cada unidade de fluxo no arco (3,4) causará o decréscimo de 1 unidade na função objetivo.

$X_{34} = \theta$

$X_{45} = 4 + \theta$

$X_{35} = 1 - \theta$

$\theta_{\max} = \{\theta_{34} = 5, \theta_{45} = 2, \theta_{35} = 1\}$

$\theta_{\max} = 5$

BIBLIOGRAFIA

Ahuja, R.K., Magnanti, T.L. & Orlin, J.B. (1993) *Network Flows: Theory, Algorithms and Applications*. Ed. Prentice-Hall, New Jersey.

Goldbarg, M. C. e Luna, H. P. L. (2000) Otimização Combinatória e Programação Linear: Modelos e Algoritmos. Editora Campus, Rio de Janeiro.

Lachtermacher, G (2002) Pesquisa Operacional na Tomada de Decisões. Editora Campus, Rio de Janeiro.

Lindo Systems Inc. (2001) LINGO: the modeling language and optmizer, Chicago.

Winston, W. L. (1993) Operations Research: Applications and algorithms, Duxbury Press, Belmont.