Problemas de Decisão em Redes

Transparências de apoio à leccionação de aulas teóricas

Slide 1

Versão 2

José Fernando Oliveira – FEUP

Problemas de Fluxos em Redes

Slide 2

Problemas de fluxos em redes

Rede: Conjunto de pontos (vértices) ligados por linhas ou canais (ramos ou arcos).

Objectivo: Enviar um certo tipo de bem a partir de certos pontos e receber esses bens noutros pontos.

Exemplos:

Slide 3

- linhas de comunicação
- redes de estradas, comboios ou de aviões
- redes de gás, electricidade ou água

Problemas de Transportes

Slide 4

O problema da distribuição de frigoríficos

Slide 5

Um fabricante de frigoríficos tem 3 fábricas, de onde abastece 3 clientes (distribuidores). No início de cada mês recebe de cada cliente a informação sobre o número de frigoríficos que pretende para esse mês. Esses frigoríficos terão que ser produzidos nas várias fábricas, atendendo à capacidade de produção de cada uma delas. O custo de transportar um frigorífico de cada fábrica para cada cliente é conhecido. O problema consiste em determinar que fábrica(s) deve(m) abastecer cada cliente, e em que quantidades, de forma a que, respeitando as capacidades de produção das fábricas e satisfazendo as necessidades dos clientes, o custo total de transporte seja minimizado. Formule este problema considerando que as capacidades de produção são iguais em todas as fábricas (20 frigoríficos) e que as necessidades dos clientes são de 10, 30 e 20 frigoríficos. Os custos unitários de transporte são os indicados na tabela seguinte (em milhares de escudos):

		Clientes		
		1	*2*	*3*
Fábricas	*1*	2	4	3
	2	1	5	2
	3	1	1	6

Modelo

x_{ij} – quantidade a transportar da fábrica i para o cliente j

$$\min \quad 2x_{11} + 4x_{12} + 3x_{13} + x_{21} + 5x_{22} + 2x_{23} + x_{31} + x_{32} + 6x_{33}$$

suj. a:

Slide 6

$$\begin{array}{ccccccccccc}
x_{11} & +x_{12} & +x_{13} & & & & & & & \leq & 20 \\
 & & & x_{21} & +x_{22} & +x_{23} & & & & \leq & 20 \\
 & & & & & & x_{31} & +x_{32} & +x_{33} & \leq & 20 \\
x_{11} & & & +x_{21} & & & +x_{31} & & & \geq & 10 \\
 & x_{12} & & & +x_{22} & & & +x_{32} & & \geq & 30 \\
 & & x_{13} & & & +x_{23} & & & +x_{33} & \geq & 20 \\
x_{11}, & x_{12}, & x_{13}, & x_{21}, & x_{22}, & x_{23}, & x_{31}, & x_{32}, & x_{33} & \geq & 0
\end{array}$$

Estrutura de um Problema de Transportes (PT)

Slide 7

- *origens* onde existe um bem ou serviço *disponível* (em quantidades limitadas);
- *destinos* onde esse bem ou serviço é *necessário*;
- existem, e são conhecidos, *custos unitários de "transportar"* entre cada origem e cada destino;
- o objectivo é determinar a política óptima de transportes, isto é, aquela que satisfazendo as necessidades e respeitando as disponibilidades, *minimiza o custo total de transporte.*

Modelo do PT

x_{ij} – quantidade a transportar da origem i para o destino j;

c_{ij} – custo de transportar uma unidade da origem i para o destino j;

d_i – disponibilidade na origem i;

n_j – necessidade no destino j.

Slide 8

$$\min \sum_i \sum_j c_{ij} x_{ij}$$

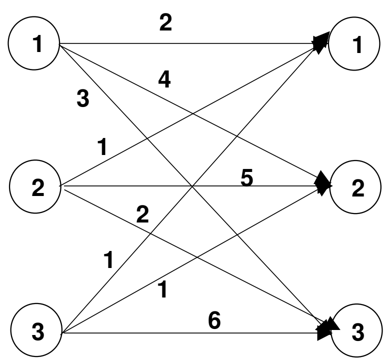
suj. a:

$$\begin{aligned} \sum_j x_{ij} &\leq d_i, \forall_i \qquad \text{(restrições da oferta)} \\ \sum_i x_{ij} &\geq n_j, \forall_j \qquad \text{(restrições de procura)} \\ x_{ij} &\geq 0 \end{aligned}$$

Distribuição de frigoríficos – a rede de transportes

Slide 9

Disponibilidades	Origens	Destinos	Necessidades
$d_1 = 20$			$n_1 = 10$
$d_2 = 20$			$n_2 = 30$
$d_3 = 20$			$n_3 = 20$



$\sum_i d_i = 60$ Disponibilidades totais = $\sum_j n_j = 60$ Procura total

Problema na forma “standard”

Forma "standard" de um PT

$$\boxed{\sum \text{oferta} = \sum \text{procura}}$$

$$\Updownarrow$$

Tudo o que está nas origens é transportado para os destinos.

$$\Downarrow$$

As restrições são satisfeitas nas igualdades.

Slide 10

Somando

- as equações das restrições da oferta
- as equações das restrições da procura

*o*btém-se a mesma equação!

$$\sum_i \sum_j x_{ij} = \boxed{\sum_i d_i} \searrow$$

iguais

$$\sum_j \sum_i x_{ij} = \boxed{\sum_j n_j} \nearrow$$

— Equações linearmente dependentes $\longrightarrow$ há uma equação a mais

Conclusão: Num PT na forma "standard" só é necessário considerar $\boxed{\text{nº de origens + nº de destinos -1}}$ equações.

Resolução de um PT

- modelo de PT é um modelo de PL $\Rightarrow$ método Simplex;
- são problemas com uma estrutura particular $\Rightarrow$ desenvolvimento de um algoritmo específico, que tira partido dessas particularidades, baseado no Simplex e noutros conceitos de PL avançada.

Slide 11

Quadro para o algoritmo de transportes

	Destinos 1	2	3	
Origens 1	2	4	3	20
2	1	5	2	20
3	1	1	6	20
	10	30	20	

$\Leftrightarrow$ Formulação como um PT

Geração de uma solução inicial (i)

Regra dos custos mínimos

Slide 12

— (2)	10 (4)	10 (3)	~~20~~ ~~10~~ 0
10 (1)	— (5)	10 (2)	~~20~~ ~~10~~ 0
— (1)	20 (1)	— (6)	~~20~~ 0
~~10~~ 0	~~30~~ ~~10~~ 0	~~20~~ ~~10~~ 0	

Geração de uma solução inicial (ii)

Regra do canto NW

Slide 13

10 (2)	10 (4)	— (3)	~~20~~ ~~10~~ 0
— (1)	20 (5)	— (2)	~~20~~ 0
— (1)	— (1)	20 (6)	~~20~~ 0
~~10~~ 0	~~30~~ ~~20~~ 0	~~20~~ 0	

Soluções iniciais (básicas) degeneradas

Slide 14

10 2	**10** 4	**--** 3
-- 1	**20** 5	**--** 2
-- 1	**--** 1	**20** 6

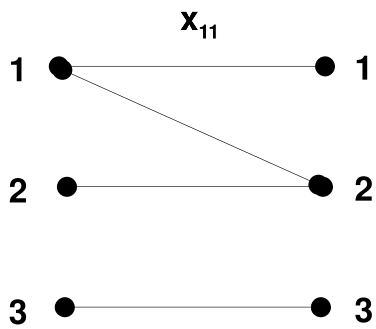
- Apenas 4 variáveis diferentes de zero.
- n + m - 1 = 5 variáveis básicas necessárias.

Logo, teremos que "promover" uma variável nula a básica. (Solução básica degenerada)

Regra para escolha da variável nula que será considerada básica:

> *O grafo representativo das variáveis básicas terá que ser uma árvore e conexo*

Logo, poderemos tomar x_{13}, x_{23}, x_{31} ou x_{32}, mas **não** x_{21}.



Algoritmo de transportes (i)

Slide 15

1 – Cálculo dos custos marginais

Para cada variável básica x_{ij} definem-se dois custos (os custos marginais) que somados dão o custo unitário de transporte dessa variável básica, c_{ij}.

O custo de transportar uma unidade de i para j pode ser entendido como a soma de 2 custos:

- u_i (custo de despacho em i)
- v_j (custo de recepção em j)

É necessário arbitrar um destes custos para uma das variáveis básicas (normalmente arbitra-se 0 para o custo do canto superior esquerdo).

	0	**2**	**1**
2	**10** 2	**10** 4	**0** 3
3	**--** 1	**20** 5	**--** 2
5	**--** 1	**--** 1	**20** 6

Algoritmo de transportes (ii)

2 – Cálculo das diferenças

Para todas as variáveis *não básicas* calcula-se um $\Delta_{ij} = c_{ij} - [u_i + v_j]$. Entrará na base a variável não básica com Δ_{ij} mais negativo (problema de minimização).

Slide 16

Ao entrar na base esta variável deixa de valer 0 passando a valer Θ, positivo. O valor das variáveis básicas altera-se de forma a respeitarem-se disponibilidades e necessidades. O valor de Θ será tal que nenhuma variável venha negativa.

	0	2	1
2	10 2	10 4	0 3
3	-- -2 1	20 5	-- -2 2
5	-- -4 1	-- -6 1	20 6

–

	0	2	1
2	10 2	10-θ 4	0+ θ 3
3	-- -2 1	20 5	-- -2 2
5	-- -4 1	θ -6 1	20 - θ 6

θ = 10

Continuando a resolver...

Slide 17

	0	-4	1
2	10-θ 2	-- 6 4	10+θ 3
9	θ -8 1	20-θ 5	-- -8 2
5	-- -4 1	10+θ 1	10 - θ 6

θ = 10

	0	4	1
2	0-θ 2	θ -2 4	20 3
1	10+θ 1	10-θ 5	-- 0 2
-3	-- 4 1	20 1	-- 8 6

θ = 0

	0	4	3
0	-- 2 2	0+θ 4	20 –θ 3
1	10 1	10-θ 5	θ -2 2
-3	-- 4 1	20 1	-- 6 6

θ = 10

	0	2	1
2	-- 0 2	10 4	10 3
1	10 1	-- 2 5	10 2
-1	-- 2 1	20 1	-- 6 6

Solução óptima

$x_{12} = 10$, $x_{13} = 10$, $x_{21} = 10$, $x_{23} = 10$, $x_{32} = 20$

Custo óptimo =

$10 \times 4 + 10 \times 3 + 10 \times 1 + 10 \times 2 + 20 \times 1 = 120$

Exemplo

Slide 18

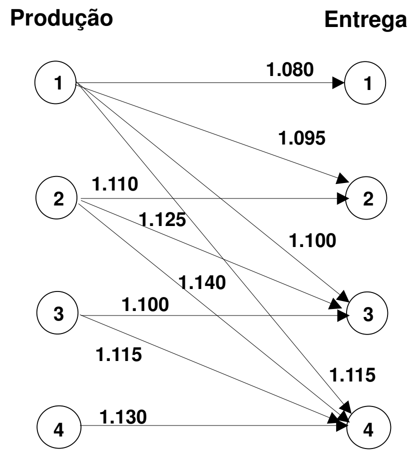
Uma companhia construtora de aviões pretende planear a produção de um motor, para os próximos 4 meses. Para satisfazer as datas de entrega contratuais necessita de fornecer os motores nas quantidades indicadas na segunda coluna do quadro. O número máximo de motores que a companhia produz por mês, bem como o custo de cada motor (em milhões de dólares) são dados na terceira e quarta colunas do referido quadro. Dadas as variações nos custos de produção, pode valer a pena produzir alguns motores um ou mais meses antes das datas programadas para entrega. Se se optar por esta hipótese, os motores serão armazenados até ao mês de entrega, com um custo adicional de 0.015 milhões de dólares/mês.

Mês	Quantidades a entregar	Produção máxima	Custo unitário de produção	Custo unitário de armazenagem
1	10	25	1.08	—
2	15	35	1.11	0.015
3	25	30	1.10	0.015
4	20	10	1.13	0.015

O director de produção quer saber quantos motores deve fabricar em cada mês (e para que meses de entrega) por forma a minimizar os custos globais de produção e armazenagem. Formule o problema como um problema de transportes.

Resolução

Slide 19



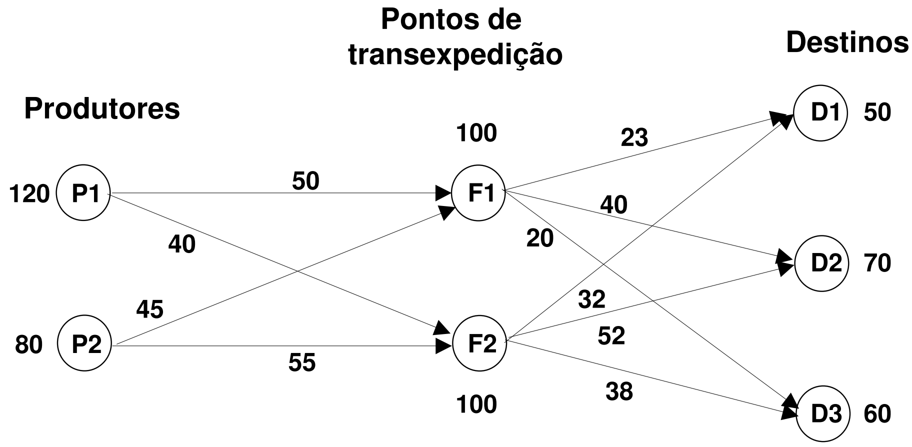
Mês de entrega

Mês de produção	1	2	3	4	x	
1	1.080	1.095	1.110	1.125	0	25
2	∞	1.110	1.125	1.140	0	35
3	∞	∞	1.100	1.115	0	30
4	∞	∞	∞	1.130	0	10
	10	15	25	20	30 /100	100

Problemas de transexpedição

Slide 20

Exemplo: Problemas de transexpedição (extensão do PT)



Questões adicionais: — Restrições de capacidade nas fronteiras (pontos de transexpedição) — Equilíbrio de fluxos

Formulação de um problema de transexpedição como um PT'

Slide 21

	D1	D2	D3	F1	F2	X	
P1	∞	∞	∞	50	40	0	120
P2	∞	∞	∞	45	55	0	80
F1	23	40	20	0	∞	∞	100
F2	34	52	38	∞	0	∞	100
	50	70	60	100	100	20	

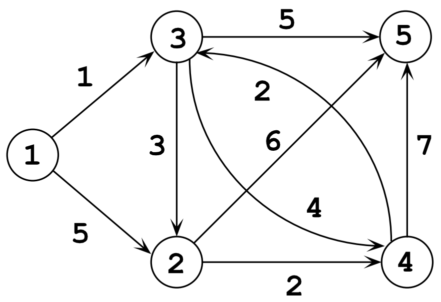
Notas:

- o que é transportado de um ponto de transexpedição para si próprio representa a capacidade de transexpedição não usada;
- caso a capacidade dos pontos de transexpedição seja infinita, deve-se usar o valor do somatório da procura ou da oferta.

Formulação de problemas de fluxos em redes genéricas

Slide 22

Exemplo



Nó		
1	Disponibilidade =	50
2	Necessidade =	40
3	Intermédio	
4	Disponibilidade =	10
5	Necessidade =	20

Restrições:

Nós fornecedores: Fluxo que entra + Disponibilidades = Fluxo que sai

Nós consumidores: Fluxo que entra - Necessidades = Fluxo que sai

Nós intermédios: Fluxo que entra = Fluxo que sai

Modelo de PL para problemas de fluxos em redes genéricas

Slide 23

x_{ij} – fluxos nos ramos

$$\min 5x_{12} + x_{13} + 2x_{24} + 6x_{25} + 3x_{32} + 4x_{34} + 5x_{35} + 2x_{43} + 7x_{45}$$

suj. a:

$$\begin{array}{rrrrrrrrrcr}
x_{12} & +x_{13} & & & & & & & & = & 50 \\
-x_{12} & & +x_{24} & +x_{25} & -x_{32} & & & & & = & -40 \\
 & -x_{13} & & & +x_{32} & +x_{34} & +x_{35} & -x_{43} & & = & 0 \\
 & & -x_{24} & & & -x_{34} & & +x_{43} & +x_{45} & = & 10 \\
 & & & -x_{25} & & & -x_{35} & & -x_{45} & = & -20 \\
 & & & & & & & & x_{ij} & \geq & 0, \forall_{i,j}
\end{array}$$

$\boxed{\text{1 equação redundante!}}$

Bibliografia

- Hillier, Frederick S. e Lieberman, Gerald (1995). *Introduction to Operations Research*, Mc Graw-Hill.
- Oliveira, José Fernando (1996). *Apontamentos de Investigação Operacional 1.* FEUP.

Slide 24

Problemas de Afectação

Slide 25

Problemas de Afectação (PA)

Exemplo típico: Afectação de n pessoas a n tarefas.

Dados: Tempo que cada pessoa demora a executar cada tarefa.

Objectivo: Minimizar o tempo total.

Modelo:

Slide 26

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

suj. a:

$$\begin{aligned} \sum_{j=1}^{n} x_{ij} &= 1, \qquad i = 1 \ldots n \\ \sum_{i=1}^{n} x_{ij} &= 1, \qquad j = 1 \ldots n \\ x_{ij} &\in \{0,1\}, \qquad \forall_{i,j} \end{aligned}$$

Caso particular de um PT em que:

$$\begin{aligned} d_i &= 1, \qquad \forall_i \\ n_j &= 1, \qquad \forall_j \end{aligned}$$

Resolução do PA

Slide 27

Método Húngaro – Deve-se ao matemático húngaro König.

Pressupostos:

- $c_{ij} \geq 0$
- problema de minimização

Quadro de resolução:

	T_1	T_2	T_3	T_4
M_1	10	9	8	7
M_2	3	4	5	6
M_3	2	1	1	2
M_4	4	3	5	6

Se fosse possível identificar uma afectação de custo nulo no quadro de resolução então, dentro dos pressupostos enunciados, essa solução seria óptima.

O método húngaro vai operar sucessivas transformações sobre o quadro, transformando-o em quadros equivalentes (com as mesmas soluções) mas com mais zeros, até que seja evidente uma solução de custo nulo. No fim basta reproduzir essa solução sobre o quadro inicial para termos o custo real da afectação.

Justificação do método húngaro

Slide 28

Princípio de funcionamento: Somar ou subtrair uma constante a uma linha ou a uma coluna da matriz dos custos de afectação (problema na forma "standard"), não altera a afectação óptima (embora altere o seu custo).

Justificação: Se, por hipótese, se reduzem todos os custos da linha 1 de k, a função objectivo do PA fica

$$\min \sum_{j=1}^{n} (c_{1j} - k)x_{1j} + \sum_{i=2}^{n} \sum_{j=1}^{n} c_{ij}x_{ij} =$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij} - k \sum_{j=1}^{n} x_{1j}$$

Como $\sum_{j=1}^{n} x_{ij} = 1$ a nova função objectivo só difere da anterior da constante k, logo, a solução óptima do respectivo programa linear não é afectada.

Algoritmo de aplicação do método húngaro

Metodologia: Subtrair custos suficientemente elevados às várias linhas e colunas de modo a que a afectação óptima seja encontrada por inspecção.

Algoritmo:

Slide 29

1. subtrair a cada linha o menor elemento dessa linha;
2. subtrair a cada coluna o menor elemento dessa coluna;
3. riscar as linhas e colunas em que algum dos elementos vale zero usando o menor número de riscos possível (*Sugestão:* riscar primeiro as linhas ou colunas com maior número de zeros ainda não riscados costuma resultar!);
4. se o número de linhas e colunas riscadas for igual a n (número de itens a afectar), encontrou-se a afectação óptima: ela é constituída por n zeros **independentes**;
5. senão, selecciona-se o menor elemento não riscado e subtrai-se a todos os elementos não riscados e adiciona-se a todos os elementos que estão no cruzamento de dois riscos. Volta-se ao ponto 3.

Layout fabril

Slide 30

Uma fábrica possui 4 locais (1,2,3,4) para receber 3 máquinas novas (A,B,C). O local 4 é demasiado pequeno para conter a máquina A. O custo de manipulação dos materiais que são processados nas máquinas, em centenas de escudos/hora, envolvendo cada máquina e as respectivas posições, é o seguinte:

	1	2	3	4
A	5	1	3	X
B	3	1	4	3
C	3	3	4	2

O objectivo é determinar que local ocupará cada uma das novas máquinas, de forma a minimizar o custo total de manipulação dos materiais.

Resolução do problema de layout fabril

Slide 31

$$\begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ \hline A & 5 & 1 & 3 & \infty \\ B & 3 & 1 & 4 & 3 \\ C & 3 & 3 & 4 & 2 \\ F & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$\begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ \hline A & 4 & 0 & 2 & \infty \\ B & 2 & 0 & 3 & \boxed{2} \\ C & 1 & 1 & 2 & 0 \\ F & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

3 riscos < 4

$$\begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ \hline A & 2 & \boxed{0} & 0 & \infty \\ B & \boxed{0} & 0 & 1 & 0 \\ C & 1 & 3 & 2 & \boxed{0} \\ F & 0 & 2 & \boxed{0} & 0 \\ \hline \end{array}$$

4 riscos

Solução óptima

Soluções óptimas:
{A2, B1, C4, F3}
ou
{A3, B2, C4, F1}
Custo: 6

Asa de Luxo Lda.

A empresa de transportes Asa de Luxo comprou 3 novos pequenos aviões. Após um estudo de mercado foram identificados 4 possíveis destinos para os novos voos a estabelecer: Monte Carlo, Ilhas Canárias, Biarritz e as Ilhas Gregas. Para cada um dos destinos foi estimado o lucro que cada avião proporcionaria:

Slide 32

Destino	A_1	A_2	A_3
Monte Carlo	8	11	10
Ilhas Canárias	10	9	9
Biarritz	9	4	8
Ilhas Gregas	6	7	5

(lucros em M$)

Numa reunião, o administrador da Asa de Luxo (que possui um apartamento em Biarritz) decidiu que Biarritz seria necessariamente o destino de um dos 3 aviões. Por outro lado, o Director de Marketing considerou que, por uma questão de estratégia, se deveria atingir o maior número possível de destinos, não enviando mais do que um avião para cada destino. O responsável pela manutenção chamou a atenção para o facto de os aviões A_1 e A_3 não poderem aterrar nas Ilhas Gregas.

Decida que avião deve seguir para cada destino e ganhe uma viagem grátis para um destino à sua escolha (oferecida pela Asa de Luxo, obviamente!)

Resolução do problema da Asa de Luxo Lda.

Slide 33

Problema de **maximização** ⇒ calcular o complemento para o máximo (neste caso 11) de todos os elementos da matriz e resolver o problema como se fosse de minimização.

$$\begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ \mathrm{F} \end{array}\begin{array}{cccc} \mathrm{MC} & \mathrm{IC} & \mathrm{B} & \mathrm{IG} \\ \hline 8 & 10 & 9 & 6 \\ \boxed{11} & 9 & 4 & 7 \\ 10 & 9 & 8 & 5 \\ 0 & 0 & 0 & 0 \end{array} \qquad \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ \mathrm{F} \end{array}\begin{array}{cccc} \mathrm{MC} & \mathrm{IC} & \mathrm{B} & \mathrm{IG} \\ \hline 3 & 1 & 2 & 5 \\ 0 & 2 & 7 & 4 \\ 1 & 2 & 3 & 6 \\ 11 & 11 & 11 & 11 \end{array}$$

$$\begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ \mathrm{F} \end{array}\begin{array}{cccc} \mathrm{MC} & \mathrm{IC} & \mathrm{B} & \mathrm{IG} \\ \hline 3 & 1 & 2 & \infty \\ 0 & 2 & 7 & 4 \\ 1 & 2 & 3 & \infty \\ 11 & 11 & \infty & 11 \end{array} \qquad \cdots$$

Bibliografia

Slide 34

- Ferreira, José António Soeiro (1995). *Apontamentos de Investigação Operacional 1.* FEUP.
- Hillier, Frederick S. e Lieberman, Gerald (1995). *Introduction to Operations Research*, Mc Graw-Hill.
- Oliveira, José Fernando (1996). *Apontamentos de Investigação Operacional 1.* FEUP.

Problemas de Fluxo Máximo

Slide 35

Slide 36

Problemas de Fluxo Máximo

Definição: Dada uma rede, com um *nó de entrada* e um *nó de saída*, com *capacidades* associadas a cada ramo, pretende-se saber qual é o fluxo máximo, de um certo bem, que se pode enviar da entrada para a saída.

Modelo:

x_{ij} – fluxo que passa no ramo (i, j), de i para j
c_{ij} – capacidade do ramo (i, j)
Nó 1 – Nó de entrada
Nó t – Nó de saída

$$\max \sum_j x_{1j} = \sum_i x_{it}$$

suj. a:

$$\begin{array}{rcll} \sum_i x_{ik} & = & \sum_j x_{kj}, \forall_k & \text{(equilíbrio de fluxos nos nós)} \\ x_{ij} & \leq & c_{ij}, \forall_{i,j} & \text{(restrições de capacidade)} \\ x_{ij} & \geq & 0, \forall_{i,j} & \end{array}$$

Algoritmo de fluxo máximo

Slide 37

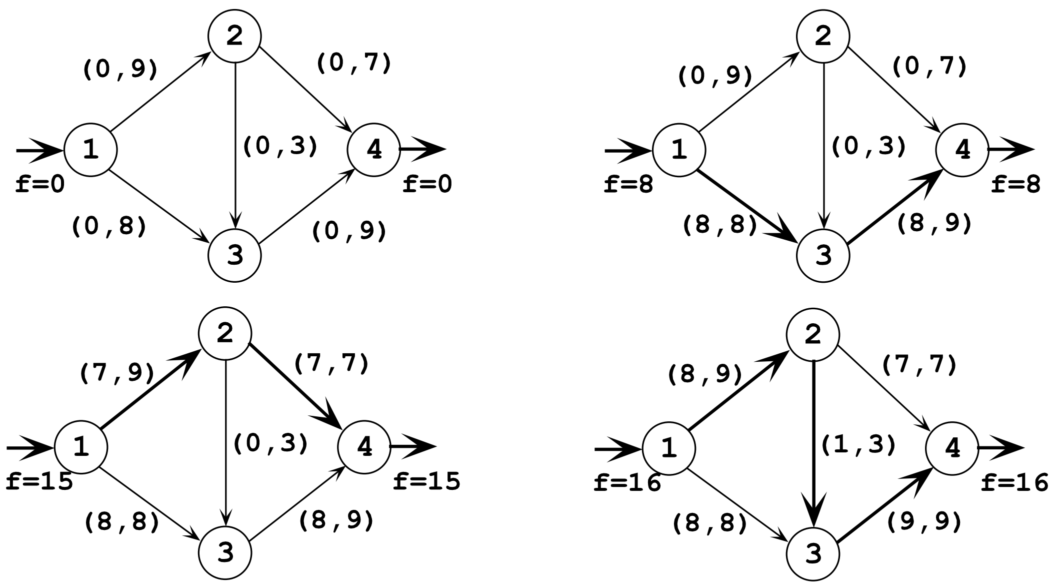
1. injectar um fluxo nulo no nó de entrada;
2. capacidades iniciais dos ramos = capacidade total dos ramos;
3. determinar um *caminho não saturado* (capacidade $\neq 0$) entre o nó de entrada e o nó de saída; se não existir, foi encontrada a SOLUÇÃO ÓPTIMA;
4. somar ao fluxo de entrada um fluxo igual à *capacidade do caminho* seleccionado;
5. alterar as capacidades dos ramos do caminho seleccionado, diminuindo-lhes o fluxo injectado;
6. voltar ao ponto 3.

Notas:

- **caminho** – conjunto de ramos, unindo o nó de entrada ao nó de saída e que não passa duas vezes pelo mesmo nó;
- **capacidade de um caminho** – menor capacidade disponível de entre todos os ramos que fazem parte do caminho;
- **caminho saturado** – caminho com capacidade nula;

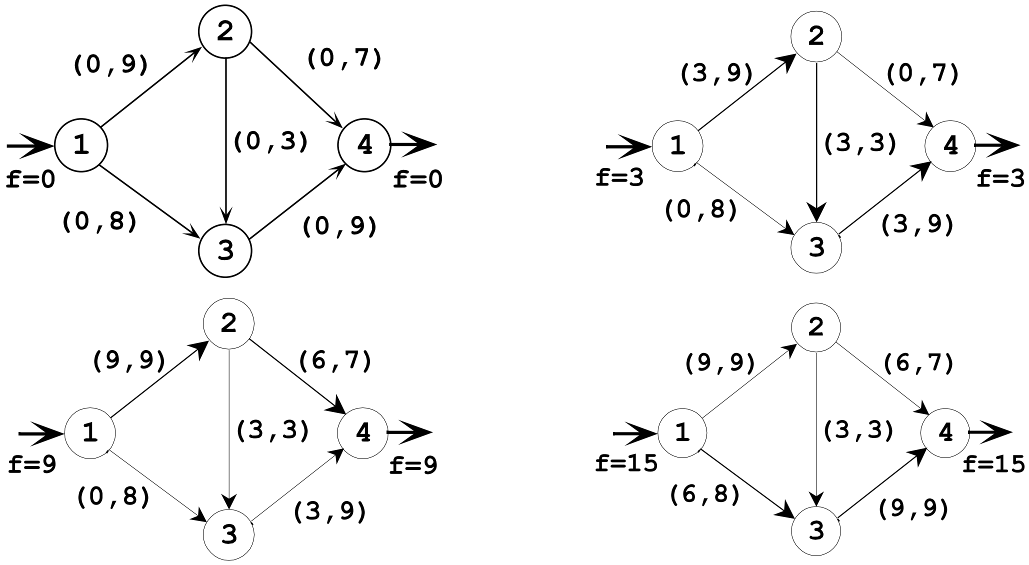
Exemplo de um problema de fluxo máximo

Slide 38



E se os caminhos fossem saturados por outra ordem?

Slide 39



Aparentemente não há nenhum caminho não saturado entre a entrada e a saída e o fluxo "máximo" deu menor do que no caso anterior...

Não é este um algoritmo de optimização?

Fluxos "negativos"

De facto o exemplo anterior não está completamente resolvido (até à optimalidade) porque existe ainda um caminho não saturado. Para se encontrar este caminho não saturado é necessário considerar o conceito de fluxo "negativo", isto é, fluxo que atravessa os ramos no sentido contrário à sua orientação.

Slide 40

Uma das restrições apresentadas no modelo do problema de fluxo máximo impunha que todos os fluxos fossem positivos ou nulos ($x_{ij} \geq 0$). E de facto, na solução final tal terá sempre que acontecer. No entanto, essa solução final obtém-se pela adição dos vários fluxos que vamos injectando na rede. Alguns desses fluxos poderão atravessar algum ramo no sentido contrário ao indicado, devendo nesse caso ser contabilizados como negativos. O resultado final (soma de todos os fluxos que foram injectados nesse ramo) é que terá que ser positivo ou nulo. No fundo um fluxo "negativo" mais não é do que deixar de fazer passar fluxo por esse ramo.

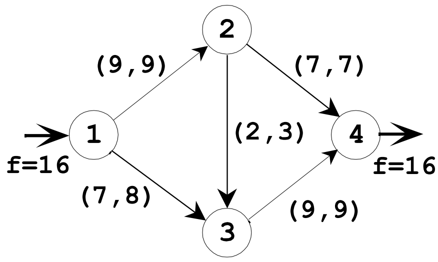
Fluxos "negativos"

Observando a rede deste ponto de vista conclui-se que o caminho 1-3-2-4 não está saturado:

- O ramo 1-3 tem uma capacidade de 2.
- O ramo 2-3, visto do lado do nó 2, está saturado. No entanto visto do lado do nó 3, que é o lado que nos interessa visto o caminho o precorrer de 3 para 2, tem uma capacidade de 3, que é o fluxo que o atravessa de 2 para 3.
- O ramo 2-4 tem uma capacidade de 1.

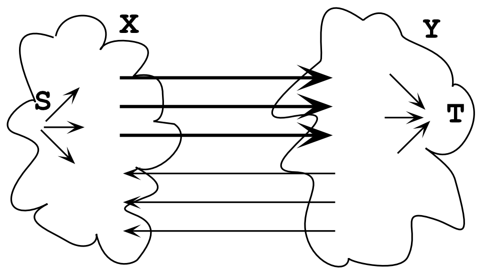
Slide 41

Então a capacidade do caminho é 1 e é possível injectar mais uma unidade de fluxo na rede:



Consegue-se ter a certeza se uma rede está na situação de fluxo máximo através da relação entre fluxo máximo e cortes mínimos.

Fluxos máximos e cortes mínimos



Definição: Um **corte** numa rede com nó de entrada S e nó de saída T é um conjunto de arcos cuja remoção separa a rede em duas partes, X e Y, uma contendo S e outra contendo T.

Slide 42

A **capacidade de um corte** é a soma das capacidades dos arcos do corte, que estão dirigidos de X para Y. Um **corte mínimo** é um corte com a menor capacidade possível.

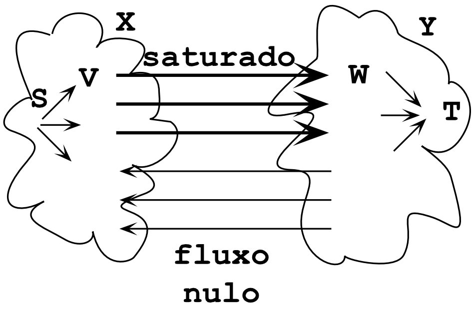
$$\begin{aligned}\text{Valor de qualquer fluxo} &\leq \text{Capacidade de qualquer corte}\\ \text{Valor do fluxo máximo} &\leq \text{Capacidade de qualquer corte}\\ \text{Valor do fluxo máximo} &\leq \text{Capacidade de um corte mínimo}\end{aligned}$$

Teorema do Fluxo Máximo–Corte Mínimo

Teorema: O valor do fluxo máximo é igual à capacidade do corte mínimo.

Slide 43

Demonstração: Seja uma rede na situação de fluxo máximo. Seja X o conjunto de nós que pode ser atingido a partir de S através de um caminho não saturado, e seja Y o conjunto dos restantes nós. $T \in Y$ porque senão $T \in X$ e não se estava numa situação de fluxo máximo.



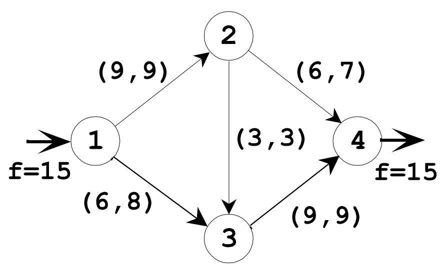
Considere-se o corte composto pelos ramos com uma extremidade em X e outra em Y. Todos os arcos dirigidos de um nó V em X para um nó W em Y estão saturados, pois caso contrário W pertenceria a X e não a Y. Qualquer arco dirigido de Y para X terá fluxo nulo pois caso contrário isso seria um retorno de fluxo de Y para X que, se fosse anulado, aumentaria o fluxo que efectivamente chega a T, o que contraria a hipótese inicial de a rede estar na situação de fluxo máximo. Então, a capacidade do corte, que é igual ao fluxo de X para Y, é igual ao fluxo na rede, que é máximo.

Aplicação de cortes a redes de fluxos

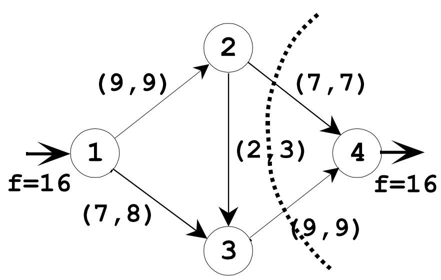
Uma rede está então na situação de fluxo máximo se existir um corte mínimo, isto é que separa a entrada da saída da rede e que só atravessa ramos orientados da entrada para a saída saturados e ramos orientados da saída para a entrada com fluxo nulo.

Slide 44

Nesta rede não existe um corte com essas propriedades.



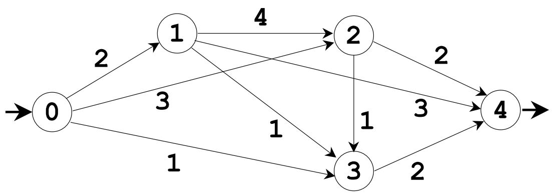
O corte constituído pelos ramos 2-4 e 3-4 é um corte mínimo.



Exercício

Slide 45

Determine a quantidade máxima de um produto que pode ser enviada através da rede seguinte, entre a origem e o destino 4. Existem limitações nas quantidades que podem atravessar cada arco, encontrando-se as respectivas quantidades máximas representadas junto a cada arco. Considere que esse produto se encontra disponível na origem 0 em quantidade ilimitada.

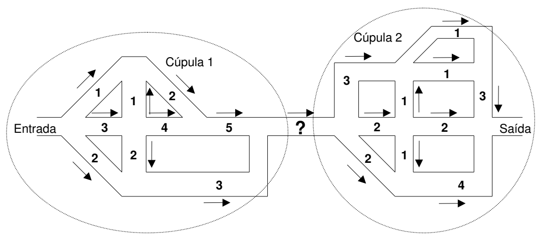


Em que arco(s) aumentaria as quantidades máximas admissíveis, de forma a conseguir aumentar a quantidade máxima de produto que é possível fazer passar pela rede?

Exercício

Slide 46

Uma parte do ShopShopping vai ser construída imitando uma plataforma de exploração subaquática: duas grandes cúpulas ligadas por um grande corredor. Para que a circulação de pessoas no centro comercial decorra de uma forma fluida, é necessário que este corredor não restrinja o fluxo máximo que pode atravessar a secção subaquática do centro comercial. Na figura seguinte representa-se esquematicamente a planta desta parte do ShopShopping.



Em cada corredor está indicada a capacidade (em dezenas de pessoas por minuto) de circulação nesse corredor. O corredor de ligação entre as duas cúpulas está ainda por dimensionar, dado o seu elevado custo, crescente com o aumento de capacidade que se lhe queira atribuir. Note que por questões de segurança e fluidez de circulação os corredores funcionam como caminhos de sentido único (ver setas na figura).

Resolvendo este problema como de fluxo máximo indique qual deve ser a capacidade do corredor de ligação de forma a que o fluxo que atravessa as cúpulas seja máximo e o custo do corredor de ligação o menor possível.

Bibliografia

Slide 47

- Dolan, Alan e Aldous, Joan (1993). *Networks and Algorithms: an introductory approach.* John Wiley and Sons.
- Oliveira, José Fernando (1996). *Apontamentos de Investigação Operacional 1.* FEUP.
- Taha, Hamdy A. (1997). *Operations Research, an Introduction.* Prentice Hall.

Problemas de Caminho Mínimo

Slide 48

Problemas de Caminho Mínimo

Definição: Determinar o caminho mais curto entre o nó de entrada e o nó de saída de uma rede. A cada ramo (i, j) está associada uma "distância" não negativa d_{ij}. É admissível que $d_{ij} \neq d_{ji}$.

Slide 49

E é um problema de fluxos? Basta reinterpretá-lo como um problema de enviar uma unidade de um bem, do nó de entrada para o nó de saída, minimizando o custo de transporte, em que d_{ij} representa o custo unitário de transporte de i para j.

Pode ser formulado, e resolvido, como um problema de PL, mas...

Problemas de Caminho Mínimo - Modelo

x_{ij}	–	quantidade a transportar do nó i para o nó j; $x_{ij} \in \{0,1\}$;
d_{ij}	–	custo de transportar uma unidade do nó i para o nó j;
I	$\in$	$\{INI, 1, 2, \ldots, n\}$;
F	$\in$	$\{1, 2, \ldots, n, FIM\}$;
K	$\in$	$\{1, 2, \ldots, n\}$.

Slide 50

$$\min \sum_i \sum_j d_{ij} x_{ij}$$

suj. a:

$$\sum_{i \in I} x_{ij} = \sum_{k \in F} x_{jk}, \forall_{j \in K} \qquad \text{(equilíbrio dos nós)}$$

$$\sum_{i \in I} x_{ij} \leq 1, \forall_{j \in F} \qquad \text{(a cada nó só pode chegar um caminho)}$$

$$\sum_{j \in F} x_{ij} \leq 1, \forall_{i \in I} \qquad \text{(de cada nó só pode sair um caminho)}$$

$$\sum_{j \in F} x_{INIj} = 1, \qquad \text{(do nó inicial tem que sair um só caminho)}$$

$$\sum_{i \in I} x_{iFIM} = 1, \qquad \text{(ao nó final tem que chegar um só caminho)}$$

Algoritmo de Dijkstra – metodologia

Atribuir a todos os nós uma etiqueta, que poderá ser:

- provisória – limite superior para a distância mais curta entre o nó de entrada e o nó em causa (menor distância conhecida até ao momento);
- definitiva – distância mais curta entre o nó de entrada e o nó em causa.

Slide 51

Inicialmente é atribuída ao nó de entrada uma etiqueta definitiva com o valor 0. A todos os outros nós é atribuída uma etiqueta provisória com valor igual a ∞.

O algoritmo tenta transformar as etiquetas provisórias em definitivas. Quando o nó de saída tiver uma etiqueta definitiva o problema foi resolvido.

Algoritmo de Dijkstra

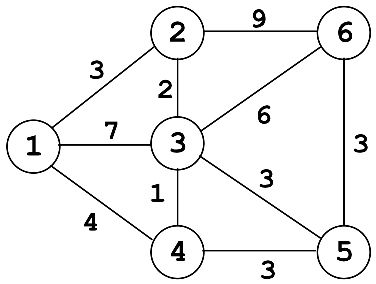
Slide 52

1. atribuir uma etiqueta definitiva ao nó de entrada, com valor 0, e etiquetas provisórias a todos os outros nós, com valor igual a ∞;
2. seja k o nó que mais recentemente recebeu uma etiqueta definitiva. Para todos os nós i ainda com etiquetas provisórias calcule-se o valor da etiqueta definitiva de k mais a distância **directa** entre k e i (d_{ki}). O mínimo, entre este valor e a anterior etiqueta provisória do nó i, é tomado como nova etiqueta provisória de i.
3. seleccionar a menor das etiquetas provisórias e declará-la permanente. Se fôr o nó de saída, determinou-se a distância mínima e termina o algoritmo, senão volta ao ponto 2.

Para determinar a sequência de nós que forma o caminho com distância mínima, deve-se, retrocedendo a partir do nó de saída, procurar os nós com etiquetas permanentes cuja diferença é igual à distância associada ao arco que os une.

Aplicação do algoritmo de Dijkstra

Slide 53



E_i – vector de etiquetas na iteração i
$E_i[j]$ – etiqueta do nó $j, j \in \{1, \ldots, n\}$ na iteração i

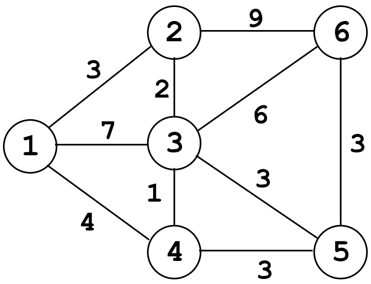
$E_0 = [0^*, \infty, \infty, \infty, \infty, \infty]$
– último nó a ter uma etiqueta definitiva: 1
$E_1[2] = \min(E_0[2], E_0[1] + d_{12}) = \min(\infty, 3) = 3$
$E_1[3] = \min(E_0[3], E_0[1] + d_{13}) = \min(\infty, 7) = 7$
$E_1[4] = \min(E_0[4], E_0[1] + d_{14}) = \min(\infty, 4) = 4$

$E_1 = [0^*, 3^*, 7, 4, \infty, \infty]$
– último nó a ter uma etiqueta definitiva: 2
$E_2[3] = \min(E_1[3], E_1[2] + d_{23}) = \min(7, 5) = 5$
$E_2[6] = \min(E_1[6], E_1[2] + d_{26}) = \min(\infty, 12) = 12$

$E_2 = [0^*, 3^*, 5, 4^*, \infty, 12]$
– último nó a ter uma etiqueta definitiva: 4
$E_3[3] = \min(E_2[3], E_2[4] + d_{43}) = \min(5, 5) = 5$
$E_3[5] = \min(E_2[5], E_2[4] + d_{45}) = \min(\infty, 7) = 7$

Aplicação do algoritmo de Dijkstra – continuação

Slide 54



$E_3 = [0^*, 3^*, 5^*, 4^*, 7, 12]$
– último nó a ter uma etiqueta definitiva: 3
$E_4[5] = \min(E_3[5], E_3[3] + d_{35}) = \min(7, 8) = 7$
$E_4[6] = \min(E_3[6], E_3[3] + d_{36}) = \min(12, 11) = 11$

$E_4 = [0^*, 3^*, 5^*, 4^*, 7^*, 11]$
– último nó a ter uma etiqueta definitiva: 5
$E_5[6] = \min(E_4[6], E_4[5] + d_{56}) = \min(11, 10) = 10$

$E_5 = [0^*, 3^*, 5^*, 4^*, 7^*, 10^*]$
Distância mínima: 10
Caminho mínimo: 1 – 4 – 5 – 6

Algoritmo de Dijkstra na forma tabular

Slide 55

	Nós					
iter	1	2	3	4	5	6
0	0^*	∞	∞	∞	∞	∞
1	0^*	3^*	7	4	∞	∞
2	0^*	3^*	5	4^*	∞	12
3	0^*	3^*	5^*	4^*	7	12
4	0^*	3^*	5^*	4^*	7^*	11
5	0^*	3^*	5^*	4^*	7^*	10^*

Distância mínima: 10
Caminho mínimo: 1 – 4 – 5 – 6

Exemplo de problema de caminho mínimo

O Sr. Ven de Dor, técnico de vendas, vai comprar um carro novo. Este veículo sofrerá uma utilização muito grande (dadas as características da profissão do Sr. Ven de Dor), pelo que, e apesar do o Sr. Ven de Dor se ir reformar daqui a 3 anos (pelo que não precisará mais do carro), poderá ser economicamente mais favorável trocar ainda de carro ao fim de 1 ou 2 anos, em vez de manter este durante os 3 anos. Isto sobretudo porque os custos de utilização e manutenção crescem muito rapidamente com o decorrer dos anos.

Slide 56

O Sr. Ven de Dor sentou-se à sua secretária e calculou o custo total, preço de um carro novo menos o que o stand dá pelo usado, mais os custos de utilização e manutenção (oficina...), de comprar um carro novo no fim do ano i e trocá-lo no fim do ano j.

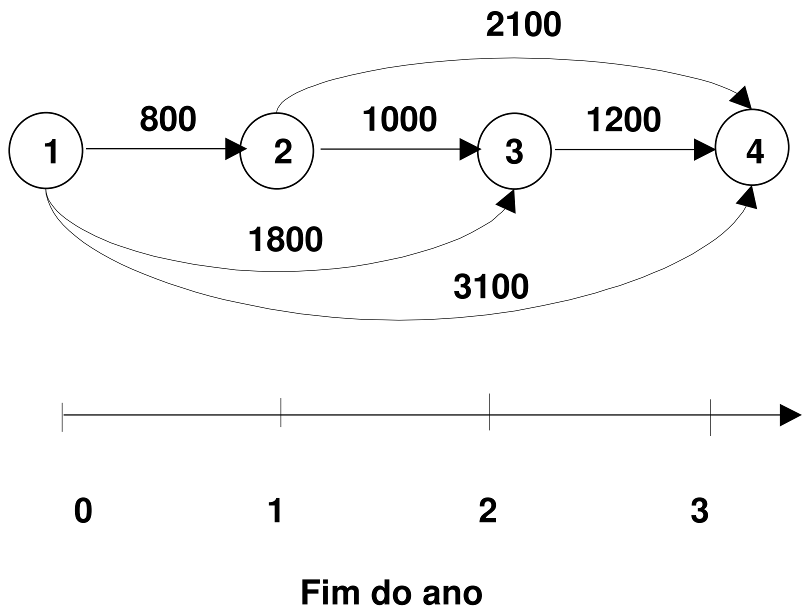
	i		
j	0	1	2
1	800		
2	1800	1000	
3	3100	2100	1200

(em milhares de escudos)

Assim, por exemplo, trocar o carro agora comprado (fim do ano 0) no fim do ano 1 e depois manter o carro comprado no fim do ano 1 até ao fim do ano 3, corresponde a um custo de $800 + 2100 = 2900$.
O problema que o Sr. Ven de Dor tem que resolver é determinar quantas vezes deve trocar de carro (se alguma!) de forma a minimizar a sua despesa total com carros durante estes 3 anos.

Resolução (parcial)

Slide 57



Bibliografia

- Oliveira, José Fernando (1996). *Apontamentos de Investigação Operacional 1.* FEUP.
- Taha, Hamdy A. (1997). *Operations Research, an Introduction.* Prentice Hall.

Slide 58

Árvore de Suporte de Comprimento Mínimo
Minimal Spanning Tree

Slide 59

Árvore de Suporte de Comprimento Mínimo

Slide 60

- Definições (para grafos não orientados):
  - uma árvore é um grafo conexo que não contém ciclos;
  - um grafo diz-se conexo se existir uma cadeia (sequência de ramos) ligando qualquer par de nós entre si.
- Problema:

  Determinar a árvore de comprimento total mínimo que suporte todos os nós da rede (i.e. que ligue todos os nós da rede) ("minimal spanning tree").
- Aplicações:
  - redes de comunicações;
  - redes de distribuição de energia.

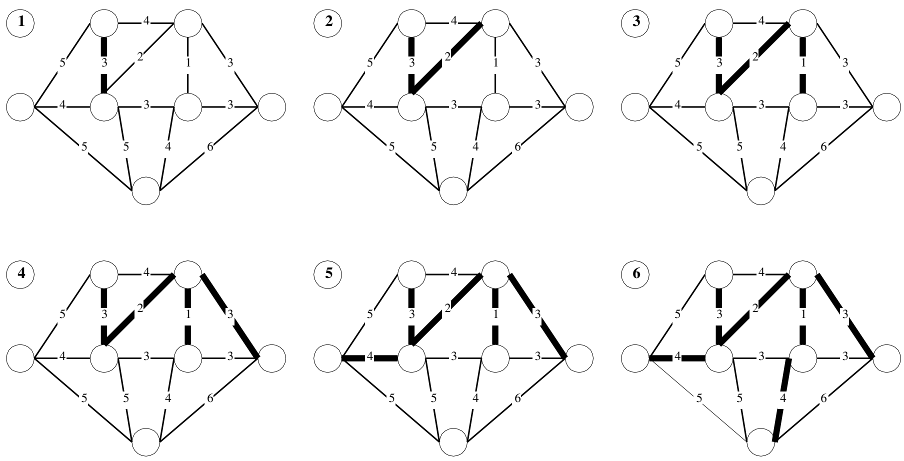
Algoritmo guloso ("Greedy Procedure")

1. Seleccionar um nó arbitrariamente, e ligá-lo ao nó mais próximo;
2. Identificar o nó ainda isolado que esteja mais próximo de um nó já ligado, e ligar estes dois nós;

   Repetir **2.** até que todos os nós estejam ligados entre si.

Slide 61

Algoritmo guloso – Exemplo



Slide 62

Circuitos Eulerianos

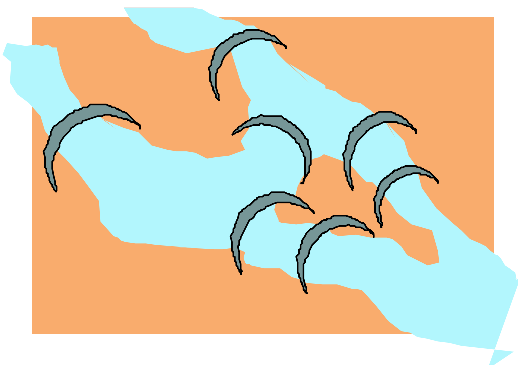
Slide 63

Circuitos Eulerianos – O problema das Pontes de Königsberg

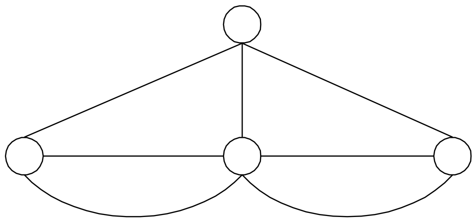
Slide 64

O rio Pregel banha a cidade de Königsberg, na Prússia Oriental, e rodeia a ilha de Kneiphof. A ligar os vários pontos das margens, havia 7 pontes dispostas como se representa na figura. No seu passeio dominical, os habitantes da cidade procuravam voltar ao ponto de partida, passando uma só vez por todas as pontes.

Leonard Euler estudou o problema e demonstrou a sua impossibilidade em 1736.



Circuitos Eulerianos



Slide 65

Definição de Circuito Euleriano: Um Circuito Euleriano, é um caminho finito em que o nó inicial coincide com o nó final e que passa uma única vez por todos os arcos da rede.

- grafo orientado: Circuito
- grafo não orientado: Ciclo

Teorema de Euler: Um grafo admite um circuito euleriano se e só se for conexo e o número de vértices de grau impar for zero. (o grau de um vértice, corresponde ao número de arcos incidente no vértice)

Problema do Carteiro Chinês “Chinese Postman Problem (CPP)”

Problema (Kwan, 1962):

Definir circuitos que se aproximem do Circuito Euleriano ideal (repetindo arcos em número mínimo ou de modo a tornar mínimo o aumento de percurso).

Slide 66

Redes de distribuição linear:

- recolha de lixo numa cidade;
- distribuição domiciliária do correio;
- inspecção periódica de redes de energia, de telefones ...

Circuitos Hamiltonianos

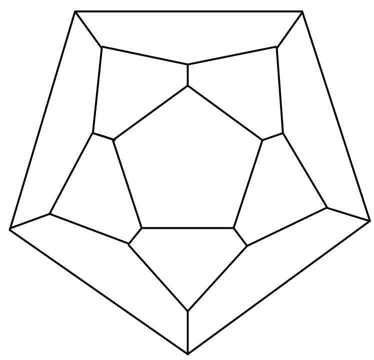
Slide 67

Slide 68

Circuitos Hamiltonianos

Definição de Circuito Hamiltoniano: Um circuito diz-se Hamiltoniano, se passar uma e uma só vez por todos os vértices de uma rede.

A designação provém de um passatempo imaginado pelo islandês Hamilton (1859), no qual se procurava encontrar um caminho que percorresse 20 cidades de todo o mundo, representadas pelos vértices de um dodecaedro de madeira, voltando ao ponto inicial.



Problema do Caixeiro Viajante "Travelling Salesman Problem (TSP)"

Problema:

Pretende-se encontrar o caminho mais curto para um caixeiro viajante que sai de uma cidade, visita n outras cidades e volta àquela de onde partiu, sem repetir nenhuma das cidades visitadas.

Slide 69

Trata-se da pesquisa do circuito hamiltoniano mais curto num grafo de $n+1$ vértices.

O número de soluções possíveis é $n!$.

- Algoritmos conducentes à **solução óptima**, baseiam-se por exemplo em "Branch and Bound" (Little, 1963) e são pouco eficientes.
- Algoritmos que levam a **soluções quase óptimas**, baseiam-se em regras heurísticas (Cicero-Ruggiero, 1972) e são muito eficientes.

Problema do Caixeiro Viajante (TSP) — Modelo

Variáveis de decisão

$$x_{ij} = \begin{cases} 1 & \text{se a cidade } j \text{ seguir imediatamente a cidade } i \\ 0 & \text{se não} \end{cases}$$

Slide 70

Função objectivo

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij}$$

Restrições

$$\begin{aligned} \textstyle\sum_{i=1}^{n} \quad x_{ij} &= 1 \quad \forall_j \\ \textstyle\sum_{j=1}^{n} \quad x_{ij} &= 1 \quad \forall_i \\ \textstyle\sum_{i \in S_t} \sum_{i \in \bar{S}_t} \quad x_{ij} &\geq 1 \quad \forall_{S_t \subset V(cidades)} \end{aligned}$$

e ainda ... restrições que eliminam Sub-tours ...

Bibliografia

- Sousa, Jorge Pinho (1985). *Apontamentos para as aulas práticas de Investigação Operacional.* FEUP.

Slide 71