

1 Relax4

1.1 Formato do input

n
m
org dst custo cap (m vezes)
vert (n vezes)

- **n**: número de vértices
- **m**: número de arcos do grafo
- **org**: vértice de origem do arco
- **dst**: vértice de destino
- **custo**: custo de transporte
- **cap**: capacidade do arco
- **vert**: oferta/procura no vértice, + e - respetivamente

2 Transportes: Introdução

2.1 Modelo geral

- Dado um grafo $G = (V, A)$, pretende-se:

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\ \text{sujeito a} \quad & - \sum_{(i,j) \in A} x_{ij} + \sum_{(j,i) \in A} x_{ji} = b_j, \quad \forall j \in V \quad (1) \\ & 0 \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in A \quad (2) \end{aligned}$$

Variáveis de decisão:

- x_{ij} : fluxo de um único tipo de entidades no arco orientado (i,j) ;

Dados:

- c_{ij} : custo unitário de transporte no arco orientado (i,j) ;
- b_j : oferta (valor positivo) ou procura (valor negativo) no vértice j ;
- u_{ij} : capacidade do arco orientado (i,j) .
- Restrições (1) designam-se por *restrições de conservação de fluxo*.
- Restrições (2) designam-se por *restrições de capacidade*.

2.2 Caracterização das soluções básicas

A uma base podemos associar uma árvore (grafo com vértices não orientados) que suporta todos os vértices.

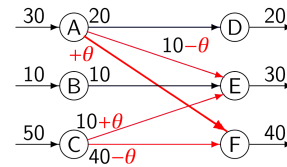
2.2.1 Propriedades da árvore de suporte de um grafo $G = (V, A)$

- é um grafo ligado (existe um caminho entre cada par de vértices)
- sem ciclos
- com $|A| = |V| - 1$ (nº de arcos = nº de vértices - 1)

2.3 Método dos multiplicadores

1. Fixar o valor de qualquer multiplicador em 0
2. Arcos básicos: $c_{ij} = u_i - u_j$
3. Arcos não-básicos: $\delta_{ij} = c_{ij} - (u_i - u_j)$

2.4 Pivô



Qual o valor máximo de θ ? $\theta_{max} = \min\{10, 40\} = 10$

3 Transportes: Grafos Bipartidos

Um grafo $G = (V, A)$ é bipartido se o conjunto de vértices V puder ser dividido em dois conjuntos disjuntos, V_1 e V_2 (i.e., $V_1 \cup V_2 = V, V_1 \cap V_2 = \emptyset$), de tal modo que todos os arcos $(i, j) \in A$ tenham origem num vértice $i \in V_1$ e destino num vértice $j \in V_2$.

3.1 Solução inicial

3.1.1 Método do canto NW

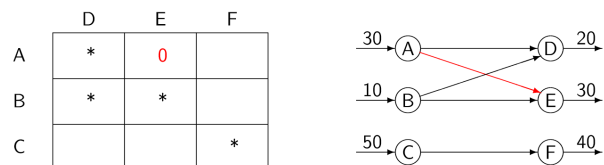
1. Colocar a maior quantidade possível na casa mais a NW \Rightarrow
 - ou a procura de um destino (coluna) é totalmente satisfeita,
 - ou a oferta de uma origem (linha) é totalmente usada,
 - ou ambas.
2. Cortar a linha ou a coluna (ou ambas)
3. Repetir se ainda houver uma casa

3.1.2 Método do canto NW

1. Colocar a maior quantidade possível na casa com custo mínimo \Rightarrow
 - ou a procura de um destino (coluna) é totalmente satisfeita,
 - ou a oferta de uma origem (linha) é totalmente usada,
 - ou ambas.
2. Cortar a linha ou a coluna (ou ambas)
3. Repetir se ainda houver uma casa

3.1.3 Seleção da variável básica com valor 0 (quando faltar uma var. básica)

- Nem todas as variáveis podem ser escolhidas!
- No seguinte exemplo, escolher a variável x_{AE} dá origem a um grafo que não é uma árvore.



- Os arcos associados às variáveis formam um ciclo (i.e., as colunas do modelo de PL são linearmente dependentes, e portanto não formam uma base)

3.2 Pivô

	D	E	F	
A	20 ₃	10- θ_6	+ θ ₅	30
B	2	10 ₅	5	10
C	1	10+ θ_2	40- θ_3	50
	20	30	40	$\theta_{max} = \min\{10, 40\} = 10$

- A variável x_{AF} entra na base e x_{AE} sai da base.

4 Transportes: Redes c/ capacidades

4.1 Caracterização das soluções básicas

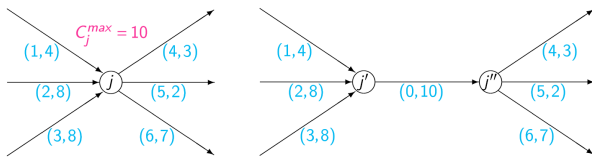
Iguais às referidas no 2.2, mas agora as variáveis no limite superior são também consideradas como não-básicas, para além das iguais a 0.

Uma variável não-básica é atrativa quando:

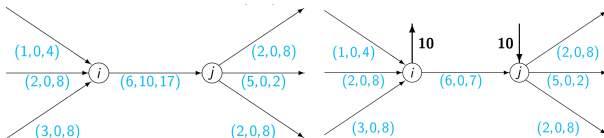
- $x_{ij} = 0$ (variável aumenta de valor) e $\delta_{ij} < 0$
- $x_{ij} = u_{ij}$ (variável decreta de valor) e $\delta_{ij} > 0$

4.2 Transformações

4.2.1 Capacidade num vértice



4.2.2 Limite inferior num arco



5 Programação Inteira: Modelos

5.1 Expressões lógicas

Expressão lógica	Restrição binária
$a \Rightarrow b$	$a \leq b$
$\bar{b} \Rightarrow \bar{a}$	$(1-b) \leq (1-a)$
$\bar{b} \Rightarrow a$	$a \leq b$
$a \Rightarrow \bar{b}$	$a+b \leq 1$
$b \Rightarrow \bar{a}$	$a+b \leq 1$
$a \vee b$ (ou exclusivo)	$a+b = 1$
seleccionar <i>exactamente</i> uma das opções	$a+b+\dots+z = 1$
seleccionar, <i>no máximo</i> , uma das opções	$a+b+\dots+z \leq 1$
$a, b \Rightarrow c$	$a+b-1 \leq c$

6 PI: Planos de corte

Estratégia do método dos planos de corte:

1. determinar a solução ótima do problema de PL
2. cortar partes do domínio em que não há sol. inteiras e reoptimizar, até obter uma sol. inteira (que é ótima!)

Algoritmo de planos de corte:

1. Optimizar *relaxação linear*
2. Enquanto a solução não for inteira:
 - identificar um plano de corte
 - adicionar plano de corte ao conjunto de restrições
 - reoptimizar (usando o método simplex dual)

7 PI: Partição e avaliação

7.1 Partição do domínio

Dado um pai com uma solução fraccionária:

1. seleccionar variável x_j fraccionária
2. criar 2 nós filhos, $x_j \leq \lfloor x_j \rfloor$ e $x_j \geq \lceil x_j \rceil$

7.2 Solução incumbente

É a melhor solução inteira encontrada até um dado passo da pesquisa (x_{SI}) com valor de função objectivo z_{SI} .

7.3 Avaliação do nó (prob. maximização)

7.3.1 [Início] Determinar sol. ótima PL ($x_{PL} \rightarrow z_{PL}$)

- se for inteira, é a melhor sol. inteira no domínio do nó
- se não, pode haver na subárvore uma sol. inteira $z_{PL} \leq z_{SI}$

7.3.2 [Opção 1] Abandonar o nó (podar a subárvore) se:

- o problema for impossível (domínio vazio)
- a solução x_{PL} for inteira (atualizar incumbente se $z_{PL} > z_{SI}$)
- a solução x_{PL} for fraccionária e não puder haver na subárvore uma solução inteira melhor do que a solução incumbente $z_{PL} \leq z_{SI}$

7.3.3 [Opção 2] Fazer partição (explorar a subárvore):

- se a solução x_{PL} for fraccionária e ainda puder haver na subárvore uma solução inteira melhor do que a solução incumbente $z_{PL} > z_{SI}$

7.4 Limites inf. e sup. para o valor do ótimo

$z_I^* \rightarrow$ solução ótima inteira

$$z_I^* \in [LI, LS]$$