

## 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 Caracterização das soluções básicas

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

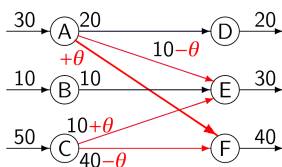
#### 2.1.1 Propriedades da árvore de suporte de um grafo $G = (V, A)$

- é um grafo ligado (existe um caminho entre cada par de vértices)
- sem ciclos
- com  $|A| = |V| - 1$  ( $n^o$  de arcos =  $n^o$  de vértices - 1)

### 2.2 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:  $\delta_{ij} = c_{ij} - (u_i - u_j)$

### 2.3 Pivô



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

## 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

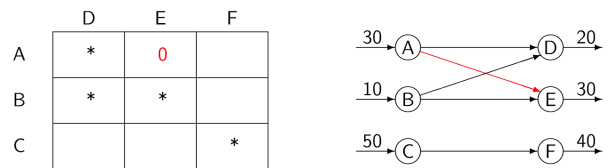
1. Colocar a maior quantidade possível na casa mais a NW  $\Rightarrow$ 
  - ou a procura de um destino (coluna) é totalmente satisfeita,
  - ou a oferta de uma origem (linha) é totalmente usada,
  - ou ambas.
2. Cortar a linha ou a coluna (ou ambas)
3. Repetir se ainda houver uma casa

### 3.1.2 Método do custo mínimo

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

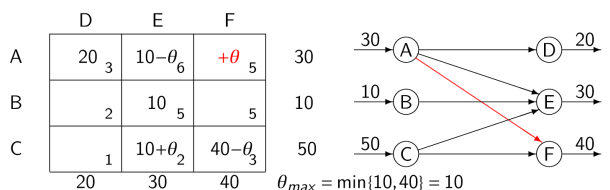
### 3.1.3 Seleção da variável básica com valor 0 (quando faltar uma var. básica)

- 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



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

## 4 Transportes: Redes c/ capacidades

### 4.1 Caracterização das soluções básicas

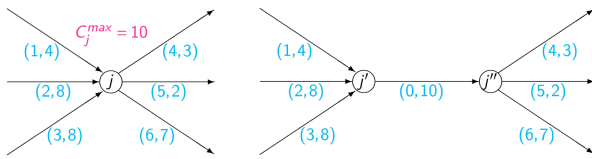
Iguais às referidas no 2.2, mas agora as variáveis no limite superior são também consideradas como não-básicas, para além das iguais a 0.

Uma variável não-básica é atrativa quando:

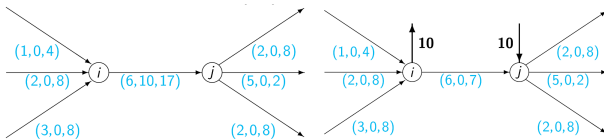
- $x_{ij} = 0$  (variável aumenta de valor) e  $\delta_{ij} < 0$
- $x_{ij} = u_{ij}$  (variável decrementa de valor) e  $\delta_{ij} > 0$

## 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

### 6.1 Algoritmo de planos de corte

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

### 6.2 Plano de corte de Chvátal-Gomory

1. Pegar na restrição da var. básica com a maior mantissa
2. Pegar na parte fracionária da restrição e meter  $\geq$
3. Inverter de modo a ficar com  $\leq$
4. Introduzir var. de folga de modo a ficar com =

#### 6.2.1 Exemplo

1. 
$$x_1 + \frac{1}{4}s_1 + \frac{3}{2}s_2 = \frac{5}{2}$$
2. 
$$\frac{1}{4}s_1 + \frac{1}{2}s_2 \geq \frac{1}{2}$$
3. 
$$-\frac{1}{4}s_1 - \frac{1}{2}s_2 \leq -\frac{1}{2}$$

4.

$$-\frac{1}{4}s_1 - \frac{1}{2}s_2 + s_3 = -\frac{1}{2}$$

## 7 PI: Partição e avaliação

### 7.1 Partição do domínio

Dado um pai com uma solução fracionária:

1. seleccionar variável  $x_j$  fracionária
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 objetivo  $z_{SI}$ .

### 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  $\leq z_{PL}$

#### 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 fracionária 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 fracionária e ainda puder haver na subárvore uma solução inteira melhor do que a solução incumbente  $z_{PL} > z_{SI}$

### 7.4 Limites para o valor do ótimo

$$z_I^* \in [L, L], z_I^* \rightarrow \text{solução ótima inteira}$$

### 7.5 Limite superior

Num problema de maximização o LS é apenas um valor de referência, não está associado a nenhuma solução inteira admissível. O valor do ótimo da relaxação linear ( $z_{RL}$ ) é um limite superior para o valor do ótimo inteiro  $z_I^*$ :

$$z_I^* \leq z_{RL}$$

Para problemas de minimização é o oposto (lim. inferior).

### 7.6 Limite inferior

O valor de qualquer solução inteira admissível é um limite inferior para  $z_I^*$ .

Para problemas de minimização é o oposto (lim. superior).