

# Relax4: guia de utilização

J.M. Valério de Carvalho  
Departamento de Produção e Sistemas  
Escola de Engenharia  
Universidade do Minho

- Este conjunto de slides ilustra a utilização do package Relax4.
- O Relax4 resolve problemas de minimização de custo do fluxos em rede (*Minimum Cost Network Flow Problems*).
- Os arcos da rede são definidos por uma origem, um destino, um custo unitário de transporte e uma capacidade.
- O programa Relax4 foi desenvolvido por Dimitri Bertsekas e Paul Tseng do Laboratory for Information and Decision Systems and the Operations Research Center, MIT, e está acessível em
- <http://www.neos-server.org/neos/solvers/Ino:Relax4/DIMACS.html>

# Conteúdo

- Execução do Programa
- Definição do input
- Exemplo 1
- Exemplo 2
- Situações especiais
- Informações adicionais

# Execução do Programa

# Instalação e comando

- O package Relax4 (relax4.exe) pode ser instalado em qualquer directoria. Sugere-se que seja colocado junto da raiz, por exemplo, na directoria `c:\relax4>`
- criar uma janela de DOS (na janela de comandos do Windows, fazer `cmd`),
- posicionar-se na directoria do Relax4 (e.g., `c:\relax4>`), e
- correr o Relax4:
- `c:\relax4> relax4 <relax4.inp`
- Os resultados são, por defeito, apresentados no écran, ou opcionalmente num ficheiro de output, através de um redireccionamento `>ficheiro`.
- `c:\relax4> relax4 <relax4.inp >relax4.out`
- `c:\relax4> relax4 <relax4.inp >con:` (neste caso o ficheiro é o écran)
- Em alternativa, usar o servidor neos (ver endereço *web* no slide inicial).

# Definição do input

# Ficheiro de input

- **Tipo:**
- O ficheiro de dados pode ser editado com qualquer processador de texto, desde que seja guardado como um ficheiro de texto, apenas com caracteres ASCII.
- No caso do Word, deve ser usada a opção “guardar como” ficheiro do tipo \*.txt.
- Ficheiros do tipo \*.doc não podem ser utilizados, porque contêm um cabeçalho com caracteres invisíveis para o utilizador que o Relax4 interpreta como dados.
- **Nome:**
- Sugere-se o uso de nomes de ficheiro pequenos, com 8 ou menos caracteres.
- No Windows, há opções que escondem a extensão do ficheiro; um ficheiro aparentemente com o nome de ex1.txt pode chamar-se, de facto, ex1.txt.txt
- **Aspectos importantes:**
- Após a última linha com dados, fazer *Enter* (o Fortran interpreta o conjunto CR+LF como delimitador do último campo com dados). Depois disso, pode colocar-se qualquer texto.
- O ficheiro é lido sem formato: em cada linha, é suficiente separar os dados por um espaço.

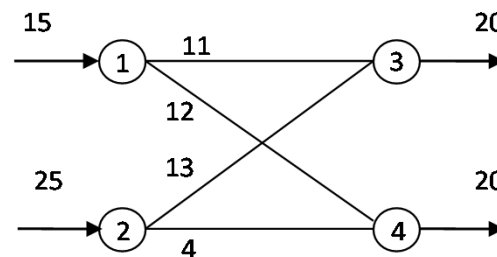
# Formato do ficheiro de input

- **Definição da Dimensão da Rede (cabeçalho)**
- Nas duas primeiras linhas:
  - $n$
  - $m$
- sendo  $n$  o número de vértices e  $m$  o número de arcos do grafo.
- **Definição dos Arcos da Rede**
- Seguem-se  $m$  linhas com uma listagem dos arcos, no seguinte formato:
  - $org \ dst \ custo \ cap$
  - sendo  $org$  o vértice de origem do arco,  $dst$  o vértice de destino,  $custo$  o respectivo custo unitário de transporte e  $cap$  a capacidade do arco (limite superior para o fluxo que nele pode ser transportado).
- **Definição das Ofertas e das Procura em cada Vértice da Rede**
- Nas restantes  $n$  linhas, definem-se as ofertas e as procura em cada vértice do grafo:
  - - uma quantidade *positiva* se existir uma *oferta* no vértice,
  - - uma quantidade *negativa* se existir uma *procura* no vértice, e
  - - *zero*, caso contrário.



# Exemplo 0

- Considere o problema com duas origens e dois destinos, apresentado na Figura, em que as ofertas nos vértices 1 e 2 são iguais a 15 e 25, respectivamente, e os consumos nos vértices 3 e 4 são iguais a 20 e 20, respectivamente:



- Os custos unitários dos arcos são iguais a:  $c_{13}=11$ ,  $c_{14}=12$ ,  $c_{23}=13$  e  $c_{24}=4$ .
- A capacidade de todos os arcos é igual a 1000 (virtualmente infinita).
- Ficheiro de input:

```
4
4
1 3 11 1000
1 4 12 1000
2 3 13 1000
2 4 4 1000
15
25
-20
-20
```

# Notas

- para o problema ser válido, a soma das ofertas deve ser igual à soma das procuras.
- os dados de entrada devem ser números inteiros.

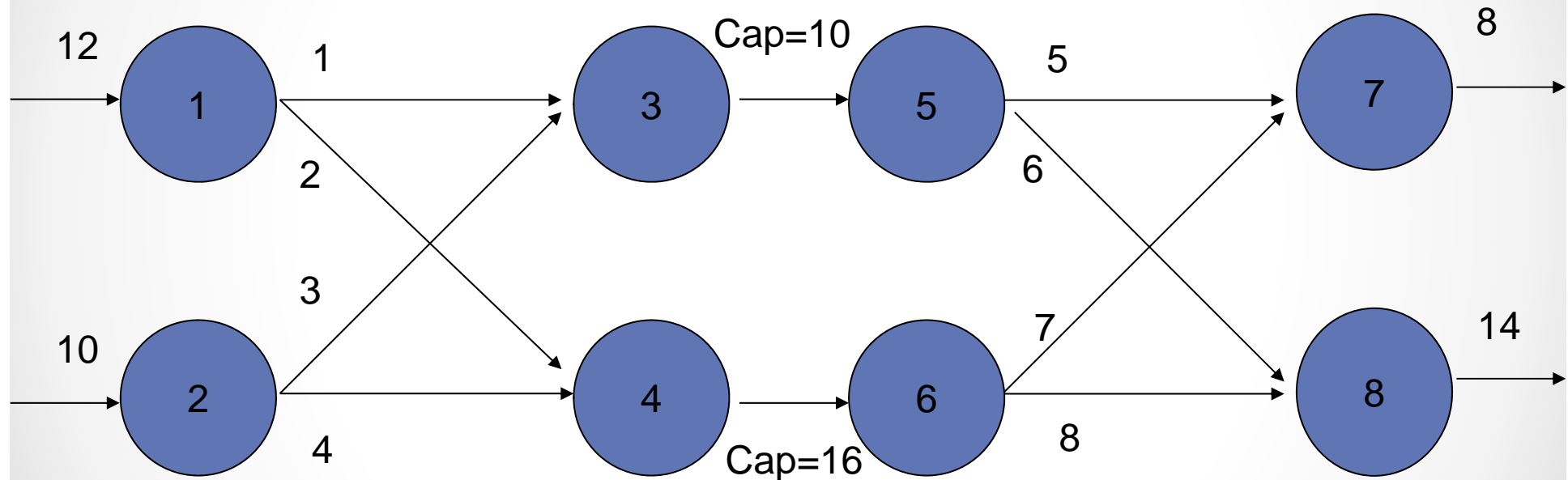
# Nota: numeração dos vértices

- Os vértices são identificados por números naturais a partir de 1.
- Pode haver vértices do grafo sem nenhuma ligação ao resto do grafo.
- O modelo a seguir apresentado é semelhante ao do Exemplo 1, mas o vértice 4 é designado por vértice 5.
- São declarados 7 vértices. De notar que os vértices 4, 6 e 7 não têm qualquer ligação ao resto do grafo.

7  
4  
1 3 11 1000  
1 5 12 1000  
2 3 13 1000  
2 5 4 1000  
15  
25  
-20  
0  
-20  
0  
0

# Exemplo 1

# Problema de Transportes com Armazéns Inter médios



Os arcos (3,5) e (4,6) têm capacidades (limites superiores) de 10 e 16, respectivamente, para modelar as capacidades dos armazéns.

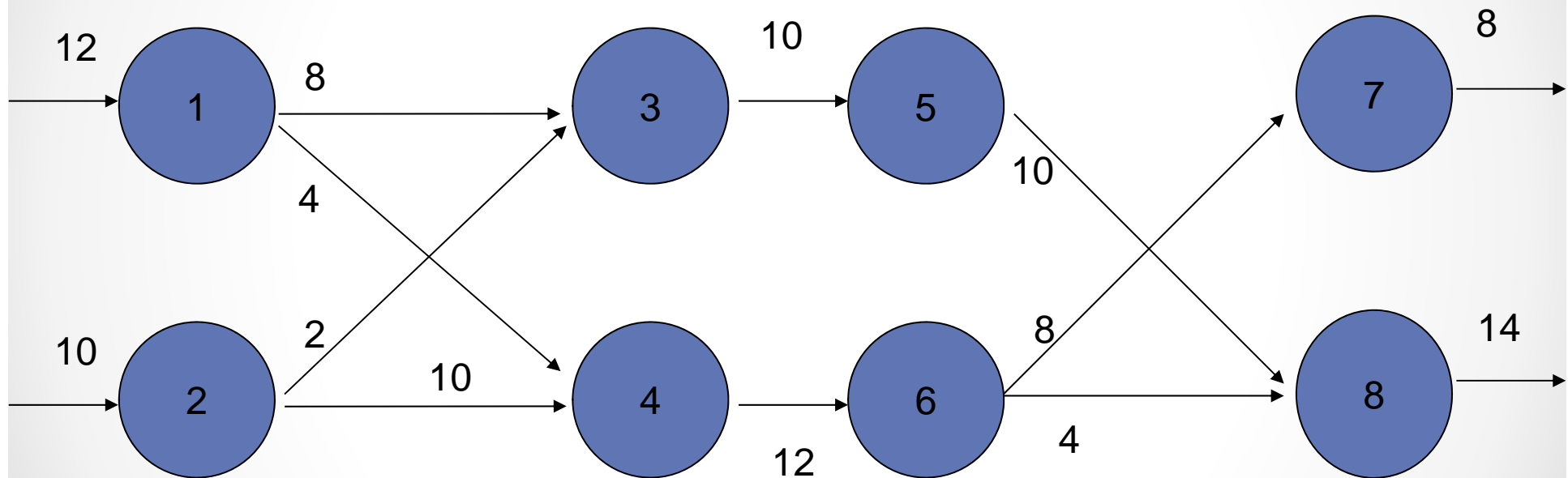
# Ficheiro de input

```
8
10
1 3 1 1000
1 4 2 1000
2 3 3 1000
2 4 4 1000
3 5 0 10
4 6 0 16
5 7 5 1000
5 8 6 1000
6 7 7 1000
6 8 8 1000
12
10
0
0
0
0
-8
-14
```

## Parte do output do Relax

- 1 3 8.
- 1 4 4.
- 2 3 2.
- 2 4 8.
- 3 5 10.
- 4 6 12.
- 5 8 10.
- 6 7 8.
- 6 8 4.
- OPTIMAL COST = 202.

## Solução óptima (quantidades transportadas)

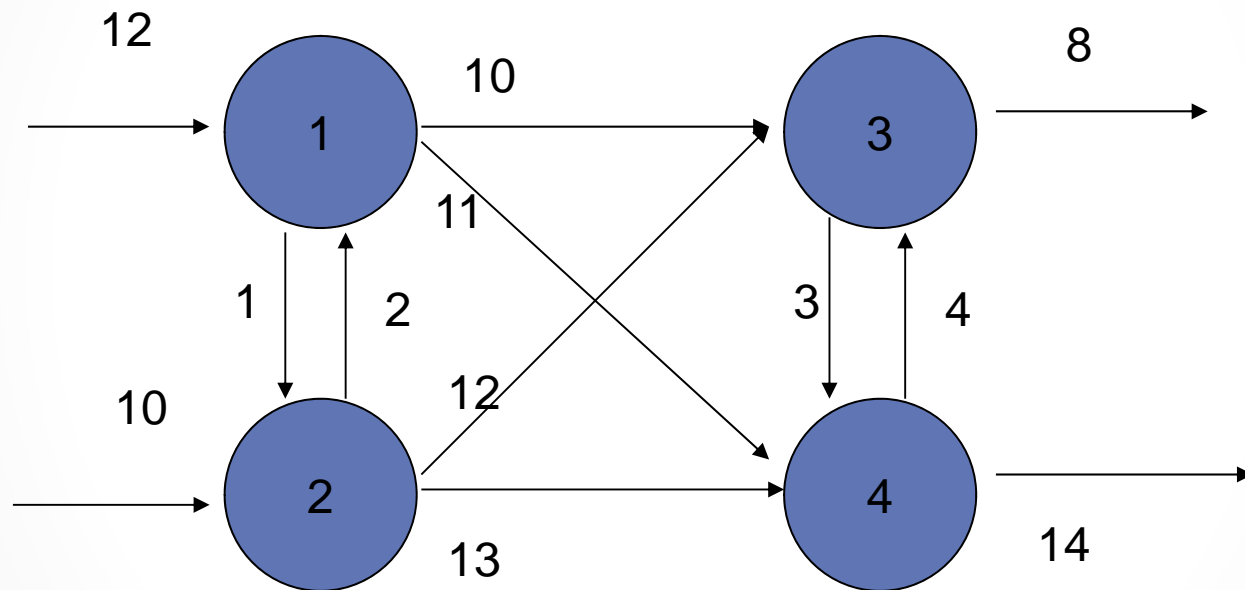


Custo óptimo = 202



# Exemplo 2

# Problema de Transportes com Transbordo



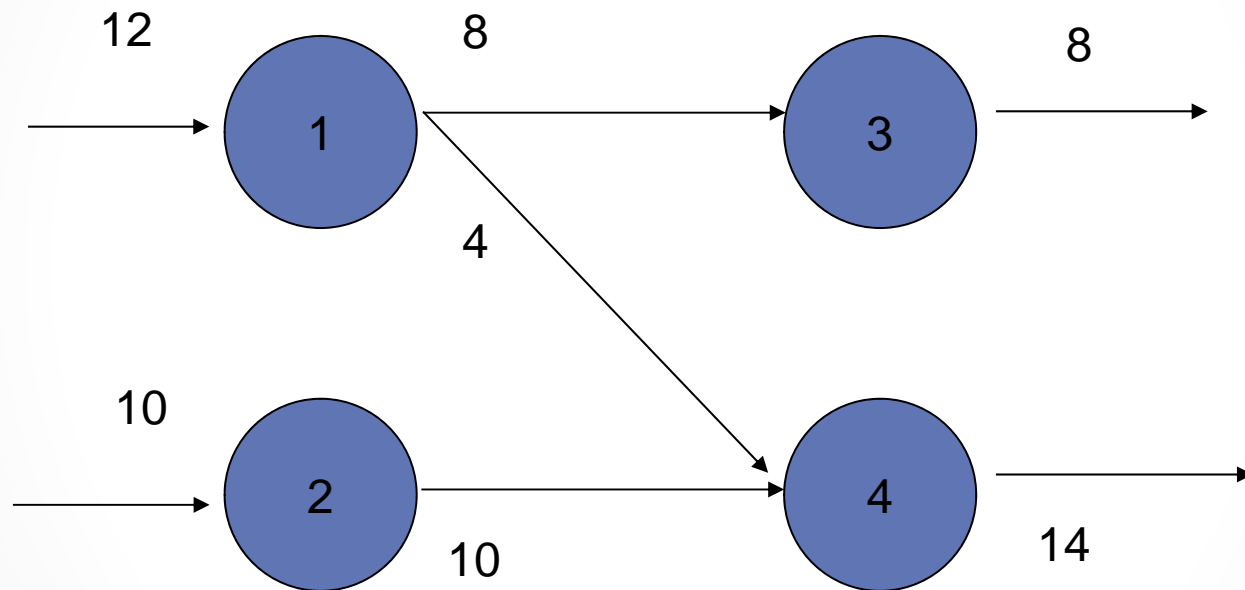
# Ficheiro de input

```
4
8
1 3 10 1000
1 4 11 1000
1 2 1 1000
2 1 2 1000
2 3 12 1000
2 4 13 1000
3 4 3 1000
4 3 4 1000
12
10
-8
-14
```

## Parte do output do Relax

- 1 3 8
- 1 4 4
- 2 4 10
- OPTIMAL COST = 254.00

## Solução óptima (quantidades transportadas)



Custo óptimo = 254

# Situações especiais

# Múltiplos arcos

- Pode haver um conjunto de múltiplos arcos entre a mesma origem e destino (e.g., 4 arcos entre os vértices 2 e 4):
  - a primeira unidade pode fluir a um custo unitário de 1,
  - a segunda unidade pode fluir a um custo unitário de 2,
  - a terceira e a quarta unidades podem fluir a um custo unitário de 3, e
  - as restantes unidades (em número virtualmente infinito) podem fluir a um custo unitário de 4.

4

7

1 3 11 1000

1 4 12 1000

2 3 13 1000

2 4 1 1

2 4 2 1

2 4 3 2

2 4 4 1000

15

25

-20

-20

- Um dos arcos (do conjunto) com maior custo só pode ter fluxo se os arcos de custo inferior estiverem *saturados* (com um fluxo igual à capacidade do arco). Isto serve para modelar custos de transporte convexos.
- Para representar custos côncavos, é necessário usar modelos de programação linear com variáveis binárias.

# Informações adicionais

- Informações adicionais sobre este programa, em particular sobre a estratégia do algoritmo, as estruturas de dados utilizadas e o desempenho computacional, podem ser obtidas no artigo:
- Dimitri Bertsekas e Paul Tseng, “The relax codes for linear minimum cost network flow problems”, Annals of Operations Research 13, 125-188, 1988.



# Fim