# Programação Linear Inteira
# PLI

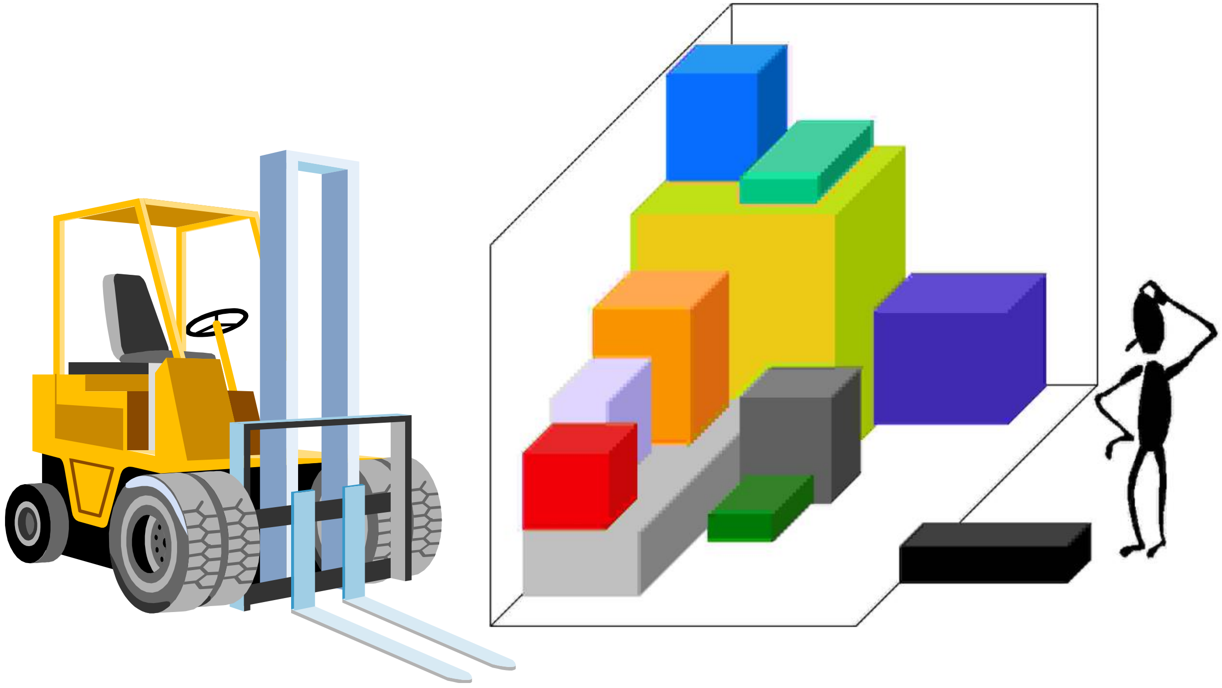
Prof. Ademir A. Constantino
Departamento de Informática
Universidade Estadual de Maringá
www.din.uem.br/~ademir

# Definição

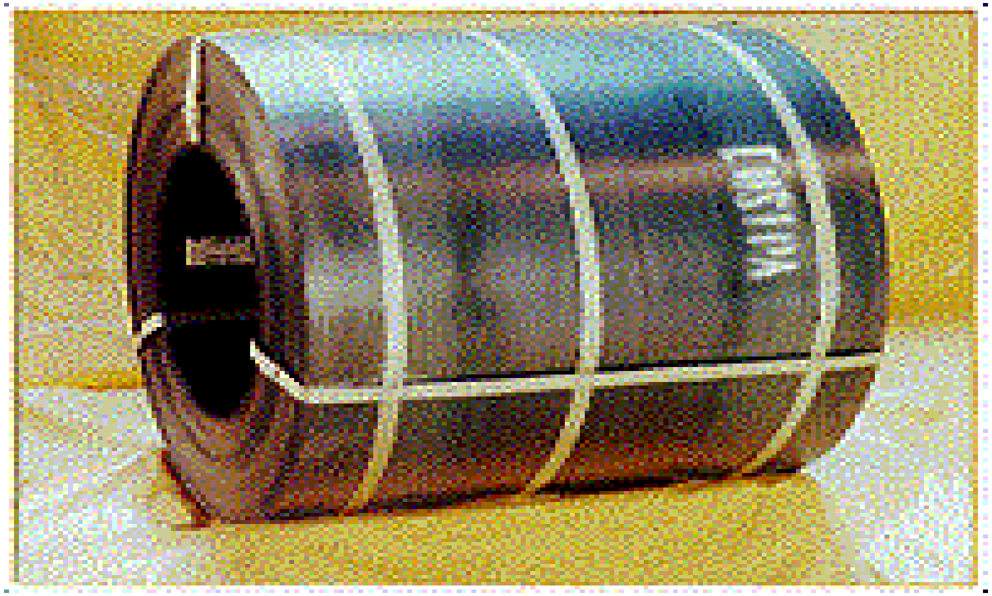
- Um programa de PLI é um problema de PL em que as variáveis de decisão assumem apenas valores inteiros.

- Casos especiais quando as variáveis assumem valores inteiros como zero e um, então o problema é chamado de Problema de Programação Linear Inteira Binário.
- Veremos alguns exemplos.

# Desafio: Problema de Corte e Empacotamento



# INDÚSTRIA SIDERÚRGICA

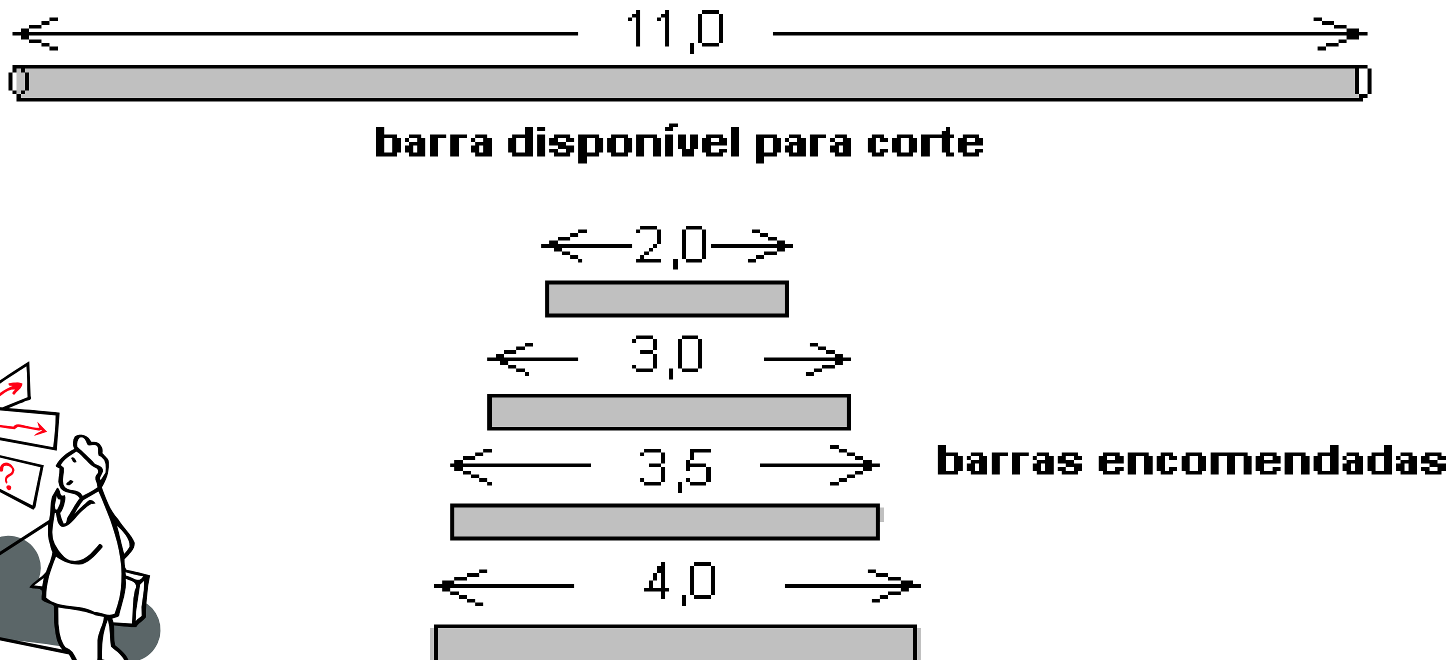


# INDÚSTRIA DE PAPEL

# Problema de Corte e Empacotamento

- uma empresa tem uma demanda de: 2800 barras de 2 metros, 1000 barras de 3 metros, 2000 barras de 3,5 metros e 1500 barras de 4 metros

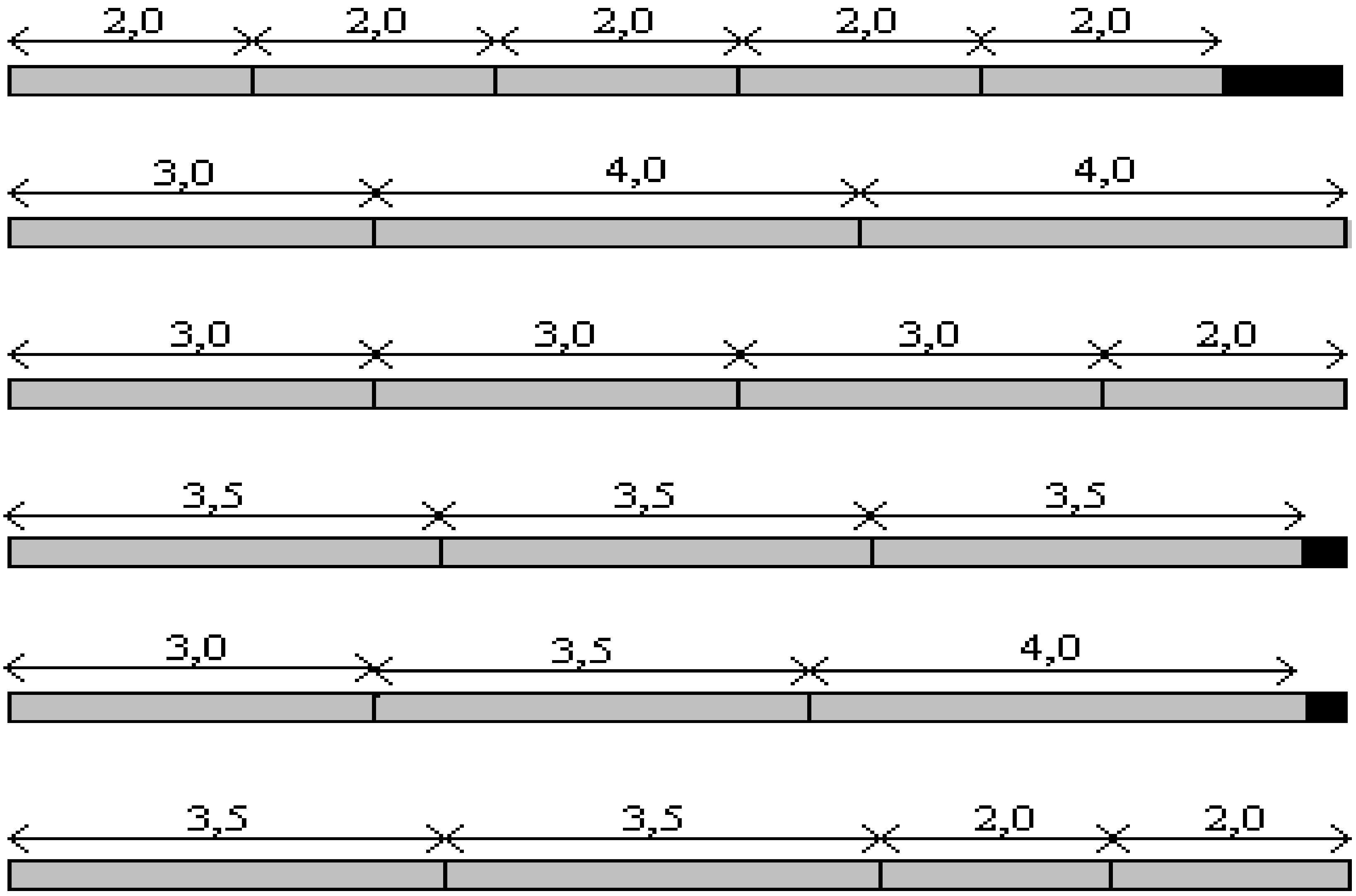
- fornecedor vende apenas barras de 11 metros





- Como atender as encomendas com o mínimo de perda e de barras?

# Padrões de Corte permitidos



# Desafio

- Formule um modelo de PLI que resolva este problema.

# Problema de Corte e Empacotamento

## • Modelagem Matemática

*Minimizar* $Z = 1{,}1x_1 + 1x_2 + 1x_3 + 1{,}05x_4 + 1{,}05x_5 + 1x_6$

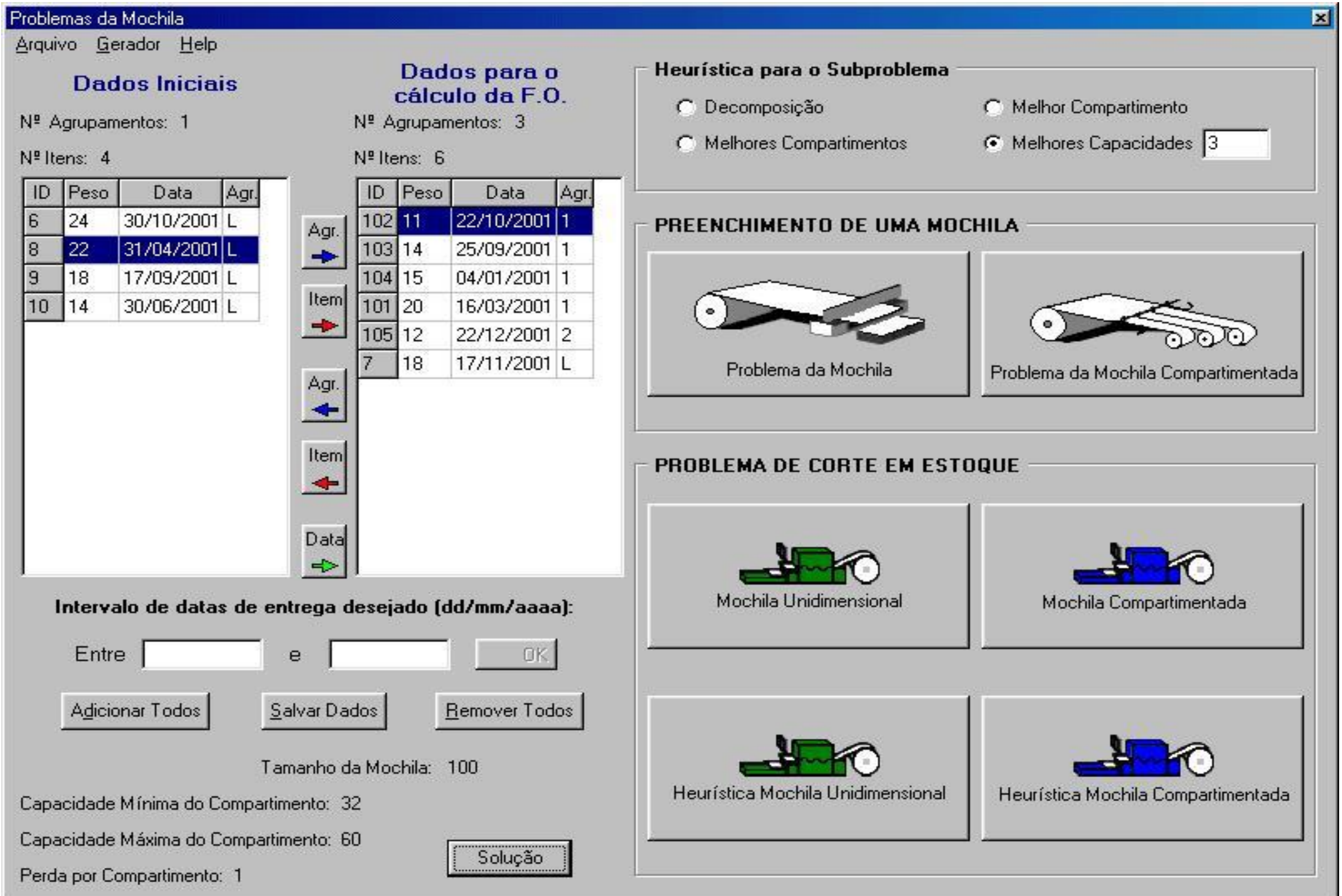
*sujeito a:*

$$\begin{bmatrix}5\\0\\0\\0\end{bmatrix}x_1+\begin{bmatrix}0\\1\\0\\2\end{bmatrix}x_2+\begin{bmatrix}1\\3\\0\\0\end{bmatrix}x_3+\begin{bmatrix}0\\0\\3\\0\end{bmatrix}x_4+\begin{bmatrix}0\\1\\1\\1\end{bmatrix}x_5+\begin{bmatrix}2\\0\\2\\0\end{bmatrix}x_6 \geq \begin{bmatrix}2800\\1000\\2000\\1500\end{bmatrix}$$

$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$ e inteiros.

Onde: $x_i$ é o número de vezes que o padrão de corte *i* é utilizado

# Problema de Corte de Bobinas de Aço



# Problemas Clássicos de PLI

- Problema da Mochila
- Problema de Cobertura de Conjunto
- Problema do Caixeiro Viajante

# Problema da Mochila 0 ou 1

- Dados n objetos que pode-se armazenar em uma mochila, onde cada objeto j (j=1,..., n) tem um peso $w_j$ e um valor de utilidade $v_j$.
- Cada objeto pode ser colocado no máximo uma vez na mochila (não pode ter objetos repetidos na mochila)
- Quais objetos escolher de tal modo que o peso total não seja maior que W (capacidade da mochila) e maximize o valor de utilidade dos objetos incluídos na mochila?

- Seja a variável:

$$x_j = \begin{cases} 1 \text{ se o objeto } j \text{ é incluído na mochila} \\ 0 \text{ caso contrário} \end{cases}$$

- Formulação:

$$max\ z = \sum_{j=1}^{n} v_j x_j$$

*sujeito a :*

$$\sum_{j=1}^{n} w_j x_j \leq W$$

$$x_j \in \{0,1\} \ (j = 1, \ldots, n)$$

# Problema da Mochila Inteiro

- Dados n objetos que pode-se armazenar em uma mochila, onde cada objeto j (j=1,..., n) tem um peso $w_j$ e um valor de utilidade $v_j$.
- Cada objeto pode ser colocado quanta vezes for necessário na mochila (pode ter objetos repetidos na mochila)
- Quais objetos escolher de tal modo que o peso total não seja maior que W (capacidade da mochila) e maximize o valor de utilidade dos objetos incluídos na mochila?

- Seja a variável: **$x_j$ quantidade de objetos j na mochila**

- Formulação:

$$max\ z = \sum_{j=1}^{n} v_j x_j$$

$$sujeito\ a:$$

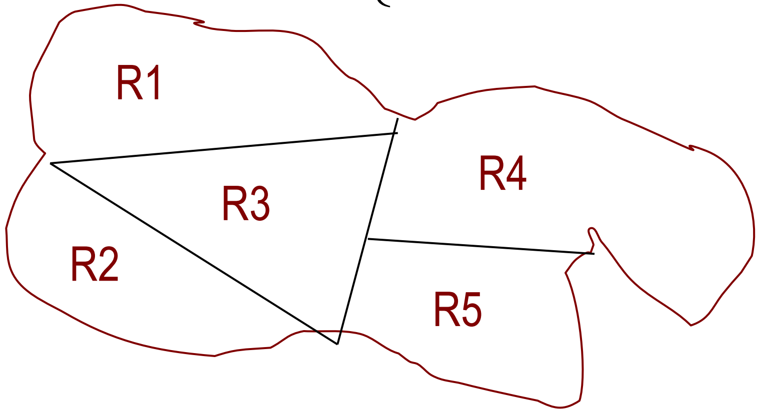
$$\sum_{j=1}^{n} w_j x_j \leq W$$

$$x_j \in Z^+ \ (j = 1, \ldots, n)$$

# Problema de Recobrimento de Conjuntos (Set Covering)

- Considere uma cidade dividida em m=5 regiões;
- Cada região requer o uso de uma facilidade: por exemplo, corpo de bombeiros, banco, hospitais, etc.
- Existem $n$=6 locais candidatos para instalação cada um com um custo $c_j$ ($j$=1,…,$n$);
- Seja $d_{ij}$ a distância entre a região $i$ e o local $j$;
- Seja $D$ a distância máxima entre a região $i$ e o local $j$ para que uma facilidade possa atendê-la;
- Seja $S_j$ = {regiões $i$ tal que $d_{ij} \leq D$} ($j$=1,…, $n$);
- Como escolher os locais de instalação das facilidade de forma que todas as regiões sejam atendidas e minimize o custo?

- Sugestão de variável: $$x_j = \begin{cases} 1 \text{ se a facilidade for construída na localidade } j; \\ 0 \text{ caso contrário.} \end{cases}$$




$S_1$={1, 2}

$S_2$={1, 3, 5}

$S_3$={2, 4, 5}

$S_4$={3}

$S_5$={1}

$S_6$={4, 5}

# Problema de Recobrimento de Conjuntos (Set Covering)

- Formulação:

$$min\ z = \sum_{j=1}^{6} c_j x_j$$

$$sujeito\ a:$$

$$\sum_{j|i \in S_j}^{n} x_j \geq 1 \qquad i = 1,\ldots,5$$

$$x_j \in \{0,1\} \qquad j = 1,\ldots,6$$

-Ou ainda: Sujeito a:

$$\begin{array}{llllllll}
x_1 + & x_2 & & & + x_5 & & \geq 1 \\
x_1 & & + x_3 & & & & \geq 1 \\
& x_2 & & + x_4 & & & \geq 1 \\
& & x_3 & & & + x_6 & \geq 1 \\
& x_2 & + x_3 & & & + x_6 & \geq 1
\end{array}$$

# Exemplo: Problema do Caixeiro Viajante

**min** $\left(\sum_{i}^{n}\sum_{j}^{n} d_{ij} \cdot X_{ij}\right) \rightarrow$minimizar o percurso total

**suj. a:**

(1) cada uma das cidades é visitada uma e só uma vez, ou seja, cada vértice é entrado uma só vez e saído uma só vez:

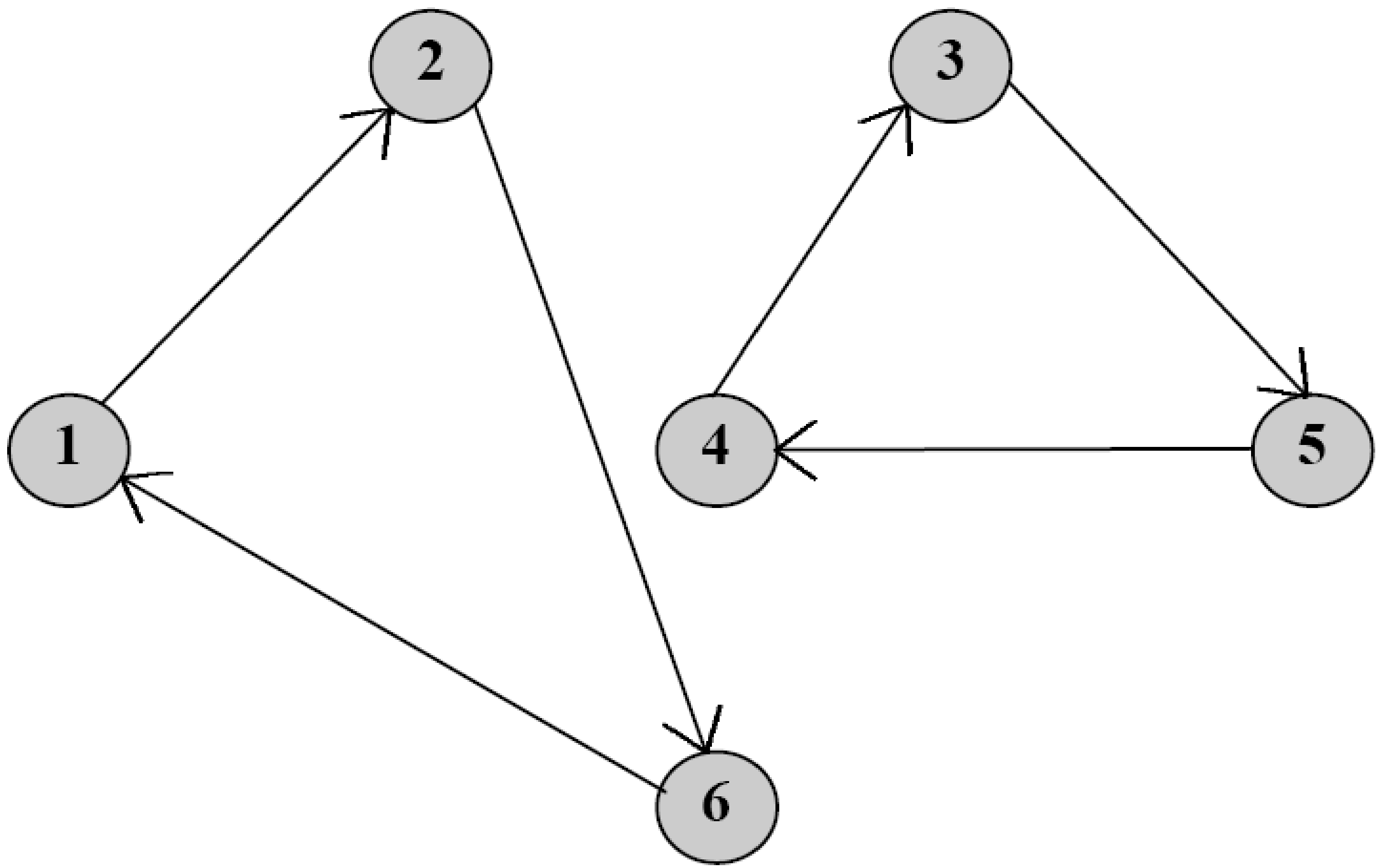
$$\sum_{j}^{n} X_{ij} = 1, \quad \forall_{\mathbf{i:\ i=1,..,n}}$$

$$\sum_{i}^{n} X_{ij} = 1, \quad \forall_{\mathbf{j:\ j=1,..,n}}$$

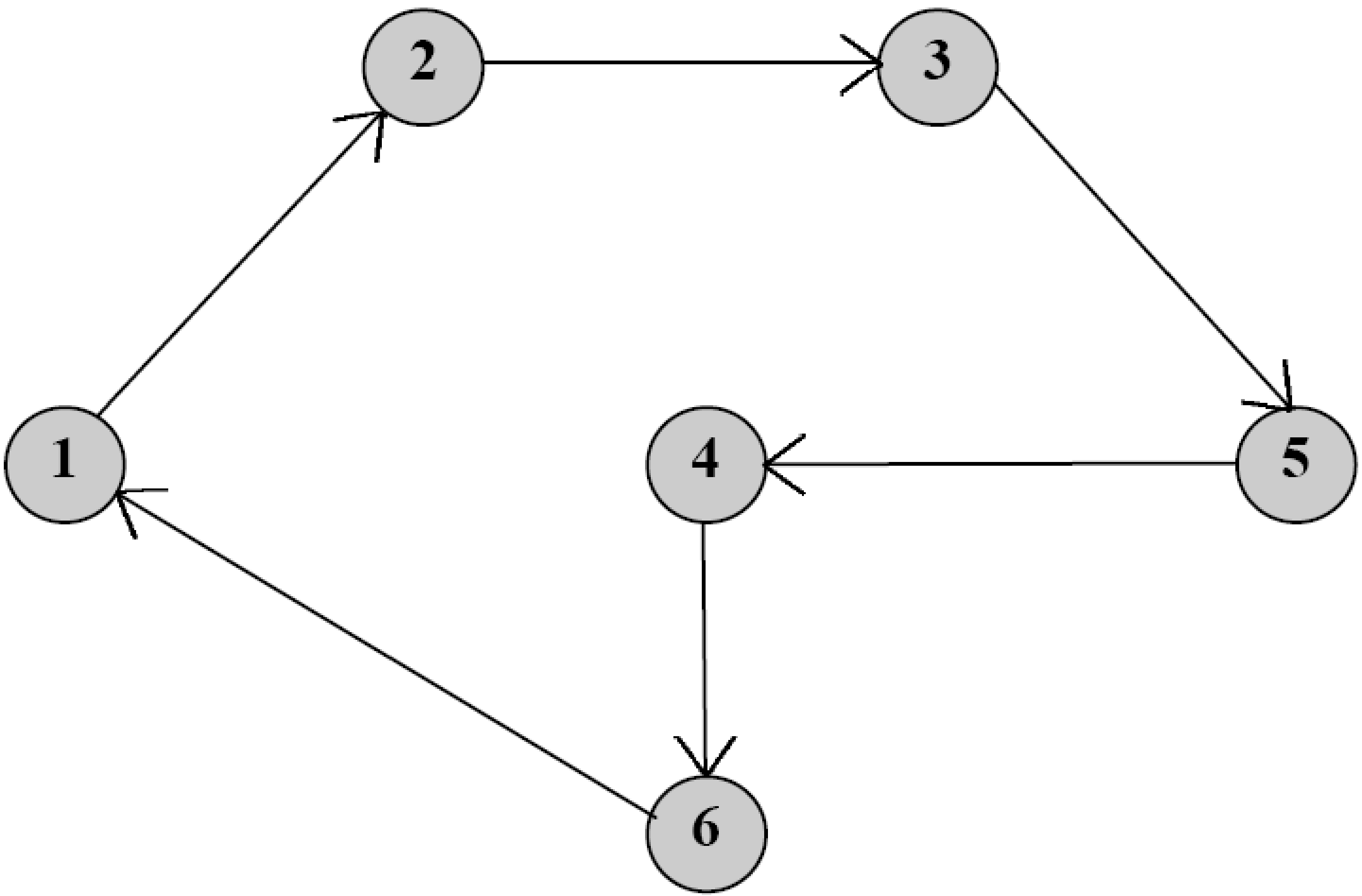
(2) entre dois quaisquer subconjuntos complementares de cidades ( S e $\overline{\text{S}}$ ) há pelo menos um arco de ligação:

$$\sum_{i \in S} \sum_{j \in \overline{S}} X_{ij} \geq 1, \quad \forall_{S \subset \text{ conjunto total das cidades a visitar}}$$

# Restrição 1 não garante a solução



Restrição 2 não permite a formação de sub-circuitos disjuntos.



# Exercício de Formulação

- Uma pessoa é obrigada pelo seu médico a fazer uma dieta que forneça, diariamente, pelo menos a quantidade de vitaminas A, B, C e D especificada na tabela abaixo. A dieta poderá incluir leite, arroz, feijão e carne, que contêm a quantidade de vitaminas, em miligramas por litro ou quilo mostrada na tabela.

| Vitaminas | A | B | C | D | Preço |
|---|---|---|---|---|---|
| leite | 10 | 8 | 15 | 20 | 1,00 |
| arroz | 5 | 7 | 3 | 2 | 0,80 |
| feijão | 9 | 6 | 4 | 3 | 1,20 |
| carne | 10 | 6 | 7 | 9 | 6,00 |
| Quant. Mín. | 80 | 70 | 100 | 60 | |

Formule um modelo de PL que determine o consumo diário de cada um dos alimentos, de tal maneira que a dieta satisfaça à prescrição médica pelo menor custo possível. Porém, o médico recomendou que o a proteína do leite não deva ser misturada com a proteína da carne.

*Dica: leve em consideração o valor máximo que a variável de decisão pode assumir. Pode ser pelas restrições do problema, ou pela valor máximo da faixa dos  números reais (float) no computador.*

# Execícios

- Fluxo Máximo em Rede
- Caminho Mínimo em Gráfos
- Problema de Designação
- Problema de Designação com Gargalo
- Problema de Sequenciamento em Processadores Paralelos e Idênticos.