Programação Linear Inteira
PLI

Prof. Igor Natal
Departamento de Informática
Universidade Estadual de Maringá

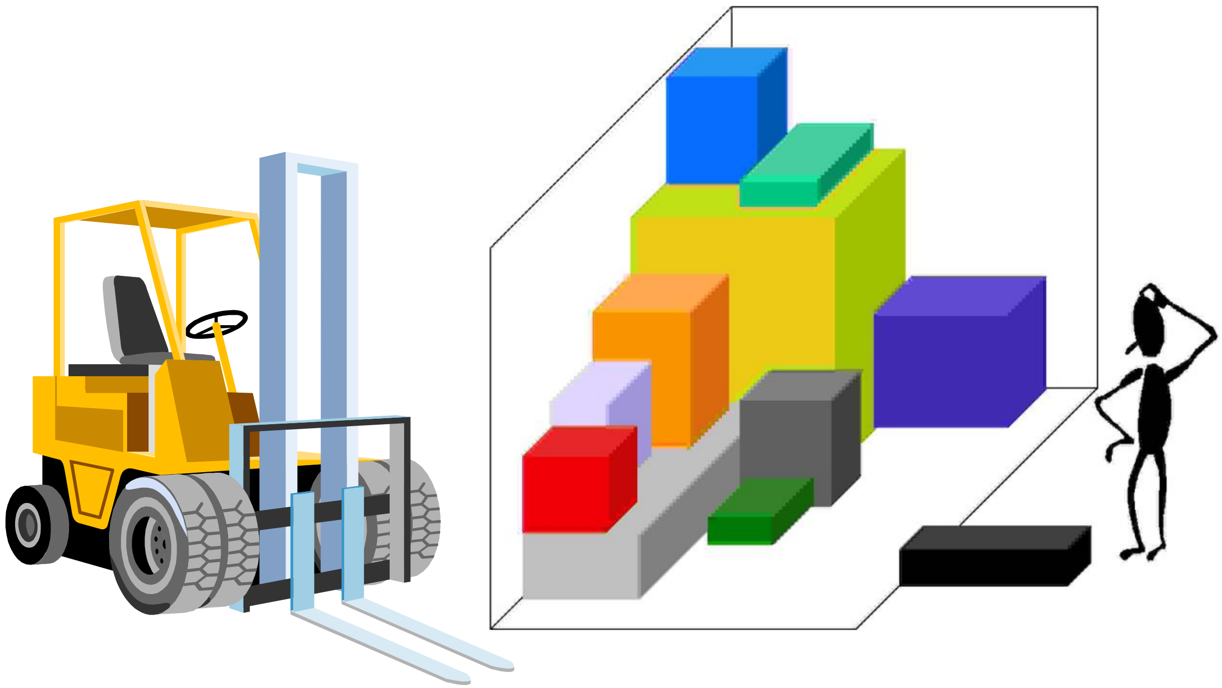
Com base no material do prof. Ademir Constantino

Definição

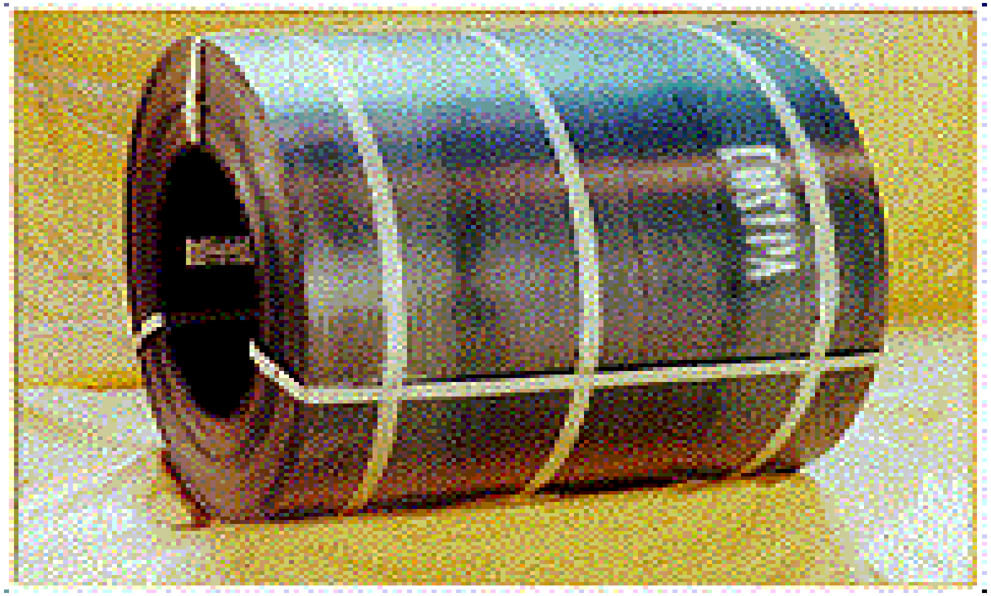
- Um programa de PLI é um problema de PL em que as variáveis de decisão assumem apenas valores inteiros.

- 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

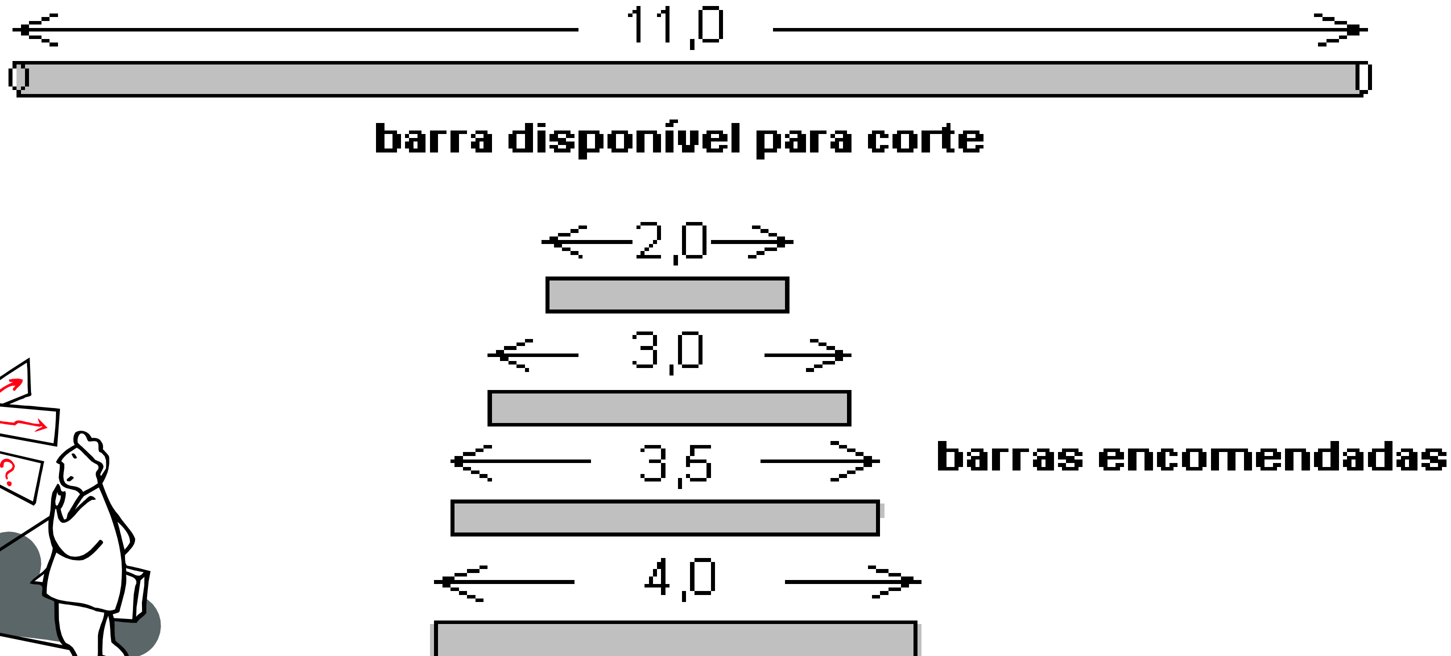


INDÚSTRIA DE PAPEL

Problema de Corte e Empacotamento

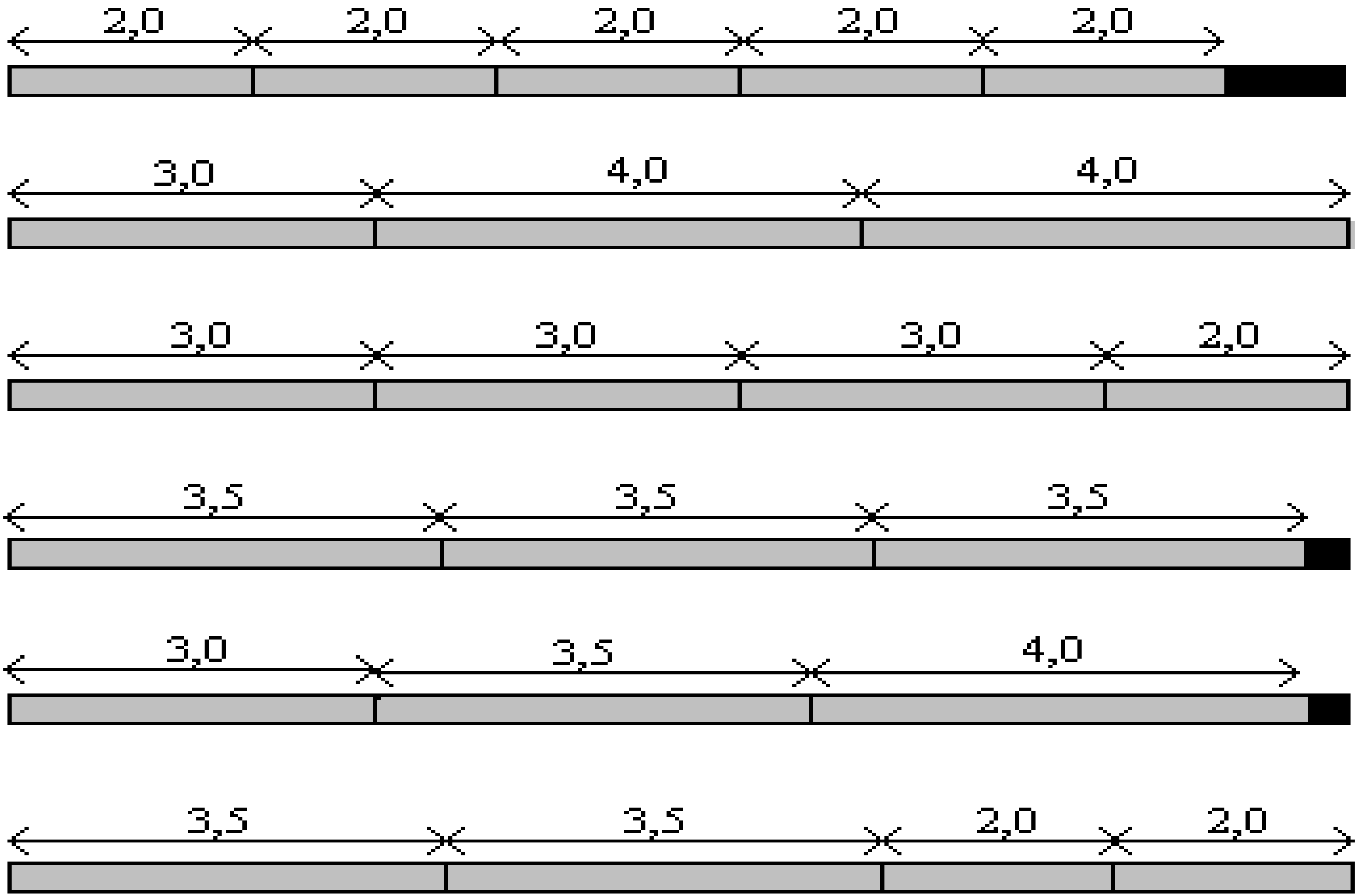
- uma empresa tem uma demanda de: 2800 barras de 2 metros,
  1000 barras de 3 metros,
  2000 barras de 3,5 metros e
  1500 barras de 4 metros

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

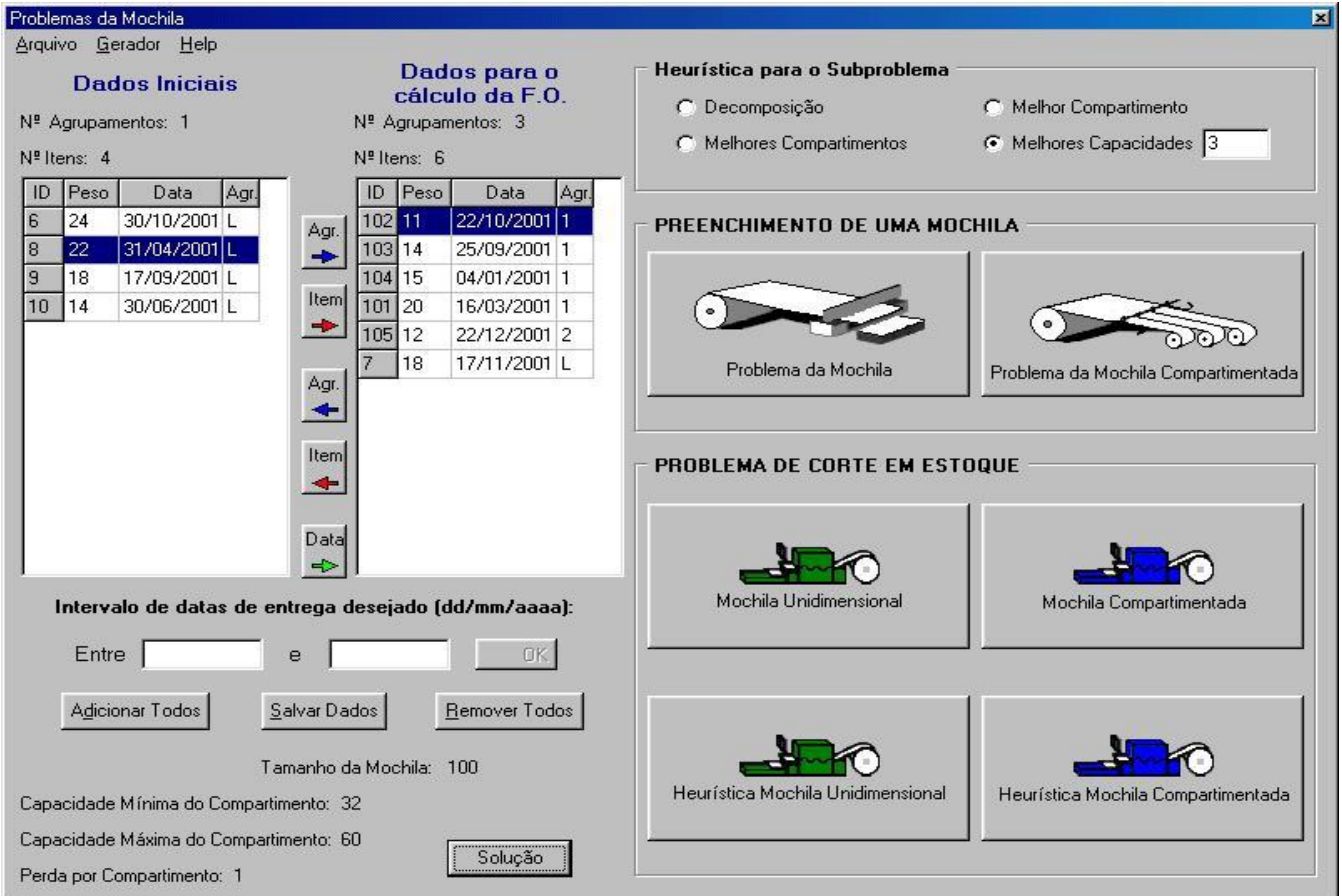
sujeito a:

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

$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$ 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 :

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

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

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: $\mathbf{x_j}$ **quantidade de objetos j na mochila**

- Formulação:

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

$$sujeito\ a:$$

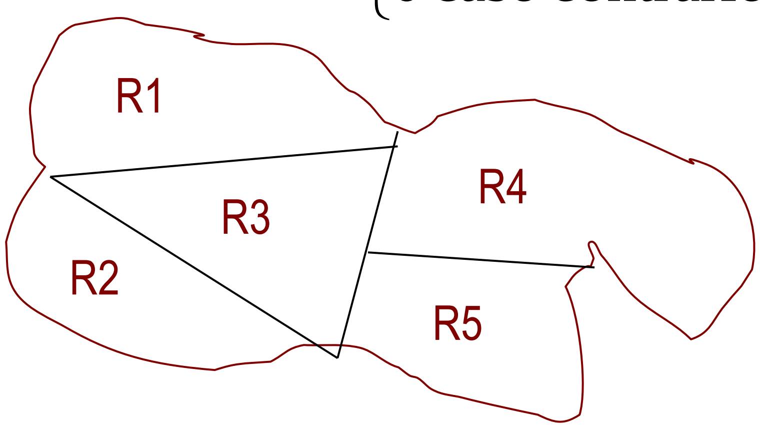
$$\sum_{j=1}^{n} w_j x_j \leq W$$

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

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:

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

$$\text{sujeito a :}$$

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

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

-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 $\left(\sum_{i}^{n} \sum_{j}^{n} d_{ij} \cdot X_{ij} \right)$ →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:

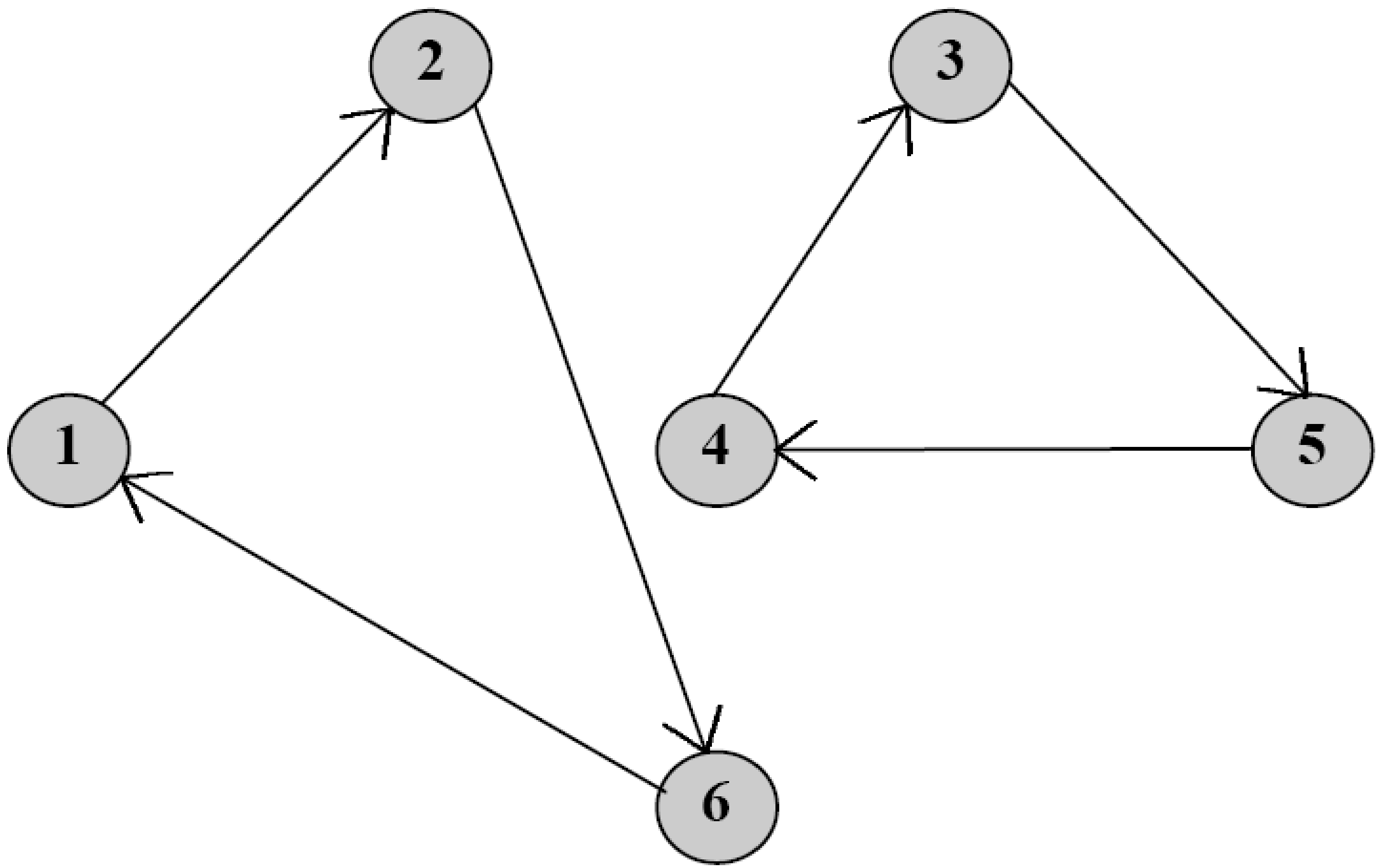
$$\sum_{j}^{n} X_{ij} = 1, \quad \forall_{\mathbf{i:\ i=1,..,n}}$$

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

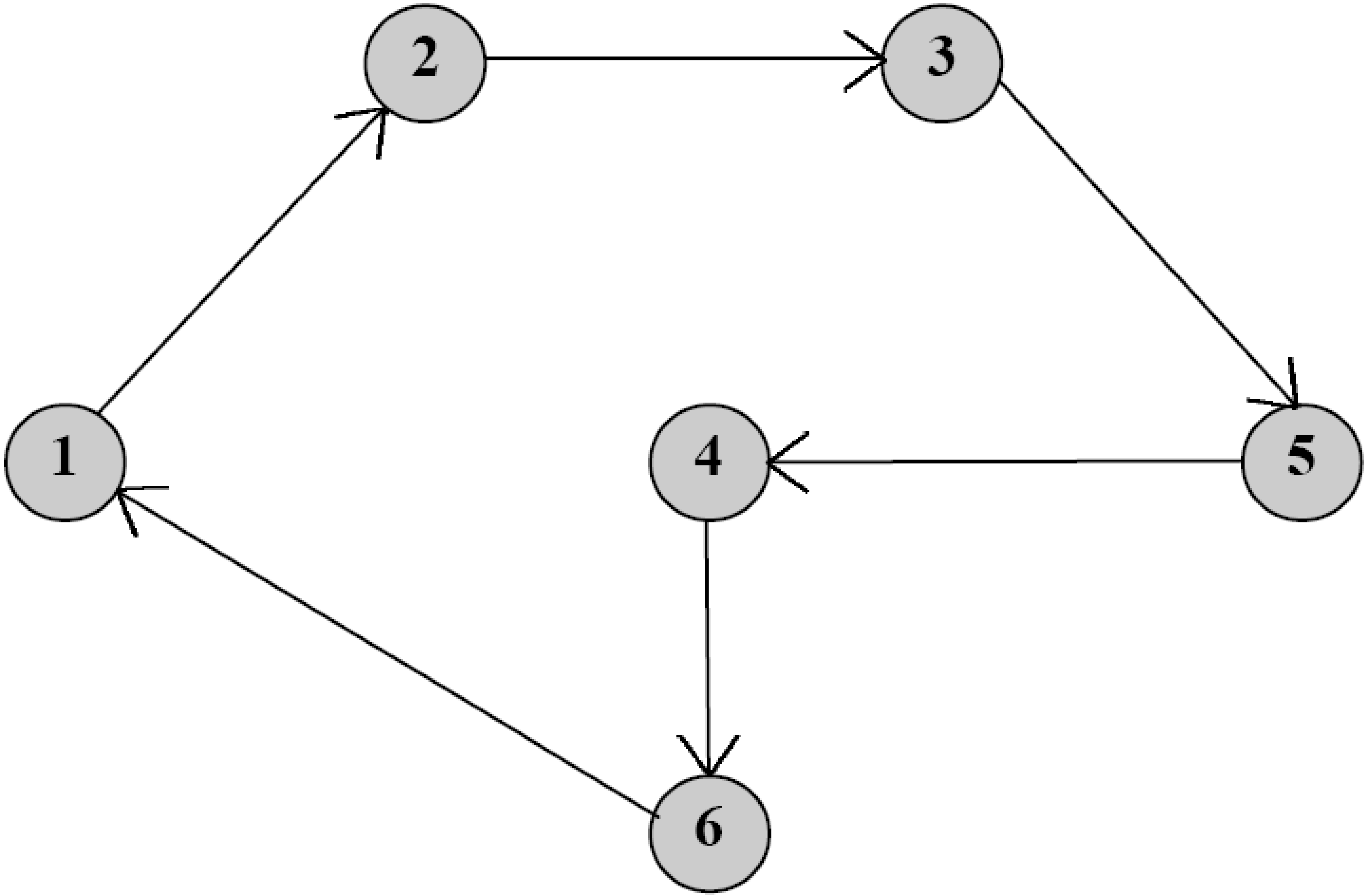
(2) entre dois quaisquer subconjuntos complementares de cidades (S e $\overline{\text{S}}$) há pelo menos um arco de ligação:

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

Restrição 1 não garante a solução



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



Exercício de Formulação

Problema da Mistura com restrição especial

- 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.

•Formulação por PL

Variável de decisão: x_i=quantidade do alimento i a ser usado na dieta.

Min Z = $1{,}0\, x_1 + 0{,}8\, x_2 + 1{,}2\, x_3 + 6{,}0\, x_4$

Sujeito a

$10\, x_1 + 5\, x_2 + 9\, x_3 + 10\, x_4 \geq 80$ //Vitamina A

$8\, x_1 + 7\, x_2 + 6\, x_3 + 6\, x_4 \geq 70$ //Vitamina B

$15\, x_1 + 3\, x_2 + 4\, x_3 + 7\, x_4 \geq 100$ //Vitamina C

$20\, x_1 + 2\, x_2 + 3\, x_3 + 9\, x_4 \geq 60$ //Vitamina D

$x_i \geq 0$, i=1,…,4, sendo 1 para leite, 2 para feijão, 3 para arroz e 4 para carne.

•Formulação por PL

As restrições anteriores são suficientes?

$x_1 \,.\, x_4 = 0$ não é uma equação linear, então não serve para PL.

Alternativa: criar variável de auxiliar: $y_i = 1$ se o alimento *i* for usado na dieta, e zero caso contrário.

$y_1 + y_4 \leq 1$

$x_1 \leq M\, y_1$

$x_4 \leq M\, y_4$, sendo a M a maior constante possível (BIG M)

Exercício de Formulação

Problema da Mistrutura com restrição especial

- Modelo no LPSolve

```
/* Objective function  - ModeloMisturaBinária.lp */
min: 1 x1 + 0.8 x2 1.2 x3 + 6 x4 ;

10 x1 + 5 x2 + 9 x3 + 10 x4 >= 80;
 8 x1 + 7 x2 + 6 x3 +  6 x4 >= 70;
15 x1 + 3 x2 + 4 x3 +  7 x4 >= 100;
20 x1 + 2 x2 + 3 x3 +  9 x4 >= 60;

y1 + y2 <=1;

x1 <= 9999999 y1;
x4 <= 9999999 y2;

bin y1,y2; /* Variável binárias */
```

Execícios

- Empresa de Construção Civil
- Fluxo Máximo em Rede
- Caminho Mínimo em Grafos
- Problema de Designação
- Problema de Designação com Gargalo
- Problema de Sequenciamento em Processadores Paralelos e Idênticos.

- Empresa de Construção Civil

Variáveis de decisão:

x_{1j}=número de contratos de 1 semestre com início no semestre j, j=1,2,3.

x_{21} =número de contratos de 2 semestre incluindo o semestre 1 e 2.

x_{22} =número de contratos de 2 semestre incluindo o semestre 2 e 3.

x_3 =número de contratos de 3 semestres.

- Empresa de Construção Civil

Função Objetivo

Min Z = $400\,(x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_3) +$

$6\,x_{11} + 6{,}5x_{12} + 7x_{13} + (6+6{,}5)x_{21} + (6{,}5+7)x_{22} + (6+6{,}5+7)x_3$

Restrições

Restrição de homens-hora

$1050(x_{11} + x_{12} + x_{13} + 2(x_{21} + x_{22}) + 3x_3) \geq 80.000$

Restrições

Restrição de homens-hora

$$1050(x_{11} + x_{12} + x_{13} + 2(x_{21} + x_{22}) + 3x_3) \geq 80.000$$

Número de operários por semestre

$$x_{11} + x_{21} + x_3 \geq 25 \qquad //\text{Semestre 1}$$

$$x_{12} + x_{21} + x_{22} + x_3 \geq 25 \qquad //\text{Semestre 2}$$

$$x_{13} + x_{22} + x_3 \geq 25 \qquad //\text{Semestre 3}$$

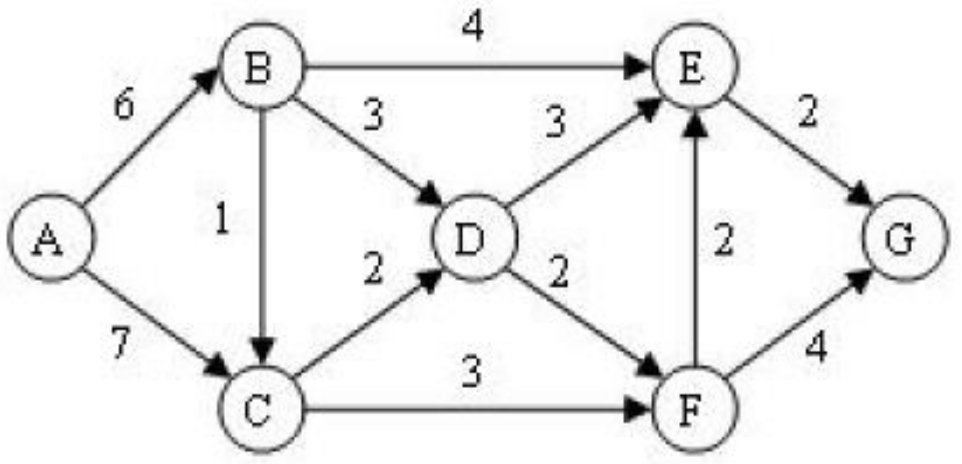
Número máximo de contratos de 3 semestres

$$x_3 \leq 15$$

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_3 \in Z^+$$

Fuxo em Rede.

A figura a seguir representa uma rede de comunicação de dados entre computadores. Os números ao lado das arestas representam a capacidade máxima em MBytes por segundo que pode ser transmitido de um computador para outro. Admita que a transmissão só seja possível no sentido especificado pela seta. Formule um modelo de programação linear que seja capaz de calcular este fluxo máximo entre o vértice A e G.



Fuxo em Rede.

Variáveis de decisão:

x_{ij}=a quantidade de fluxo a ser enviada do vértice *i* ao vértice *j*.

Função Objetivo

Max $F = x_{AB} + x_{AC} = x_{EG} + x_{FG}$

Sujeito a

Equilíbrio do fluxo

Fuxo em Rede.

Equilíbrio do fluxo

$x_{EG} + x_{FG} - (x_{AB} + x_{AC}) = 0$

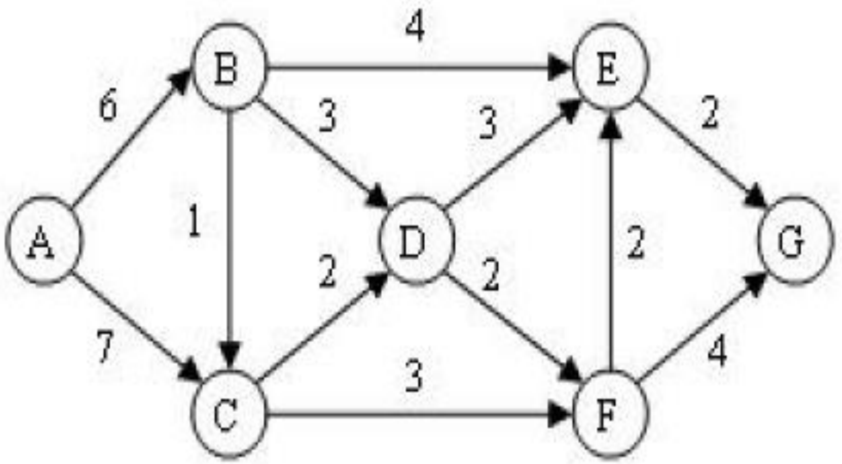
$x_{AB} - (x_{BE} + x_{BD} + x_{BC}) = 0$

$x_{AC} + x_{BC} - (x_{CD} + x_{CF}) = 0$

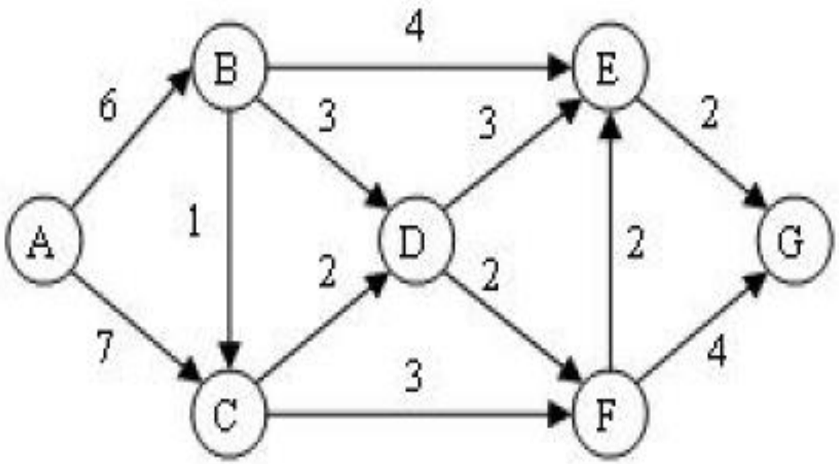
$x_{BD} + x_{CD} - (x_{DE} + x_{DF}) = 0$

$x_{DE} + x_{BE} + x_{FE} - x_{EG} = 0$

$x_{DF} + x_{CF} - (x_{FE} + x_{FG}) = 0$



Fuxo em Rede.



$x_{AC} \leq 7$ $x_{AB} \leq 6$

$x_{BE} \leq 4$ $x_{BD} \leq 3$ $x_{BC} \leq 1$

$x_{CD} \leq 2$ $x_{CF} \leq 3$

$x_{DF} \leq 2$ $x_{DE} \leq 3$

$x_{EG} \leq 2$

$x_{FG} \leq 4$ $x_{FE} \leq 2$

$x_{ij} \in Z^{+}$

Fluxo em Rede

Generalize o modelo.

Considere os seguinte parâmetros:

V: o conjunto de vértices

Cap_{ij}: a capacidade da aresta (i , j)

n : a cardinalidade do conjunto de vértices, isto é, $n = |V|$.

Considere que o vértice 1 seja o vértice de origem e n como o vértice de destino, então generalize o modelo padra encontrar o fluxo máximo na rede.

•Modelo Generalizado do Fluxo em Rede

$$Max \sum_{j \in V} x_{1j}$$

S.a

$$\sum_{j \in V} x_{ij} \; - \sum_{j \in V} x_{ji} = 0 \quad \forall i \in V, i \neq 1 \; e \; i \neq n$$

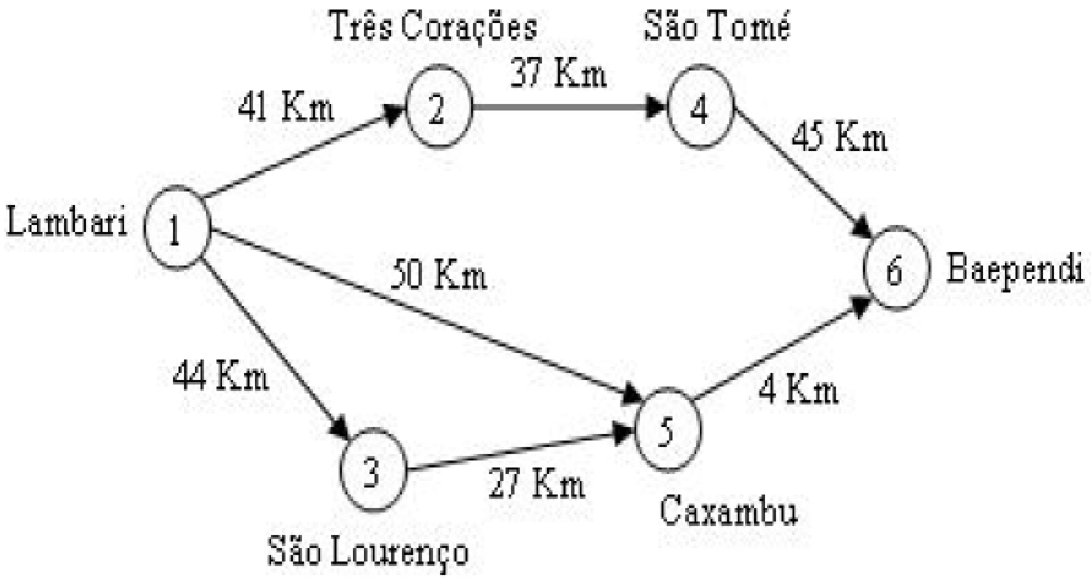
$$\sum_{j \in V} x_{1j} \; - \sum_{i \in V} x_{in} = 0$$

$$x_{ji} \leq Cap_{ij} \; \forall i, j \; \in V$$

$$x_{ji} \in \mathbb{Z}^{+} \qquad \forall i, j \in V$$

Caminho Mínimo

Uma fábrica de artigos de decoração, localizada em Lambari (MG), deve entregar uma grande quantidade de peças na cidade de Baependi (MG). A empresa quer saber qual o caminho que seu caminhão de entregas deve fazer para minimizar a distância total percorrida. A figura a seguir representa, na forma de rede, as ligações entre as cidades da região.



Caminho Mínimo

Variáveis de decisão:

x_{ij}= 1 se a aresta (i,j) pertencer ao caminho e zero caso contrário.

$$x_{ij} = \begin{cases} 1 \text{ se a aresta } (i,j) \text{ pertencer ao caminho} \\ \quad\quad\quad\quad 0 \text{ caso contrário} \end{cases}$$

Função Objetivo

Min F= $\sum_{i=1}^{6}\sum_{j=1}^{6} d_{ij}x_{ij}$

d_{ij} = distância entre o vértice *i* e o vértice *j*

Caminho Mínimo

Sujeito a

Equilíbrio de fluxo em rede

$$x_{12} + x_{13} + x_{15} = 1$$

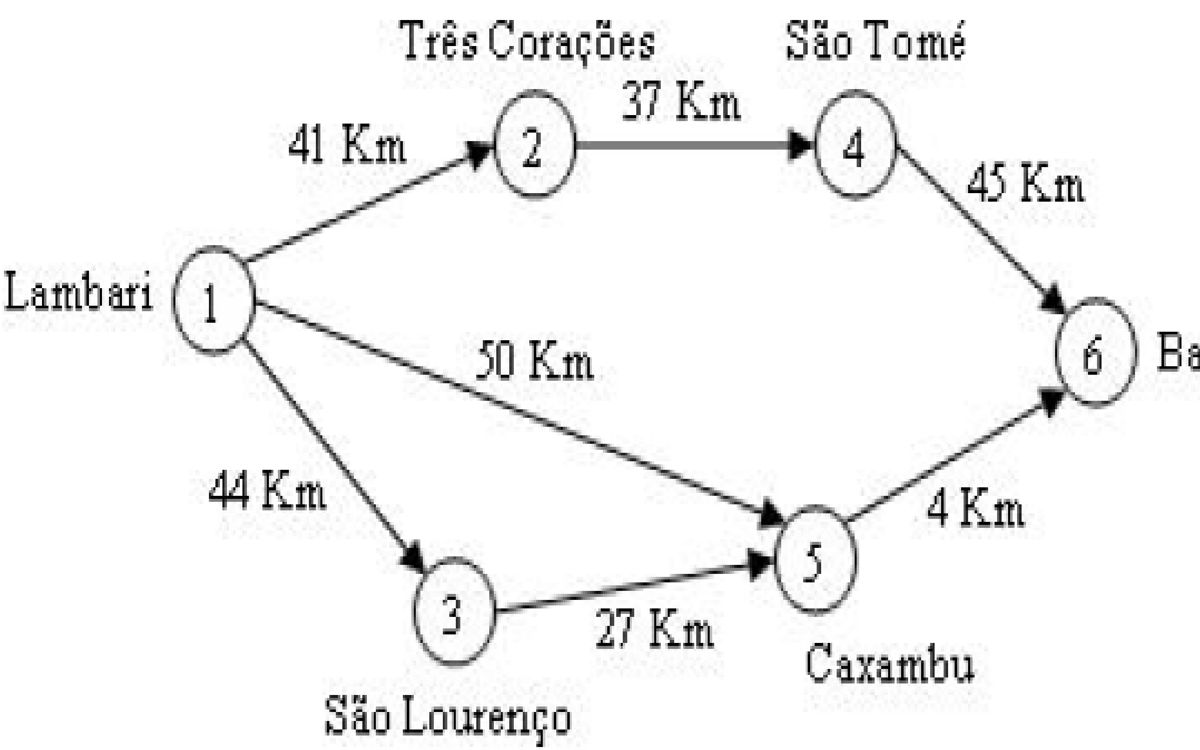
$$x_{12} - x_{24} = 0$$

$$x_{24} - x_{46} = 0$$

$$x_{15} + x_{35} - x_{56} = 0$$

$$x_{13} - x_{35} = 0$$

$$x_{56} + x_{46} = 1$$



Caminho Mínimo

Generalize o modelo.

Considere os seguinte parâmetros:

V: o conjunto de vértices

d_{ij}: a distância do vértice i ao vértice j

n : a cardinalidade do conjunto de vértices, isto é, $n = |V|$.

Considere que o vértice 1 seja o vértice de origem e n como o vértice de destino, então generalize o modelo para encontrar o caminho mínimo.

•Modelo Generalizado do Caminho Mínimo

$$Min \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}$$

S.a

$$\sum_{j \in V} x_{ij} \; - \sum_{j \in V} x_{ji} = 0 \quad \forall i \in V, i \neq 1 \; e \; i \neq n$$

$$\sum_{j \in V} x_{1i} = 1$$

$$\sum_{i \in V} x_{in} = 1$$

$$x_{ji} \in \{0,1\} \;\; \forall i, j \in V$$

- **Problema de designação linear.**

Considere n tarefas que devem ser designadas para n máquina, uma tarefa para cada máquina. Seja c_{ij} o custo de associar a tarefa i para a máquina j. Construa um modelo de programação matemática de forma que a soma dos custos de todas as atribuições seja mínima.

Variável de decisão:

x_{ij}: 1 se a tarefa i for designada para a máquina j, e 0 caso contrário.

•Problema de designação linear.

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

S.a

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall\, i = 1, \dots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall\; j = 1, \dots, n$$

$$x_{ji} \in \{0,1\} \;\; \forall i, j = 1, \dots, n$$

- **Problema de designação com gargalo.**

Considere o problema anterior, porém, o custo c_{ij} seja interpretado como carga de trabalho. O objetivo é designar as tarefas às máquinas de forma que carga de trabalho entre as máquinas seja distribuída de forma equitativa, ou seja, que a diferença entre as atribuições seja mínima.

•Problema de designação com gargalo.

$$Min\ z$$

S.a

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall\, i = 1, \dots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall\ j = 1, \dots, n$$

$$c_{ij}x_{ij} \leq z \quad \forall\ i, j = 1, \dots, n$$

$$x_{ji} \in \{0,1\} \ \forall i, j = 1, \dots, n$$

- **Sequenciamento em processadores paralelos e idênticos**

Suponha que seja necessário executar uma lista de 10 *jobs* em um conjunto de 3 processadores. Sabe-se que cada *job* pode ser executado em qualquer ordem e em qualquer processador, sendo o tempo de processamento independente do processador. O tempo (em minutos) gasto para execução de cada *job* é: 6, 4, 5, 4, 3, 7, 8, 5, 3 e 3. Elabore o modelo de programação matemática que minimize o tempo de execução de todos os *jobs*.

•Modelo generalizado

$$x_{ij} = \begin{cases} 1, & se\ o\ Job\ j\ for\ executado\ no\ processador\ i. \\ 0, & caso\ contrário. \end{cases}$$

Notaçãoes:

Jobs*:* *Conjunto de jobs*

Procs*:* *Conjunto de processadores*

$t_j =$ *Tempo de execução do job j*

$Min\ C_{max}$

S.a

$$\sum_{i \in Procs} x_{ij} = 1 \qquad \forall\ j\ \in Jobs$$

$$\sum_{j \in Jobs} t_j x_{ij} \leq C_{max} \quad \forall\ i \in Procs$$

$$x_{ji} \in \{0,1\}\ \forall\ i \in Procs, \qquad \forall\ j \in Jobs$$

- **Problema do Caixeiro Viajante**
- Dado um conjunto de n cidades e uma matriz de distâncias d_{ij} entre elas, o Problema do Caixeiro Viajante (PCV), ou *Traveling Salesman Problem* - TSP, consiste em estabelecer uma rota para um Caixeiro, iniciando seu percurso em uma cidade, chamada cidade origem, passar por todas as demais $n - 1$ cidades uma única vez e retornar à cidade origem percorrendo a menor distância possível. Seja o grafo G = (Cidades, A), onde Cidades é conjunto de cidades (clientes) e A o conjunto de arestas ligando duas cidades, isto é, $A = \{(i, j) \mid i \neq j\}$. Seja d_{ij} a distância da cidade i para a cidade j. Formule um modelo de PLI que possa solucionar o problema. Sugestão: utilize ideias do modelo de fluxo em rede para eliminar possíveis subciclos.

•Caixeiro Viajante

(a) Variáveis de decisão:

x_{ij}: variável binária que assume valor 1 se o arco (i, j) for utilizado e 0, caso contrário

f_{ij}: quantidade de fluxo enviada da cidade i para a cidade j

(b) Função objetivo:

$$\min \sum_{i \in Cidades} \sum_{j \in Cidades} d_{ij} x_{ij}$$

(c) Restrições:

c.1) À cada cidade k só chega um arco:

$$\sum_{i \in Cidades} x_{ik} = 1 \quad \forall k \in Cidades$$

c.2) De cada cidade k só sai um arco:

$$\sum_{j \in Cidades} x_{kj} = 1 \ \ \forall k \in Cidades$$

c.3) Eliminação de subciclos:

$$\sum_{i \in Cidades} f_{ik} - \sum_{j \in Cidades} f_{kj} = 1 \ \ \forall k \in Cidades \mid k \neq 1$$

$$f_{ij} \leq (n-1)x_{ij} \ \ \forall i \in Cidades, \ \ \forall j \in Cidades$$

c.4) Integralidade e não-negatividade:

$$x_{ij} \in \{0,1\} \ \ \forall i \in Cidades, \ \ \forall j \in Cidades$$

$$f_{ij} \geq 0 \ \ \forall i \in Cidades, \ \ \forall j \in Cidades$$

Reformulação do PCV

Problema do Caixeiro Viajante

Reformulação

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

sujeito a

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,\dots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,\dots, n$$

$$\sum_{i\in S} \sum_{j\in S} x_{ij} \le |S| - 1 \qquad \forall S \subset \{1 \dots n\}$$

$$x_{ij} = 0/1, \forall i, j$$

Existem $(2^n - 2)$ restrições para eliminação de subrotas.

Reformulação Automática : Planos de corte

- Número exponencial de restrições para eliminação de subrotas. O que fazer? $(2^n - 2)$
- Gerar apenas as que são estritamente necessárias!

Algoritmo de Planos de Corte

- Resolve o Problema da Designação associado(Relaxação)
- Se a solução ótima é um circuito hamiltoniano (CH), PARE.
- Enquanto (it <nmax) ou (solução não é um CH) faça:
  - Gere inequações para eliminar sub-rotas e acrescente ao Problema Atual.
  - Resolva o novo Problema.