Pesquisa Operacional / Programação Matemática

Otimização discreta

Modelagem com variáveis binárias: problemas clássicos

Alysson M. Costa – ICMC/USP

Problemas clássicos

- Importância histórica...

  ... e prática.

- Usados para modelar problemas reais e como subproblemas em problemas maiores (e mais frequentes na prática).

Problema da mochila

- Idéia básica:

  diversos ítens, cada um com um valor de *utilidade* e um peso. Queremos levar a maior soma de *utilidades* possível (não podemos ultrapassar a capacidade da mochila)

Ex. (problema do ladrão)

Item	Size	Value
1 - ring	1	15
2 - candelabra	5	10
3 - radio	3	9
4 - elvis	4	5

capacidade: 8

Outras aplicações (!!):

- investimentos, produção, logística, etc. etc. etc...

retirado de: http://webspace.ship.edu/thbrig/DynamicProgramming/Knapsack%20Program/index.html

Exemplo 3.7 Considere um capital para investimento $b = 100$, $n = 8$ projetos, e os seguintes vetores de parâmetros:

$$\mathbf{p} = [p_j] = [41 \quad 33 \quad 14 \quad 25 \quad 32 \quad 32 \quad 9 \quad 19]$$
$$\mathbf{a} = [a_j] = [47 \quad 40 \quad 17 \quad 27 \quad 34 \quad 23 \quad 5 \quad 44]$$

Sol ótima ?

A solução ótima é dada por $x_2 = x_4 = x_6 = x_7 = 1$, com valor 99. Esta solução utiliza 40 + 27 + 23 + 5 = 95 unidades do capital. ■

Formulação

Variáveis:

$$x_j = \begin{cases} 1 & \text{se o projeto } j \text{ é selecionado} \\ 0 & \text{caso contrário} \end{cases}$$

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

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$\mathbf{x} \in B^n$$

Problema da mochila (variações)

- mochila inteira:
  - múltiplas unidades de um mesmo item podem ser colocadas na mochila.

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

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$\cancel{\mathbf{x} \in B^n} \qquad \mathbf{x} \in Z_+^n$$

Problema da mochila (variações)

- múltiplas mochilas:
  - cada item pode entrar em uma de várias mochilas (caminhões, contêineres)...

variáveis:

Cada item j tem uma lucratividade p_j e um peso w_j,

$$x_{ij} = \begin{cases} 1 & \text{se o item } j \text{ é colocado na mochila } i \\ 0 & \text{caso contrário} \end{cases}$$

$$\max \sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij}$$

$$\sum_{j=1}^{n} w_j x_{ij} \leq b_i, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} x_{ij} \leq 1, \quad j = 1, \ldots, n$$

$$\mathbf{x} \in B^{mn}$$

Variação das múltiplas mochilas

- Múltiplos processadores paralelos:

  o peso (*tempo de processamento*) de cada item pode depender da mochila (*processador*) ao qual ele for alocado.

bin packing

- Encontrar o menor número de mochilas tal que *todos* os itens sejam empacotados.

$$y_i = \begin{cases} 1 & \text{se a mochila } i \text{ é usada} \\ 0 & \text{caso contrário} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{se o item } j \text{ é colocado na mochila } i \\ 0 & \text{caso contrário} \end{cases}$$

$$\min \sum_{i=1}^{n} y_i$$

$$\boxed{\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, \ldots, n}$$

$$\boxed{\sum_{j=1}^{n} w_j x_{ij} \le b y_i, \quad i = 1, \ldots, n}$$

$$\mathbf{x} \in B^{nn}, \mathbf{y} \in B^{n}$$

todos os itens são alocados

as capacidades das mochilas são respeitadas

Problemas de designação

- Já visto anteriormente. Alocar n tarefas a n agentes de modo a minimizar o custo total de designação;

A execução da tarefa j pelo agente i tem um custo c_{ij}.

$$x_{ij} = \begin{cases} 1 & \text{se a tarefa } j \text{ é designada ao agente } i \\ 0 & \text{caso contrário} \end{cases}$$

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

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

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

$$\mathbf{x} \in B^{nn}$$

Problemas de designação *generalizada*

- m agentes, n tarefas
- cada tarefa deve ser realizada por um único agente.
- cada agente pode realizar mais de uma tarefa.
- cada agente i gasta a_{ij} de um dado recurso (tempo, e.g.) para executar a tarefa j.
- cada agente dispõe de b_i unidades do recurso.

Problemas de designação generalizada

$$x_{ij} = \begin{cases} 1 & \text{se a tarefa } j \text{ é designada ao agente } i \\ 0 & \text{caso contrário} \end{cases}$$

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

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1, \ldots, n$$

$$\cancel{\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, n} \qquad \sum_{j=1}^{n} a_{ij} x_{ij} \le b_i, \quad i = 1, \ldots, m$$

$$\mathbf{x} \in B^{nn}$$

Exemplo 3.8 Considere $m = 3$ agentes, $n = 8$ tarefas e os seguintes parâmetros:

$$\mathbf{C} = [c_{ij}] = \begin{bmatrix} 15 & 61 & 3 & 94 & 86 & 68 & 69 & 51 \\ 21 & 28 & 76 & 48 & 54 & 85 & 39 & 72 \\ 21 & 21 & 46 & 43 & 21 & 3 & 84 & 44 \end{bmatrix}$$

$$\mathbf{A} = [a_{ij}] = \begin{bmatrix} 31 & 69 & 14 & 87 & 51 & 65 & 35 & 54 \\ 23 & 20 & 71 & 86 & 91 & 57 & 30 & 74 \\ 20 & 55 & 39 & 60 & 83 & 67 & 35 & 32 \end{bmatrix}$$

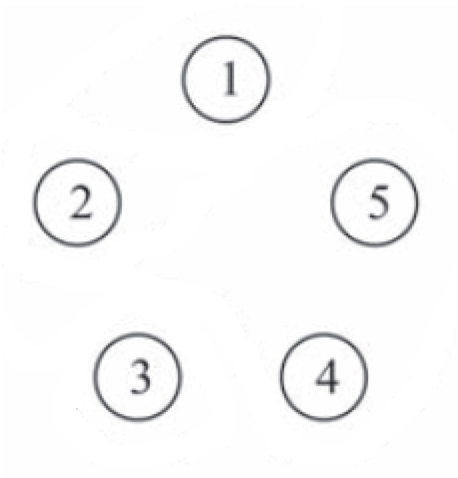
$$\mathbf{b} = [b_i] = [100 \quad 100 \quad 100]$$

A solução ótima é dada por $x_{13} = x_{15} = x_{17} = 1$; $x_{21} = x_{22} = x_{26} = 1$; $x_{34} = x_{38} = 1$, isto é, as tarefas 3, 5 e 7 são designadas ao agente 1, as tarefas 1, 2 e 6 são designadas ao agente 2, e as tarefas 4 e 8 são designadas ao agente 3. O valor da solução ótima é 379. Note que somente o agente 3 tem folga de recurso de 8 unidades. Se a capacidade dos agentes 1 ou 2 é reduzida para 99, então o exemplo não tem solução factível. ■

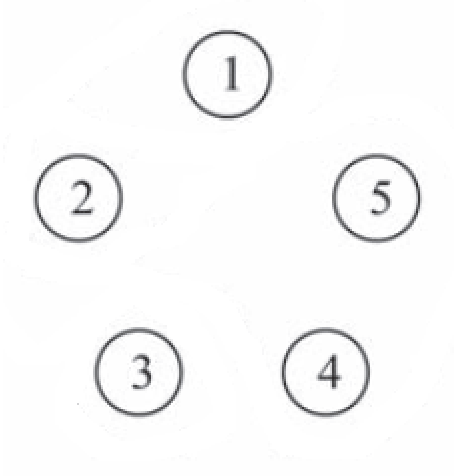
Problemas de cobertura/partição/empacotamento

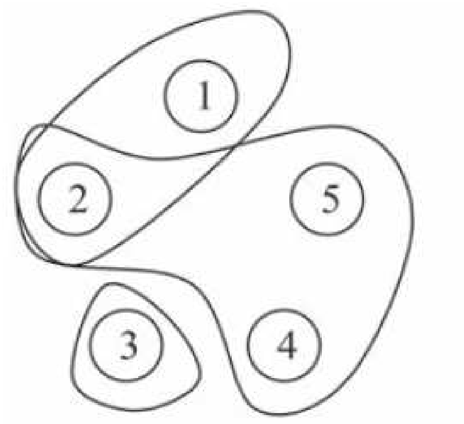
- Selecionar subconjuntos de um conjunto inicial de forma a *cobrir, particionar ou empacotar* o conjunto inicial.

$S = \{1, 2, 3, 4, 5\}$

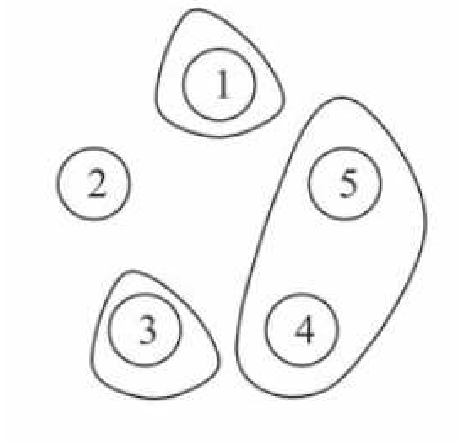


Problemas de cobertura/partição/empacotamento

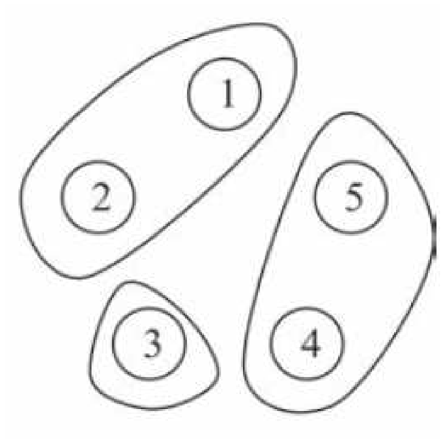




Cobertura



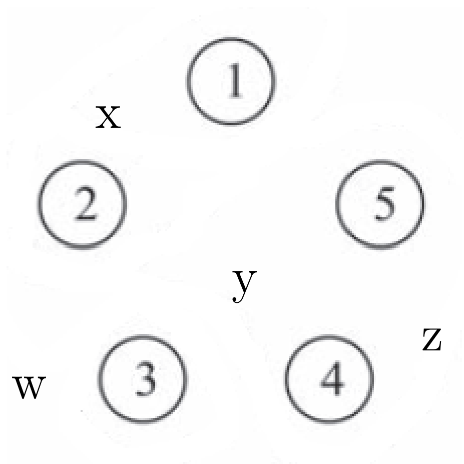
Empacotamento



Partição

- Exemplo de aplicação:

  localização de facilidades de emergência (corpo de bombeiros, ambulâncias)



x consegue atender em 10 minutos (tempo máximo desejado) os bairros 1 e 2;

x: (1,2)

y: (2,4,5)

w: (3)

z: (4,5)

cobertura, empacotamento ou particionamento ?

Cobertura

- Exemplo:

x y w z

$$S_1 = \{1,2\},\ S_2 = \{1,3,5\},\ S_3 = \{2,4,5\},\ S_4 = \{3\},\ S_5 = \{1\},\ S_6 = \{4,5\}$$

facilidade de atendimento j com custo de instalação c_j.

$$x_j = \begin{cases} 1 & \text{se a facilidade } j \text{ é selecionada} \\ 0 & \text{caso contrário.} \end{cases}$$

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

$$\begin{array}{lr}
x_1 + x_2 + x_5 \geq 1 & \text{(bairro 1)} \\
x_1 + x_3 \geq 1 & \text{(bairro 2)} \\
x_2 + x_4 \geq 1 & \text{(bairro 3)} \\
x_3 + x_6 \geq 1 & \text{(bairro 4)} \\
x_2 + x_3 + x_6 \geq 1 & \text{(bairro 5)}
\end{array}$$

$$\mathbf{x} \in B^6$$

De maneira geral

$$\begin{aligned} &\min \mathbf{c}^{\mathbf{T}}\mathbf{x} \\ &\mathbf{Ax} \geq \mathbf{1} \\ &\mathbf{x} \in B^n, \end{aligned}$$

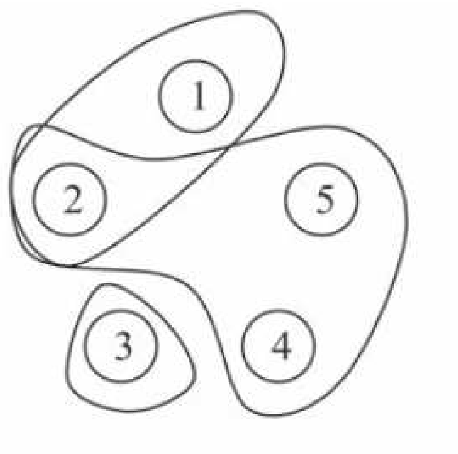
Cobertura

$$\begin{aligned} &\max \mathbf{c}^{\mathbf{T}}\mathbf{x} \\ &\mathbf{Ax} \leq \mathbf{1} \\ &\mathbf{x} \in B^n \end{aligned}$$

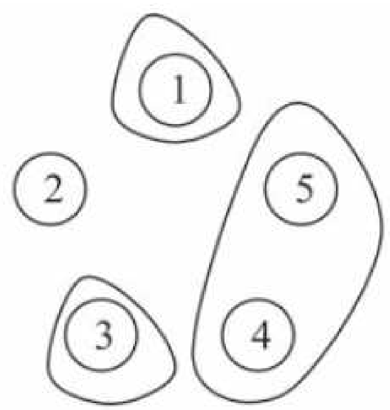
Empacotamento

$$\begin{aligned} &\min \mathbf{c}^{\mathbf{T}}\mathbf{x} \\ &\mathbf{Ax} = \mathbf{1} \\ &\mathbf{x} \in B^n \end{aligned}$$

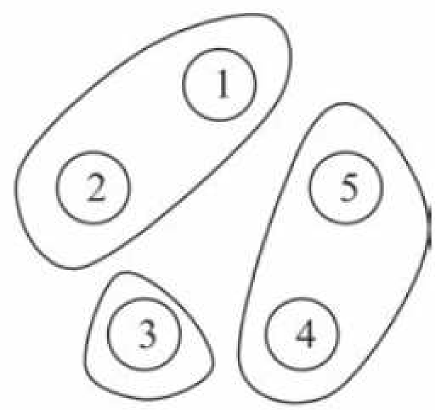
Particionamento



Cobertura

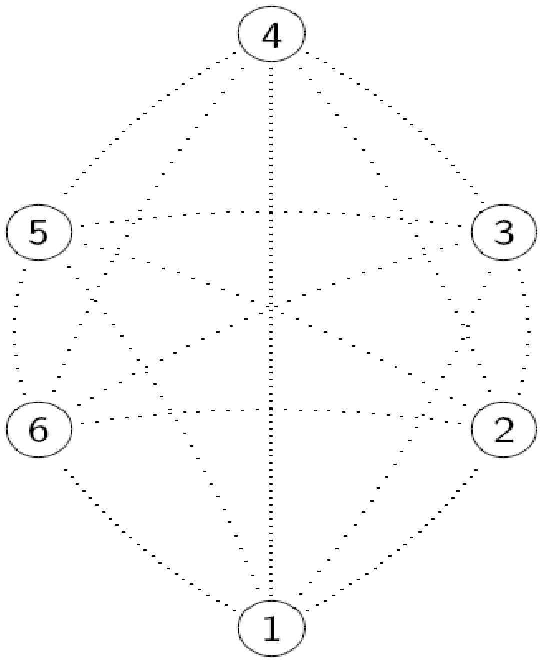


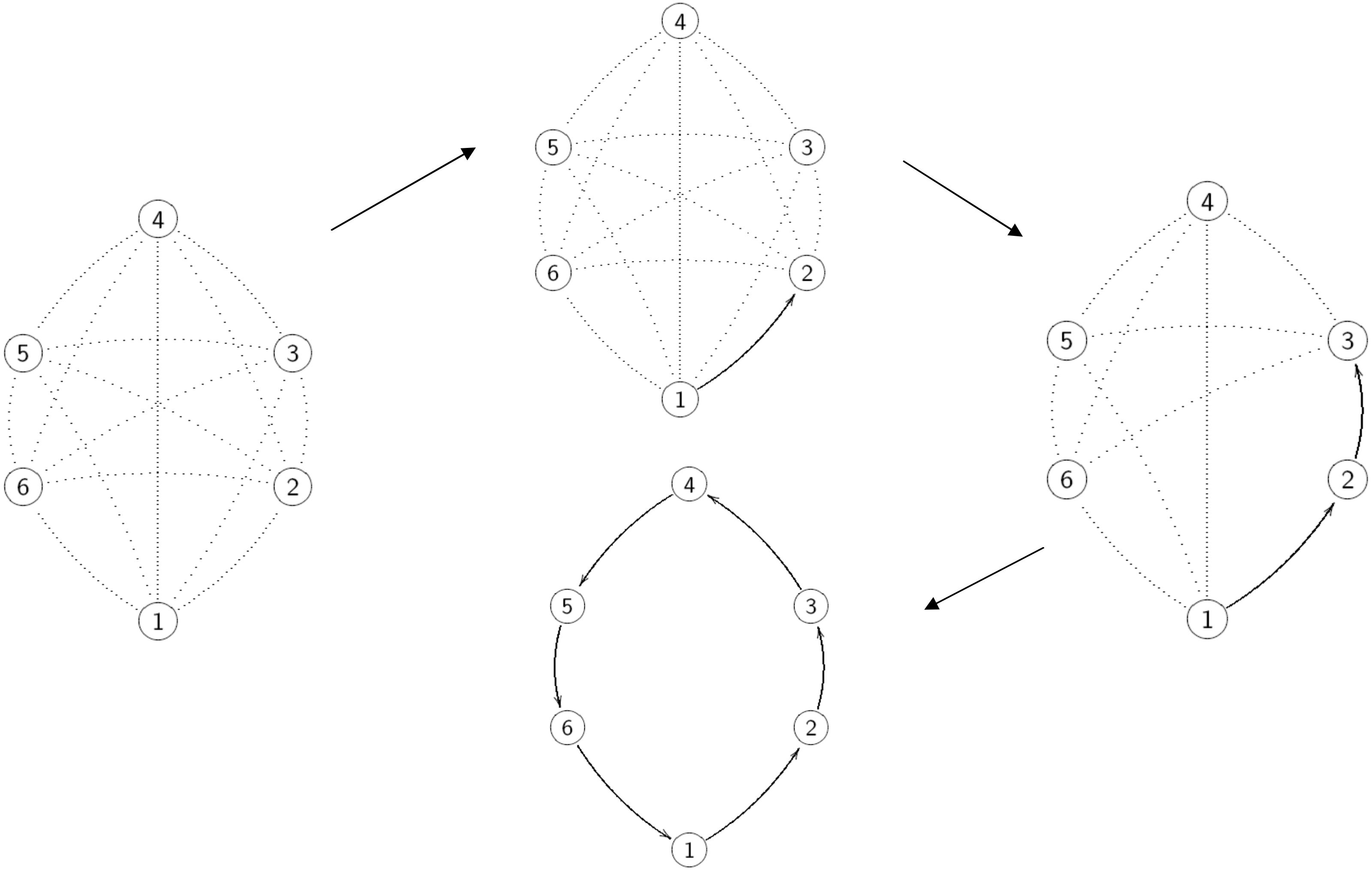
Empacotamento



Partição

Caixeiro viajante




4
5
3
6
2
1
4
5
3
6
2
1
4
5
3
6
2
1
4
5
3
6
2
1

Formulação matemática

$$\text{Min} \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\sum_{i,j\in S} x_{ij} \leq |S| - 1, \quad S \subseteq N - \{1\}, |S| \geq 2$$

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

$$\sum_{j=1}^{n} x_{ji} = 1 \quad i = 1, \ldots, n$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A$$