Pesquisa Operacional / Programação Matemática

Otimização discreta

Modelagem com variáveis binárias: problemas clássicos

Breve Comentários (aula anterior)

- Em geral, não faz sentido resolver a relaxação linear e arredondar a solução ótima obtida. A solução arredondada pode nem sequer ser factível ou pode estar muito afastada da solução ótima.
- Há circunstâncias práticas em que o arredondamento pode fazer sentido, tendo-se a consciência de que está obtendo uma solução que pode não ser ótima (por exemplo, na prática será muito diferente x=1999.5 ou x=2000?).
- Um problema PI (programação linear) é muito mais difícil de resolver do que o PL (perdeu-se a convexidade do conjunto de soluções factíveis...).
- Para PL são conhecidas condições necessárias e suficientes de otimalidade.
- Para PI não são conhecidas condições necessárias nem suficientes de otimalidade: dada uma solução factível, a única forma de determinar se ela é ótima ou não é demonstrar que não existe nenhuma solução factível com melhor valor.

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 (*Knapsack problem*)

- Idéia básica:

  diversos itens, 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

Problema da mochila (*Knapsack problem)*

- variáveis de decisão xj que significam o número de itens do tipo j a incluir na mochila. Cada item está associado a um lucro pj e um peso aj.
- Caso1: xj representa a quantidade de objetos do tipo j selecionado e pode assumir um valor fracionário. O problema pode ser formulado como:

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

$$sujeito\ a:$$

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

$$x_j \geq 0, j = 1, \ldots, n$$

Exemplo – Mochila - PL

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

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

Sol ótima ?

Considere um vetor que faça a razão de p_j / a_j

Razão: $p = [0.87, 0.83, 0.82, 0.92, 0.94, 1.39, 1.5, 0.43]$

Ordene os itens em ordem decrescente de razão, teremos o vetor ordenado:

$p = [7, 6, 5, 4, 1, 2, 3, 8]$

como temos uma só restrição, a solução será $x_7 = 16.67$

as demais variáveis são nulas e o valor da função objetivo 150.

Problema da mochila (variações)

- Caso 2: xj representa a quantidade de objetos do tipo j selecionado e não pode assumir um valor fracionário
- mochila inteira:
  - múltiplas unidades de um mesmo item podem ser colocadas na mochila.

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

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

$$\mathbf{x} \in Z_+^n$$

Exemplo – Mochila - Inteira

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

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

<u>Solução não trivial.</u>

Considere um vetor que faça a razão de p_j/a_j

Razão: $p=[0.87,0.83,0.82,0.92,0.94,1.39,1.5,0.43]$

Ordene os itens em ordem decrescente de razão, teremos o vetor ordenado:

$p=[7,6,5,4,1,2,3,8]$

Considerando a mesma estratégia e arredondando temos $x_7=16$

as demais variáveis são nulas e o valor da função objetivo 144. Não é a solução ótima.

$x_7=9$ $\quad x_6=2$ e o valor da solução é 145 (Resolva o exemplo no excel.

Problema da mochila (variações)

- Caso 3: xj pode assumir um valor fracionário mas podemos levar no máximo uma unidade do objeto.

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

$$\text{sujeito a:}$$

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

$$0 \leq x_j \leq 1 \quad j = 1, \ldots, n$$

Solução trivial.... Por que?

Exemplo – Mochila

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

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

Considere um vetor que faça a razão de p_j/a_j

Razão: $p = [0.87, 0.83, 0.82, 0.92, 0.94, 1.39, 1.5, 0.43]$

Ordene os itens em ordem decrescente de razão, teremos o vetor ordenado:
$p = [7,6,5,4,1,2,3,8]$

Faça $x_j = 1$ enquanto couber na mochila e $x_k = \left(\dfrac{b - \sum_{j=1}^{k-1} a_j}{a_k} \right)$

No exemplo:

$x_7 = 1$ e sobra um espaço na mochila de 94.
$x_6 = 1$ e sobrou espaço de 71 na mochila
$x_5 = 1$ e sobrou espaço de 37 na mochila
$x_4 = 1$ e sobrou espaço de 10 na mochila
$x_1 = 0.21$

Formulação

Caso 4: podemos selecionar no máximo uma unidade de cada objeto.

Variáveis:

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

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

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

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

Solução não trivial

Mochila – 0-1

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

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

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. ■

Problema da mochila (variações)

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

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

variáveis:

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

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

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

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

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

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}.

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

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

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

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

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

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. ■

Problema de Localização de Facilidades.

- São dadas n potenciais localizações para a instalação de facilidades e m clientes que devem ser atendidos por estas facilidades. O custo fixo de instalar uma facilidade no local j é igual a c_j. Existe ainda um custo f_{ij} para atender um cliente i a partir da facilidade j. Cada cliente deve ser alocada a uma única facilidade. Determinar os locais onde devem ser instaladas as facilidades de modo a minimizar o custo total. Observe que um cliente só pode ser alocado a uma facilidade se ela tiver sido instalada. Ou seja, teremos as restrições: 1) o cliente só pode ser atendido por uma facilidade e será atendido pela facilidade j somente se ela estiver instalada.

Problema de Localização de Facilidades.

$$y_j = \begin{cases} 1 \text{ se a facilidade for instalada no local j.} \\ 0, \text{ caso contrário.} \end{cases}$$

$$x_{ij} = \begin{cases} 1 \text{ se o cliente i for atendido pela facilidade j.} \\ 0, \text{ caso contrário.} \end{cases}$$

Minimizar $z = \sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} x_{ij}$

Sujeito a:

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

$$x_{ij} \le y_j \quad \forall, i \in m, j \in n$$

$$x_{ij} \in \{0,1\},\ y_j \in \{0,1\} \forall i \in m,\ j \in n$$

Problema de Localização de Facilidades capacitado.

- São dadas n potenciais localizações para a instalação de facilidades e m clientes que devem ser atendidos por estas facilidades. O custo fixo de instalar uma facilidade no local j é igual a c_j. Existe ainda um custo f_{ij} para atender um cliente i a partir da facilidade j. Cada cliente tem uma demanda d_i e deve ser alocada a uma única facilidade. Cada facilidade tem uma capacidade conhecida ej. Determinar os locais onde devem ser instaladas as facilidades de modo a minimizar o custo total. Observe que um cliente só pode ser alocado a uma facilidade se ela tiver sido instalada. Ou seja, teremos as restrições: 1) o cliente só pode ser atendido por uma facilidade e será atendido pela facilidade j somente se ela estiver instalada.

Problema de Localização de Facilidades.

$$y_j = \begin{cases} 1 \text{ se a facilidade for instalada no local j.} \\ 0, \text{ caso contrário.} \end{cases}$$

$$x_{ij} = \begin{cases} 1 \text{ se o cliente i for atendido pela facilidade j.} \\ 0, \text{ caso contrário.} \end{cases}$$

Minimizar $z = \sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$

Sujeito a:

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

$$x_{ij} \le y_j \quad \forall, i \in m, j \in n$$

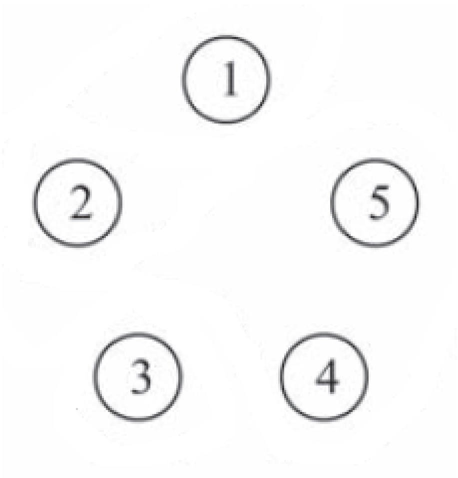
$$\sum_{i=1}^{m} d_i x_{ij} \le e_j y_j \quad \forall, \ j \in n$$

$$x_{ij} \in \{0,1\}, \ y_j \in \{0,1\} \forall i \in m, \ j \in n$$

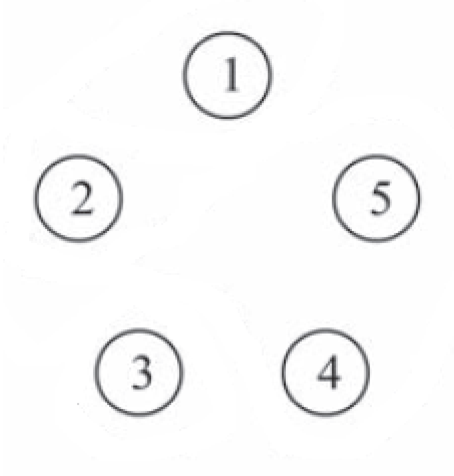
Problemas de cobertura/partição/empacotamento

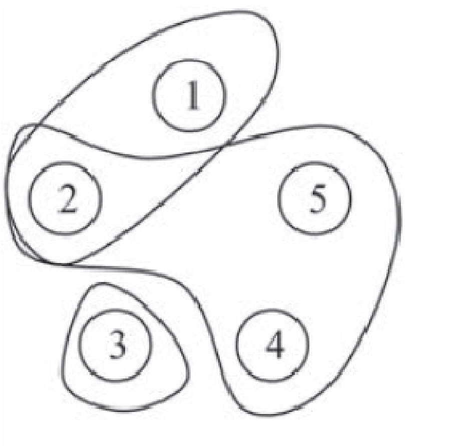
- Selecionar subconjuntos de um conjunto inicial de forma a *cobrir, particionar ou empacotar* o conjunto inicial.

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

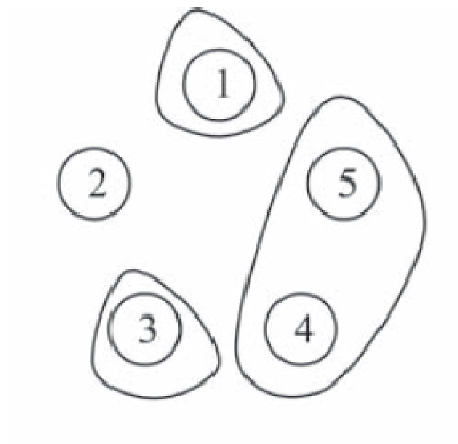


Problemas de cobertura/partição/empacotamento

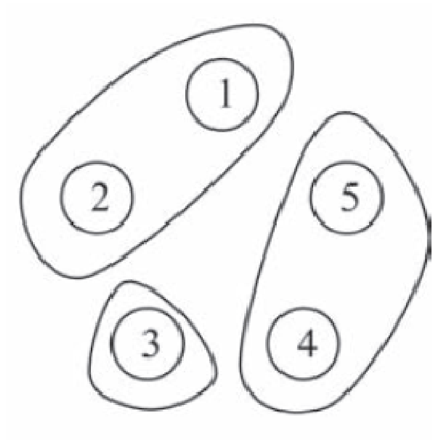




Cobertura



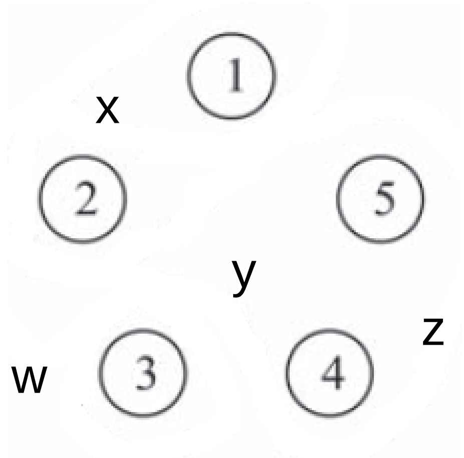
Empacotamento



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.

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

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

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

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

De maneira geral

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

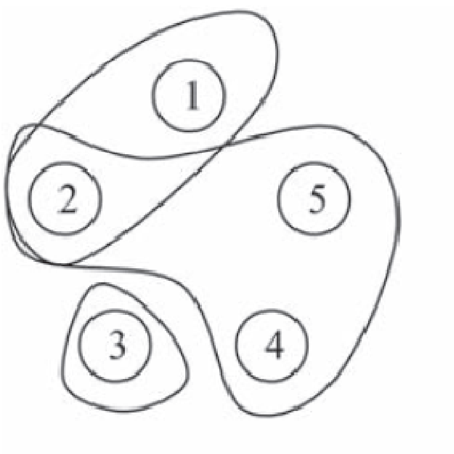
Cobertura

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

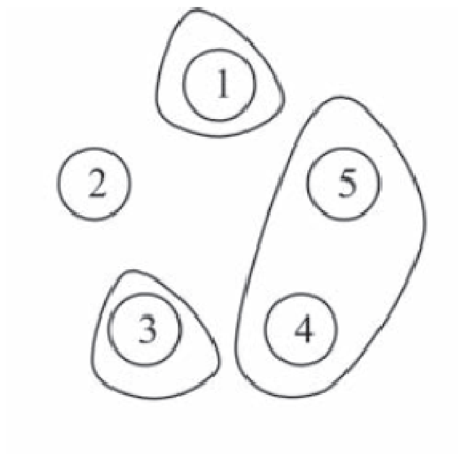
Empacotamento

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

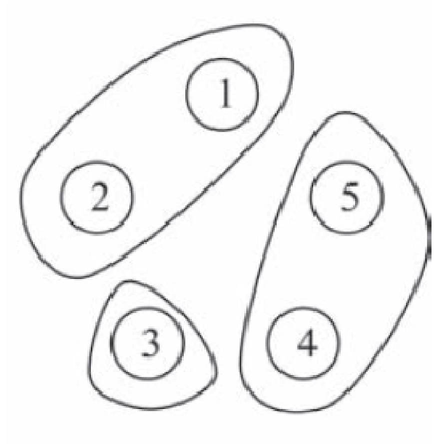
Particionamento



Cobertura



Empacotamento



Partição

O Problema do Caixeiro viajante

- Formulação de Dantzig –Fulkerson-Johnson
- O objetivo do modelo é determinar o ciclo hamiltoniano de custo (distância) mínimo. A formulação foi feita sobre um grafo G=(N,A) da seguinte forma:
- Definindo:

$$x_{ij} = \begin{cases} 1, \text{se a aresta (i,j) for visitada} \\ 0, \text{caso contrário.} \end{cases}$$

$$\text{Minimizar} \quad z = \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij}$$

$$\sum_{i=1}^{N} x_{ij} = 1 \quad \forall, \ j \in N$$

$$\sum_{j=1}^{N} x_{ij} = 1 \quad \forall, i \in N$$

Modela o Problema?

O Problema do Caixeiro viajante

$$x_{ij} = \begin{cases} 1, \text{se a aresta (i,j) for visitada} \\ 0, \text{caso contrário.} \end{cases}$$

$$\text{Minimizar} \quad z = \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij}$$

$$\sum_{i=1}^{N} x_{ij} = 1 \quad \forall, \ j \in N$$

$$\sum_{j=1}^{N} x_{ij} = 1 \quad \forall, i \in N$$

$$\sum_{i,\, j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N$$

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

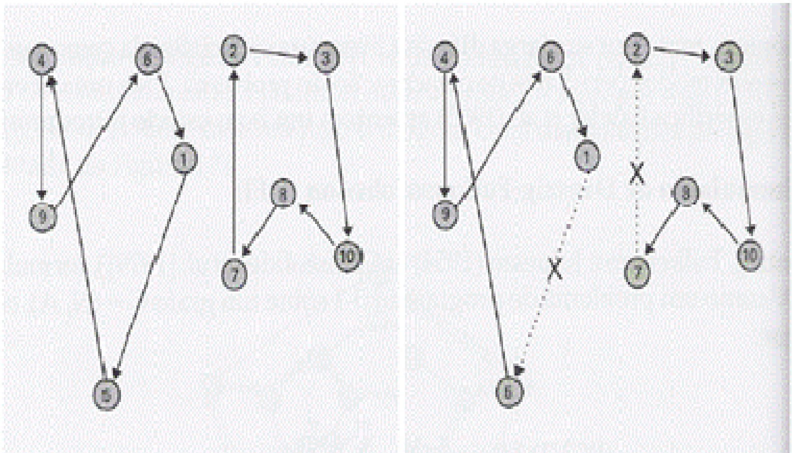


Figura 3: (a) Solução ilegal ($\sum_{i,j \in S_1} x_{i,j} = 5 \quad \sum_{i,j \in S_2} x_{i,j} = 5$); (b) Restrições associadas ($\sum_{i,j} x_{i,j} \leq | 5 | -1 \leq 4$), respectivamente

S é um sub-grafo de G e |S| é o número de vértices desse sub-grafo.
Assumimos que não temos a variável xii

Exercícios

- 1) Um aventureiro planeja fazer uma viagem acampando. Há cinco itens que o aventureiro deseja levar consigo, mas estes juntos, excedem o limite de 40 kg que ele supõe ser capaz de carregar. Para ajudar o processo de seleção, ele atribui valores, por ordem de crescente importância, a cada um dos itens segundo a tabela. Que itens devem ser colocados na mochila de forma a maximizar o valor total sem exceder as restrições de peso?

item	1	2	3	4	5
Peso (kg)	30	13	25	15	7
valor	100	60	70	15	15

Exercícios

- Um hospital deseja planejar os horários das enfermeiras de seu turno da noite. A demanda por enfermeiras no turno da noite no dia $j=1,...,7$ da semana é um número inteiro d_j. Cada enfermeira trabalha cinco dias consecutivos e descansa nos dois dias seguintes. O objetivo consiste em minimizar o número de enfermeiras contratadas.

- Variáveis de decisão (defina como):

- x_j: número de enfermeiras que começam seu horário no dia $j=1,...,7$

Exercícios

- Pretende-se instalar um número mínimo de postos de saúde para servir 6 povoados. Quantos, e em quais povoados deverão ser instalados os postos, de forma a que cada povoado não fique a mais do que 15Km de um posto? A Tabela a seguir fornece as distâncias (em km) dos povoados aos possíveis locais de instalação dos postos de saúde.

	P1	P2	P3	P4	P5	P6
P1	0	10	20	30	30	20
P2	10	0	25	35	20	10
P3	20	25	0	15	30	20
P4	30	35	15	0	15	25
P5	30	20	30	15	0	14
P6	20	10	20	25	14	0

- Apresente um modelo matemático de modo a minimizar o problema acima e discuta um procedimento heurístico para obtenção de uma solução factível para o problema e apresente a solução obtida. Como você faria para comparar a qualidade da solução heurística (deve ser respondida com base nos conteúdos dados em sala). Você modelou este problema baseado em qual problema clássico?