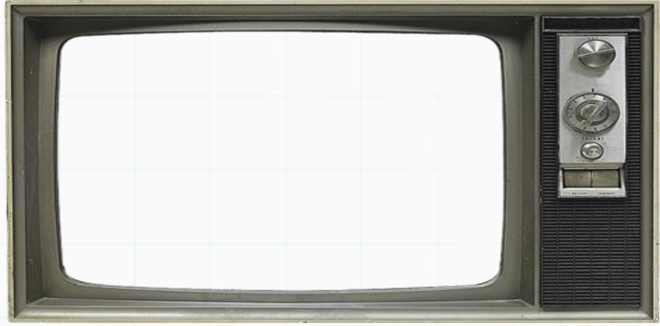
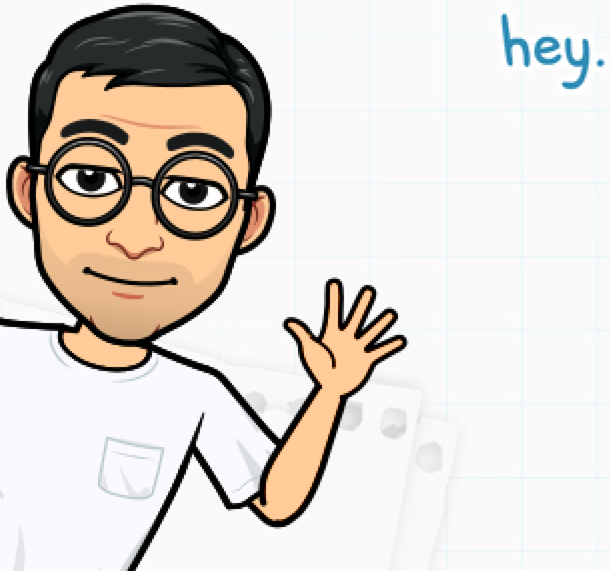


Programação Inteira

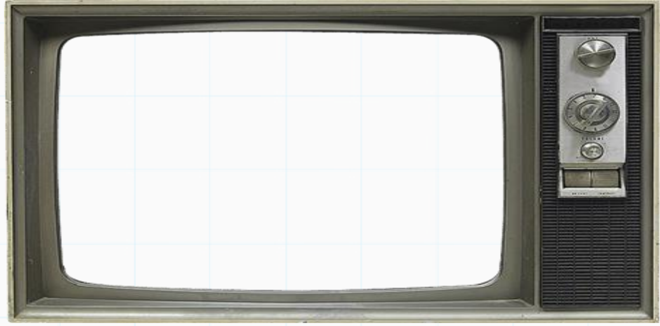
Professor : Yuri Frota

www.ic.uff.br/~yuri/pi.html

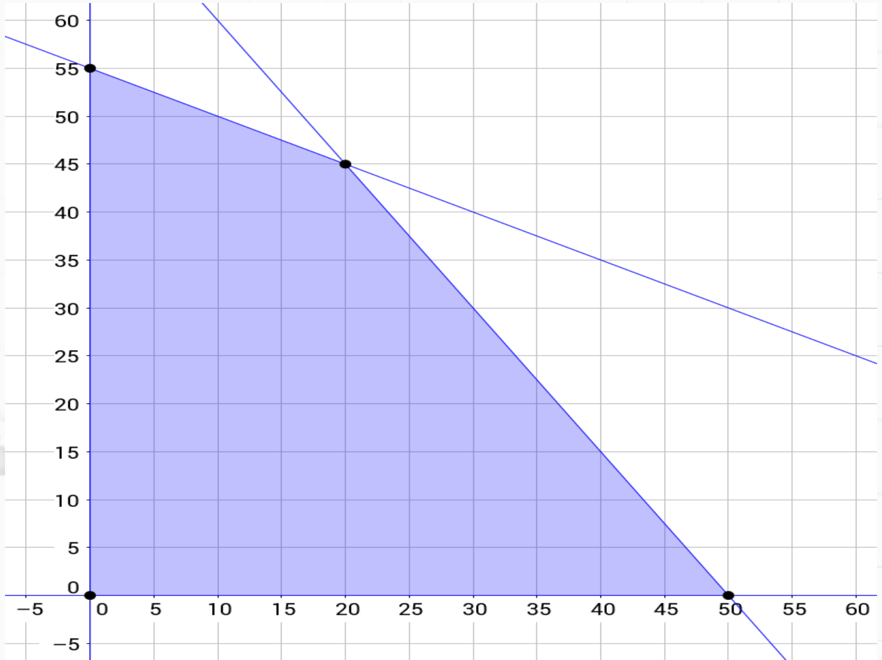
yuri@ic.uff.br



PPI



- O que é necessário para definir um Problema de Programação Matemática?
  - <u>Variáveis de Decisão</u>: São incógnitas a serem determinadas pela solução solução (determinam a dimensão)
  - <u>Restrições</u>: São as limitações de sua região viável
  - <u>Função Objetivo</u>: Função matemática que define a qualidade da solução em função das variáveis de decisão

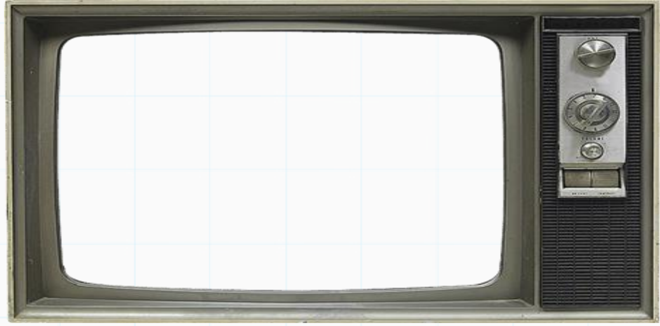


Formato geral:

$$\max \text{ ou } \min \; f(x)$$
$$g(x) \leq, =, \geq b_i$$
$$x \in X$$

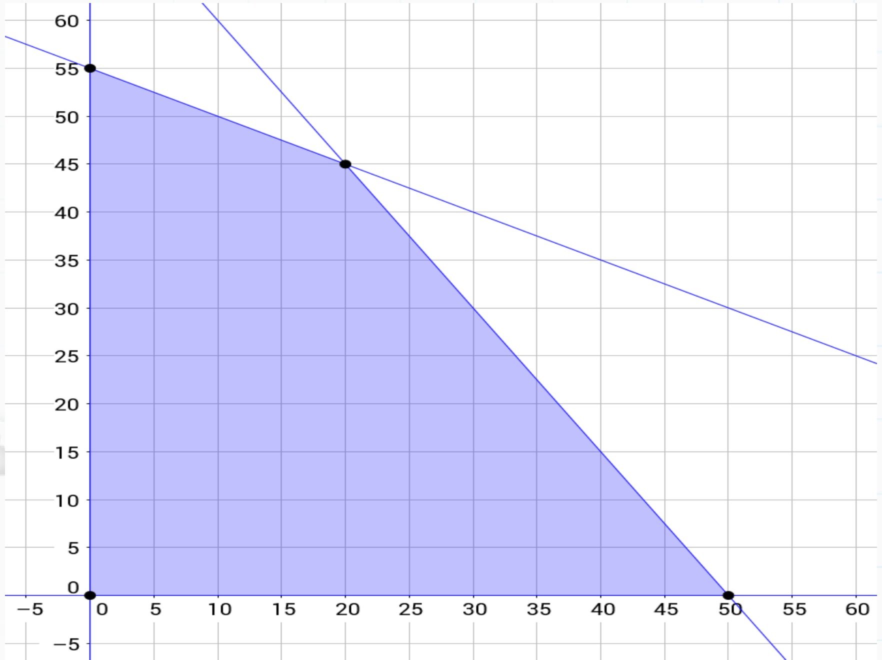
onde X pode ser um <u>conjunto discreto</u>

PPI



- O que é necessário para definir um Problema de Programação Matemática?
  - Variáveis de Decisão: São incógnitas a serem determinadas pela solução solução (determinam a dimensão)
  - Restrições: São as limitações de sua região viável
  - Função Objetivo: Função matemática que define a qualidade da solução em função das variáveis de decisão

Veja que PPI é um PPL com restrições de integralidade

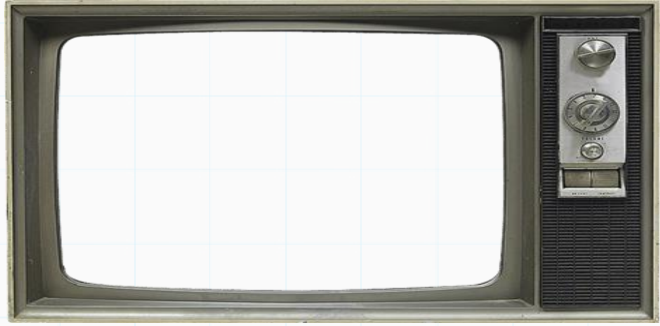


Formato geral:

$$\max \text{ ou } \min \ f(x)$$
$$g(x) \leq, =, \geq b_i$$
$$x \in X$$

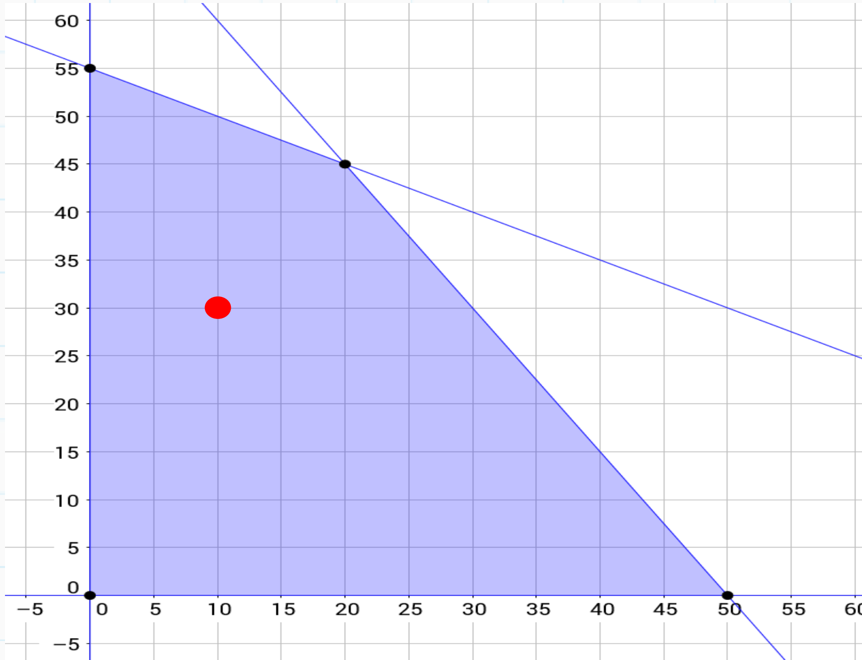
onde X pode ser um conjunto discreto

PPI

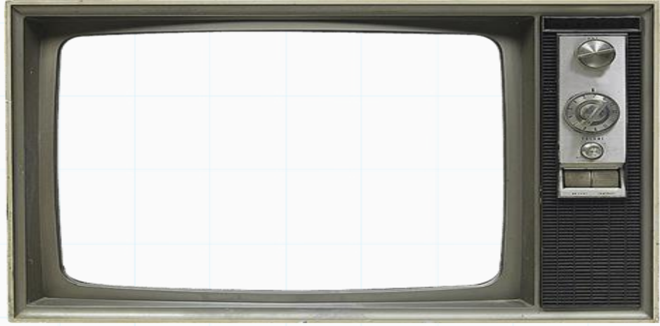


- Definições:
  - <u>solução</u>: conjunto de valores atribuídos a variáveis

$$
\begin{aligned}
\text{max ou min } & f(x) \\
& g(x) \le, =, \ge b_i \\
& x \in X
\end{aligned}
$$

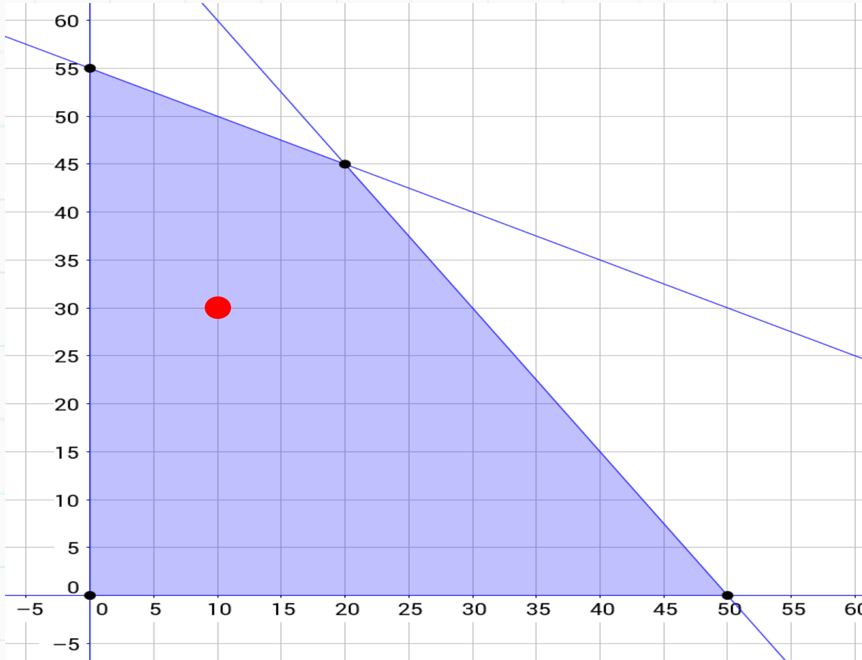


PPI

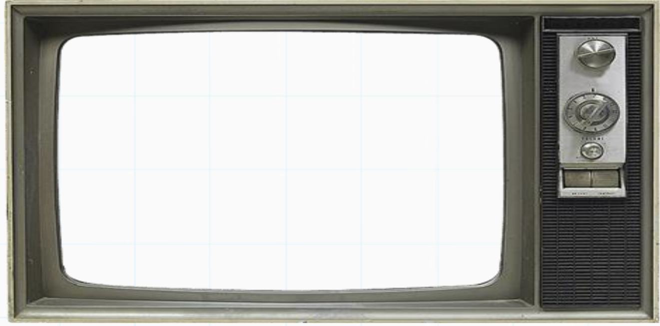


- Definições:
  - <u>solução</u>: conjunto de valores atribuídos a variáveis
  - <u>solução viável</u>: solução que satisfaz as restrições do problema

$$
\begin{aligned}
\max \ \text{ou} \ \min \ & f(x) \\
& g(x) \leq, =, \geq b_i \\
& x \in X
\end{aligned}
$$

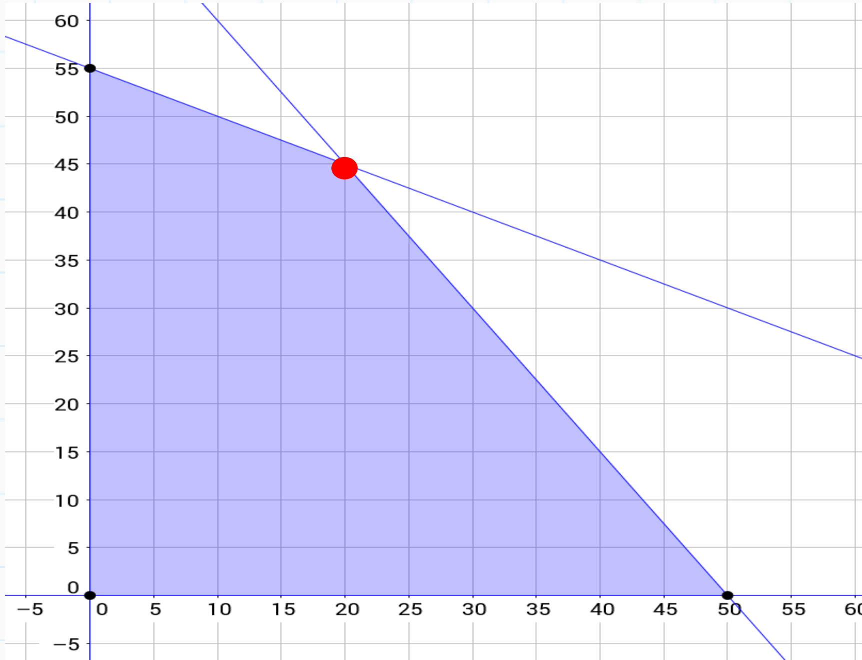


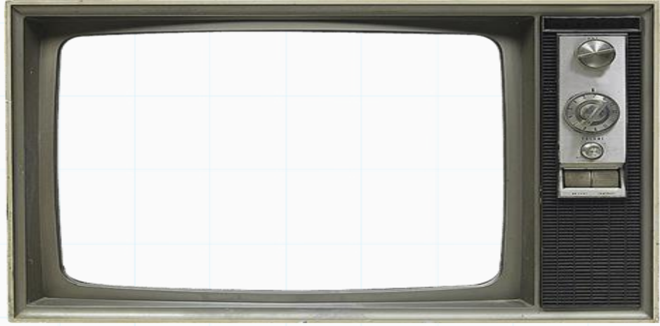
PPI



- Definições:
  - <u>solução</u>: conjunto de valores atribuídos a variáveis
  - <u>solução viável</u>: solução que satisfaz as restrições do problema
  - <u>solução ótima</u>: é uma solução viável que tem valor de função objetivo (fo) maior (ou menor) que qualquer outra solução viável

$$
\begin{aligned}
\text{max ou min } & f(x) \\
& g(x) \leq, =, \geq b_i \\
& x \in X
\end{aligned}
$$



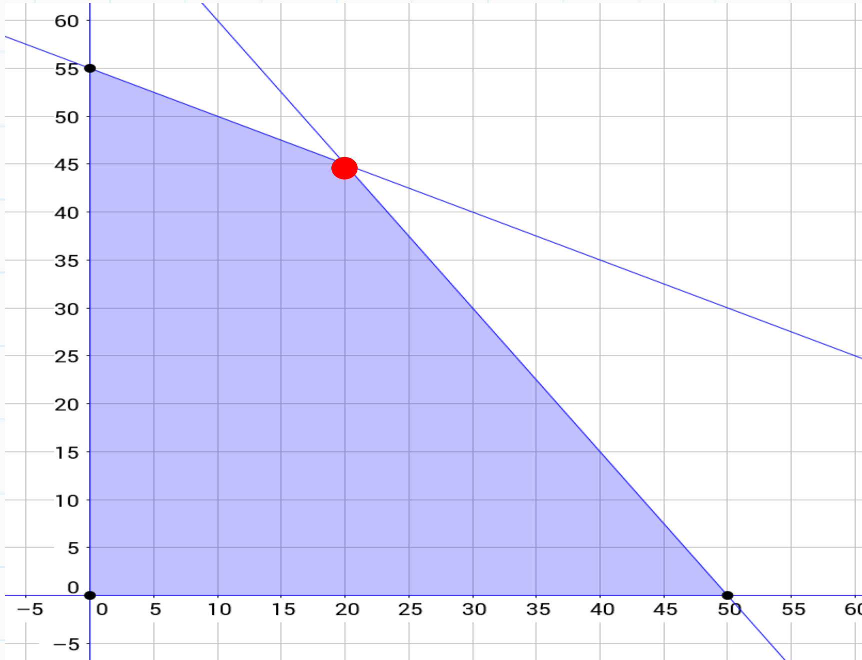


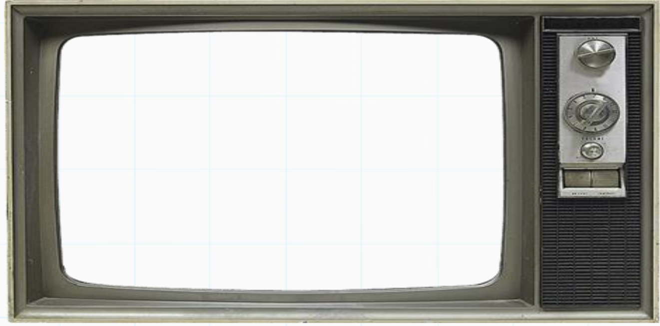
PPI

- Definições:
  - <u>solução</u>: conjunto de valores atribuídos a variáveis
  - <u>solução viável</u>: solução que satisfaz as restrições do problema
  - <u>solução ótima</u>: é uma solução viável que tem valor de função objetivo (fo) maior (ou menor) que qualquer outra solução viável

Neste curso vamos considerar que: $\boxed{X = \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}}$

$$\begin{aligned} \max \text{ ou } \min \ & f(x) \\ & g(x) \leq, =, \geq b_i \\ & x \in X \end{aligned}$$





PPI

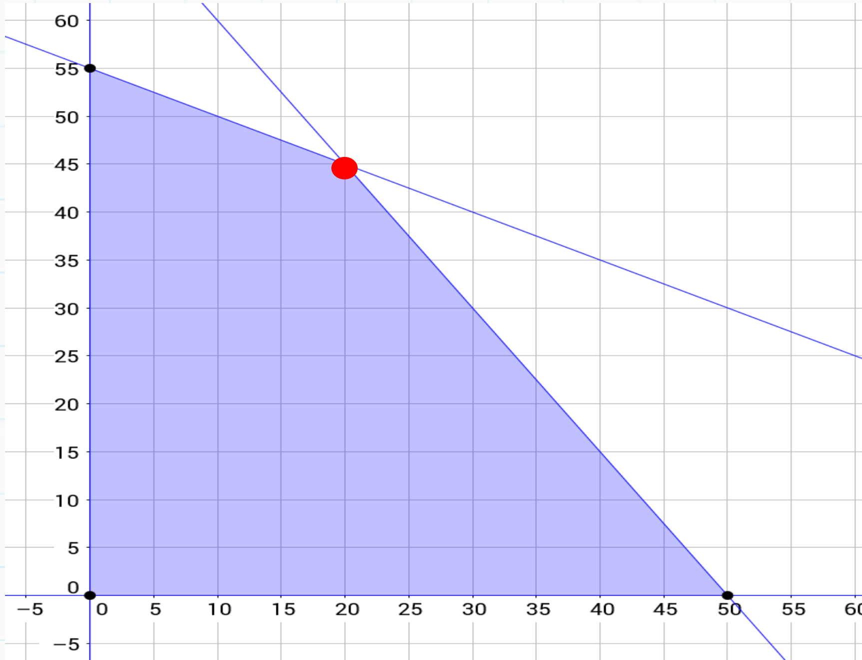
- Definições:
  - <u>solução</u>: conjunto de valores atribuídos a variáveis
  - <u>solução viável</u>: solução que satisfaz as restrições do problema
  - <u>solução ótima</u>: é uma solução viável que tem valor de função objetivo (fo) maior (ou menor) que qualquer outra solução viável

Neste curso vamos considerar que: $\boxed{X = \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}}$

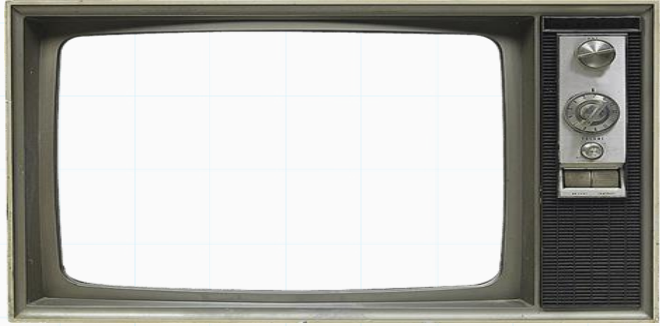
Estes problemas são chamados de Problemas de Programação Mista (PPM):

$$\max c^t x$$
$$Ax \leq b$$
$$x \in \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}$$

$$A \in \mathbb{R}^{m \times n}$$
$$b \in \mathbb{R}^m$$
$$c \in \mathbb{R}^n$$



PPI



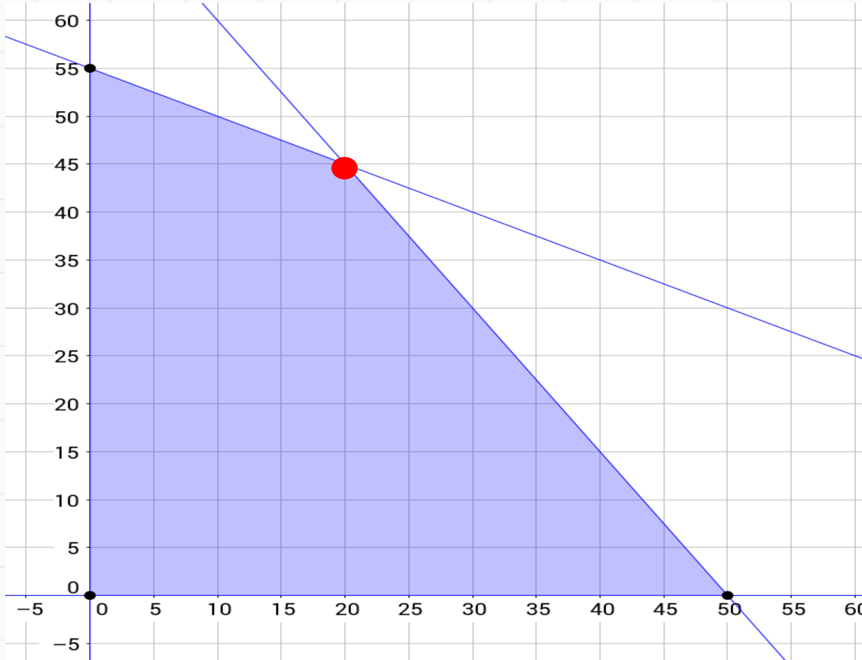
- Definições:
  - solução: conjunto de valores atribuídos a variáveis
  - solução viável: solução que satisfaz as restrições do problema
  - solução ótima: é uma solução viável que tem valor de função objetivo (fo) maior (ou menor) que qualquer outra solução viável

Neste curso vamos considerar que: $\boxed{X = \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}}$

Estes problemas são chamados de Problemas de Programação Mista (PPM):

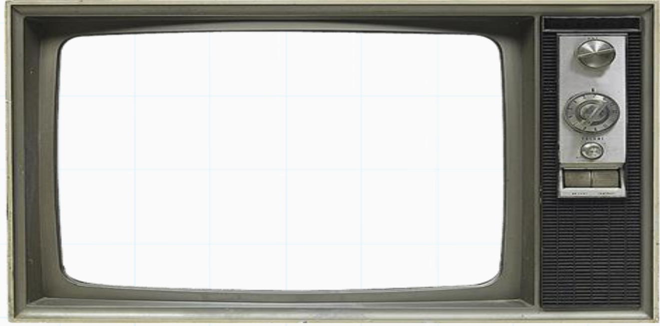
$$\max c^t x$$
$$Ax \leq b$$
$$x \in \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}$$

$$A \in \mathbb{R}^{m \times n}$$
$$b \in \mathbb{R}^m$$
$$c \in \mathbb{R}^n$$

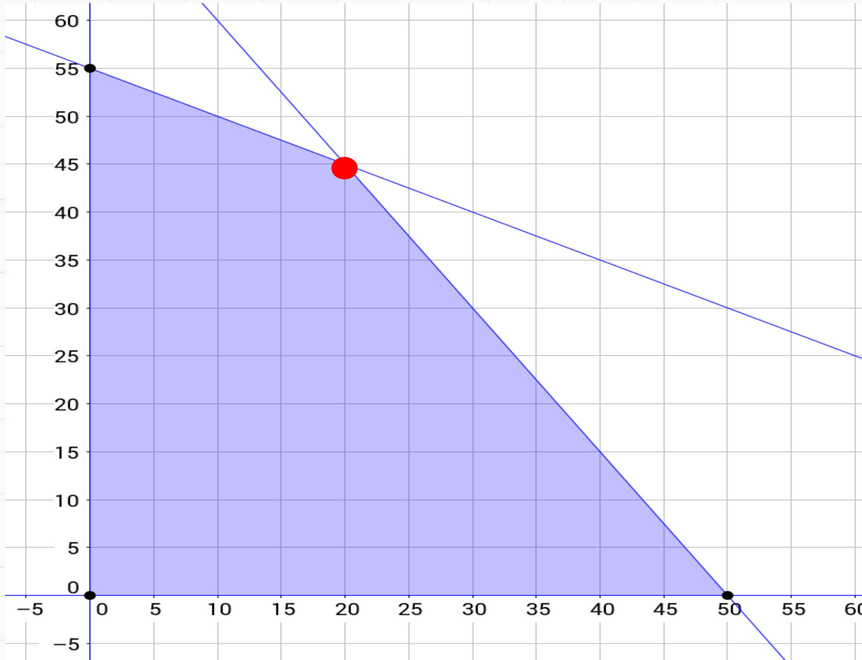


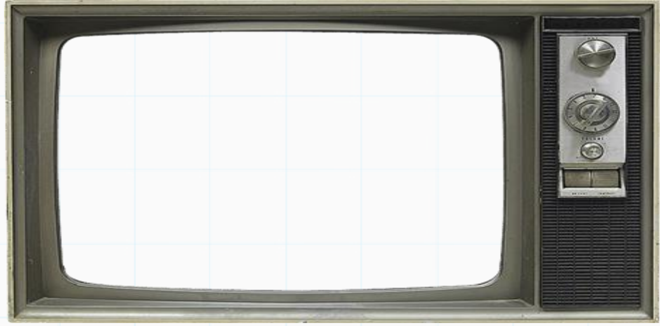
Se (p=n), chamamos de Problema de Programação Inteira (PPI)

PPI



- Caso Especial: quando as variáveis de um PPI assumem valores de decisão 0 e 1, i.e., chamamos de Problema de Programação Inteira Binária (PPIB) $\boxed{X = \mathbb{B}^n}$



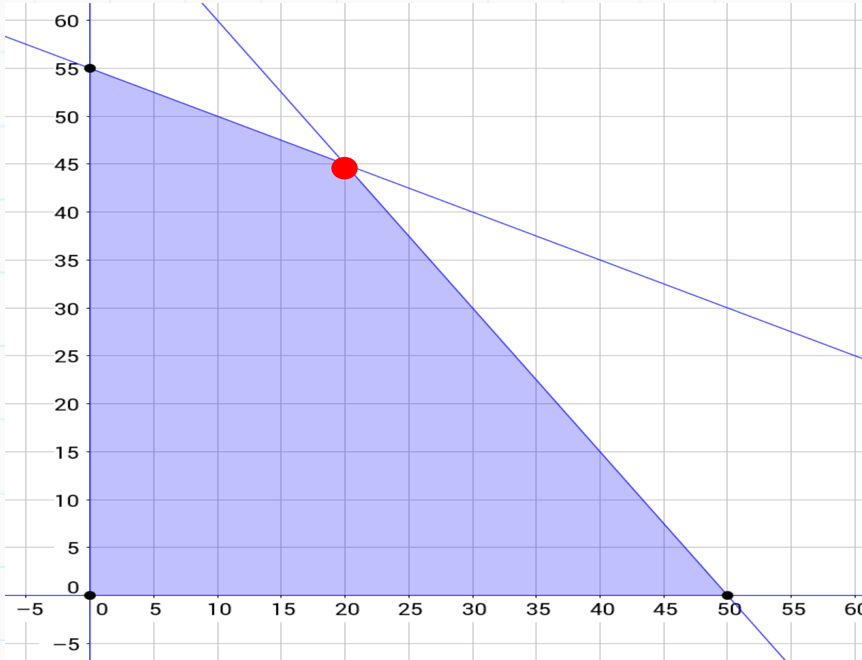


PPI

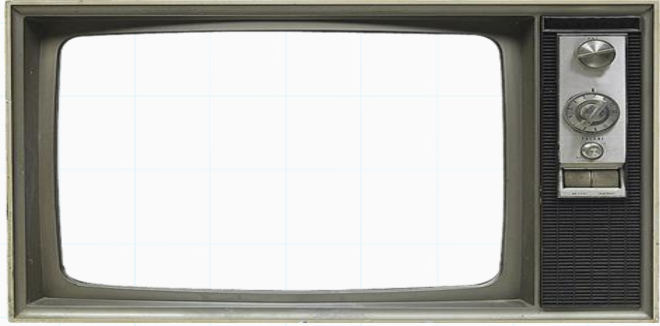
- Caso Especial: quando as variáveis de um PPI assumem valores de decisão 0 e 1, i.e., chamamos de Problema de Programação Inteira Binária (PPIB)

$$\boxed{X = \mathbb{B}^n}$$

Ex:
- Problema da Mochila
- Problema do caixeiro viajante clássico
- Qualquer problema puro de combinatória

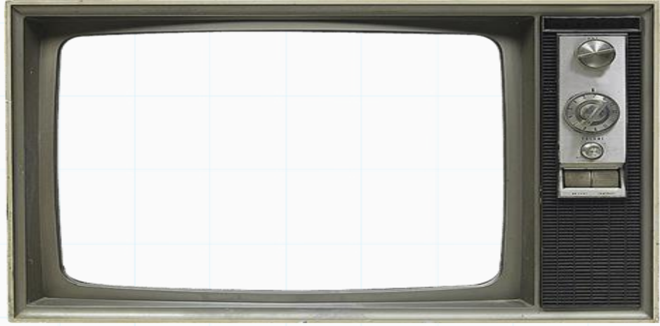


PPI



- <u>É difícil resolver um PPI</u> ?
    - Sabemos que resolver um PPL é polinomial

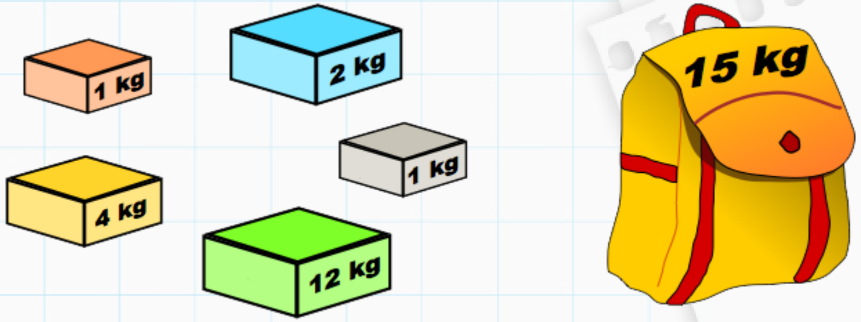
PPI



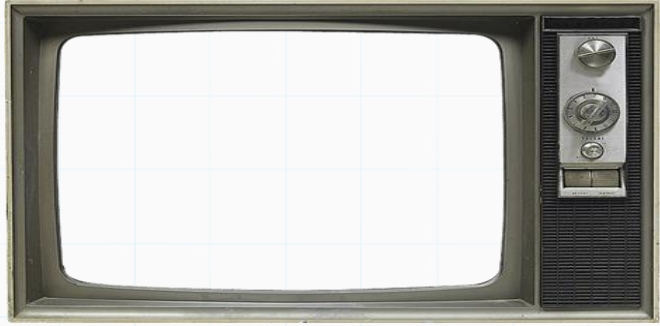
- <u>É difícil resolver um PPI</u> ?
  - Sabemos que resolver um PPL é polinomial

Vamos considerar um seguinte PPIB onde as variáveis podem assumir 0 ou 1. Seja o problema clássico da mochila:

Como é mesmo a formulação mesmo ?



PPI



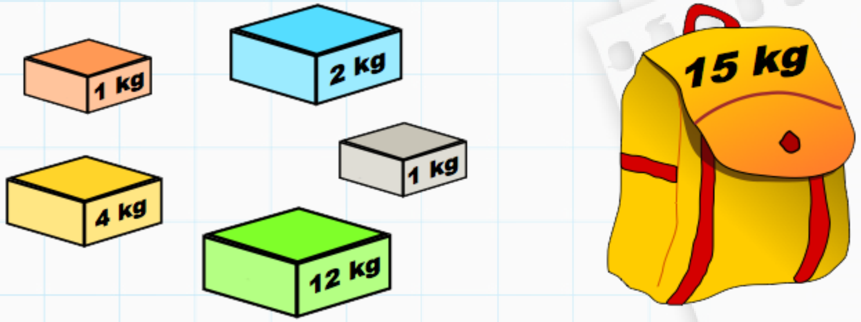
- <u>É difícil resolver um PPI</u> ?
    - Sabemos que resolver um PPL é polinomial

Vamos considerar um seguinte PPIB onde as variáveis podem assumir 0 ou 1. Seja o problema clássico da mochila:

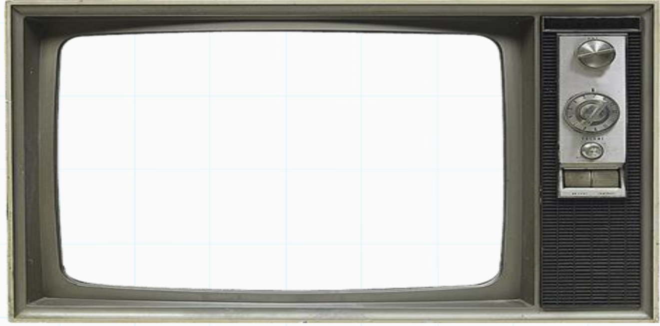
$$\max c^t x$$

$$\sum_{i \in I} p_i x_i \leq C$$

$$x \in \{0,1\}^n$$



PPI



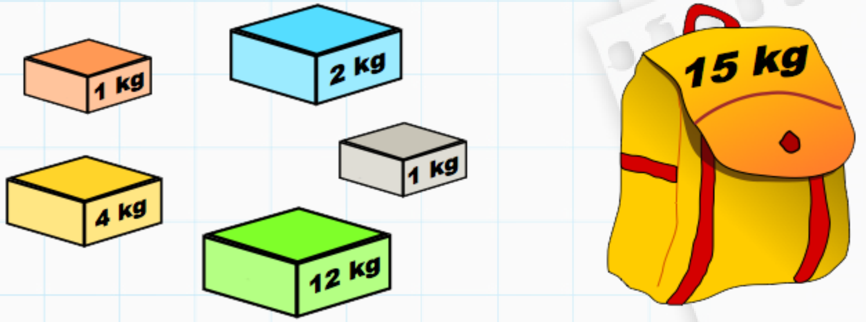
- É difícil resolver um PPI ?
  - Sabemos que resolver um PPL é polinomial

Vamos considerar um seguinte PPIB onde as variáveis podem assumir 0 ou 1. Seja o problema clássico da mochila:

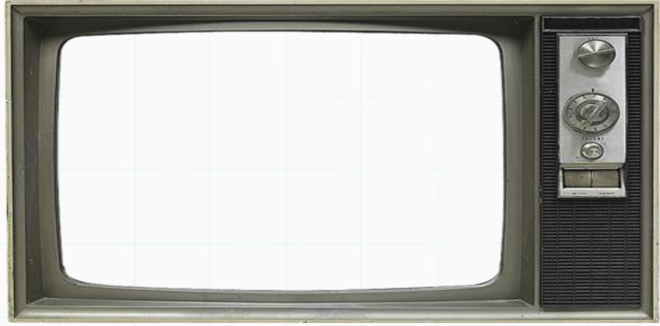
$$\max\ c^t x$$
$$\sum_{i \in I} p_i x_i \leq C$$
$$x \in \{0,1\}^n$$

Uma variável binária pode ser representada de forma contínua:

$$\boxed{x_i^2 = x_i,} \quad \forall i \in I$$



PPI



- É difícil resolver um PPI ?
    - Sabemos que resolver um PPL é polinomial

Vamos considerar um seguinte PPIB onde as variáveis podem assumir 0 ou 1. Seja o problema clássico da mochila:

$$\max\ c^t x$$
$$\sum_{i \in I} p_i x_i \leq C$$
$$x \in \{0,1\}^n$$

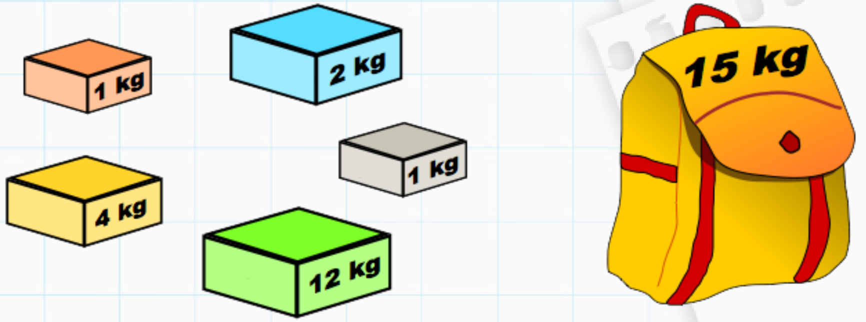
Uma variável binária pode ser representada de forma contínua:

$$\boxed{x_i^2 = x_i,} \quad \forall i \in I$$

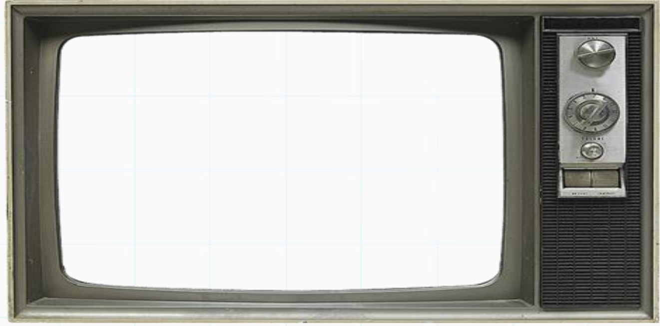
Logo: $\max\ c^t x$

$$\sum_{i \in I} p_i x_i \leq C$$
$$x_i^2 = x_i, \quad \forall i \in I$$
$$x \geq 0$$

É difícil ?



PPI



- <u>É difícil resolver um PPI</u> ?
  - Sabemos que resolver um PPL é polinomial

Vamos considerar um seguinte PPIB onde as variáveis podem assumir 0 ou 1. Seja o problema clássico da mochila:

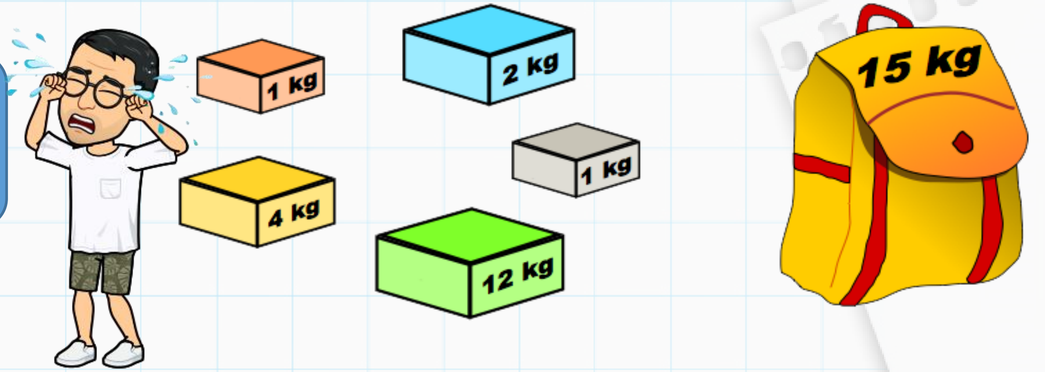
$$\max\ c^t x$$
$$\sum_{i \in I} p_i x_i \leq C$$
$$x \in \{0,1\}^n$$

Uma variável binária pode ser representada de forma contínua:

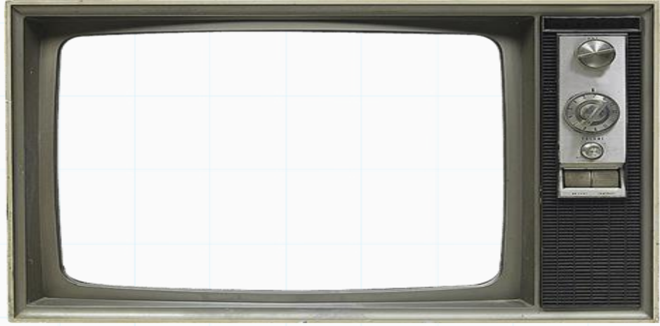
$$\boxed{x_i^2 = x_i,} \quad \forall i \in I$$

Logo: $$\max\ c^t x$$
$$\sum_{i \in I} p_i x_i \leq C$$
$$x_i^2 = x_i, \quad \forall i \in I$$
$$x \geq 0$$

problema de
programação quadrático
(NP-HARD)



PPI

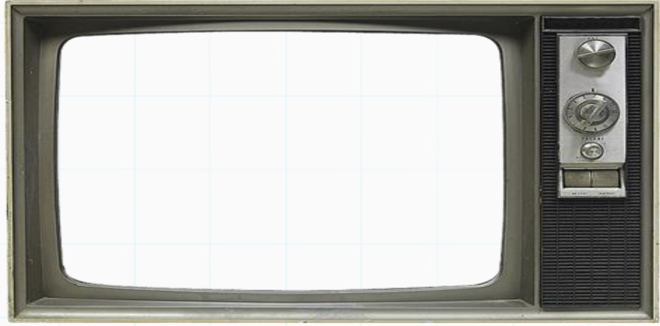


- <u>É difícil resolver um PPI</u> ?
  - Sabemos que resolver um PPL é polinomial

E se o PPI tivesse variáveis inteiras (invés de binárias) ainda seria redutível a um problema de programação quadrática ?

$$0 \leq x \leq d$$

PPI



- <u>É difícil resolver um PPI</u> ?
    - Sabemos que resolver um PPL é polinomial

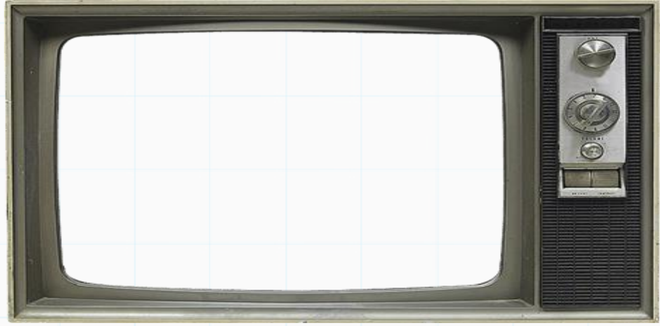
E se o PPI tivesse variáveis inteiras (invés de binárias) ainda seria reduzível a um problema de programação quadrática ?

$$0 \leq x \leq d$$

Toda variável inteira pode ser representada por uma soma de variáveis binárias:

Como?

PPI



- É difícil resolver um PPI ?
    - Sabemos que resolver um PPL é polinomial

E se o PPI tivesse variáveis inteiras (invés de binárias) ainda seria reduzível a um problema de programação quadrática ?

$$0 \leq x \leq d$$

Toda variável inteira pode ser representada por uma soma de variáveis binárias:

$$x = \sum_{i=0}^{d} i.s_i \qquad \sum_{i=0}^{d} s_i = 1$$

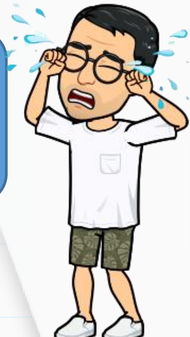
$$s_i^2 = s_i, \quad \forall i = 0, \ldots, d$$

Ex: x inteiro onde 0 <= x <= 3

$x = 0.s_0 + 1.s_1 + 2.s_2 + 3.s_3$

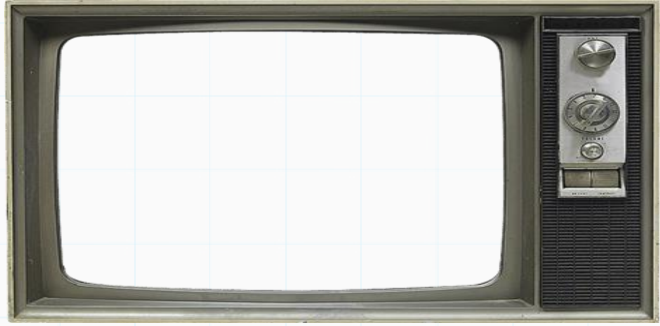
$s_0 + s_1 + s_2 + s_3 = 1$

problema de programação quadrático (NP-HARD)

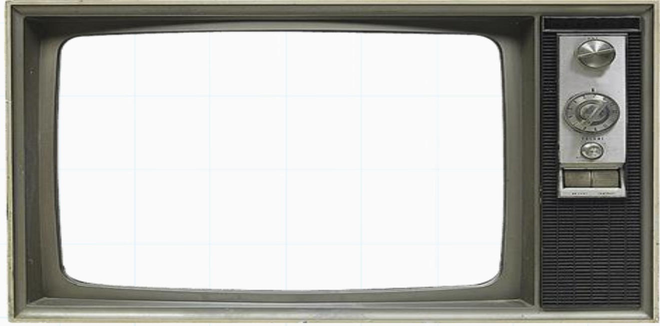


PPI

- <u>É difícil resolver um PPI</u> ?
  - Sabemos que resolver um PPL é polinomial
  - Não existe algoritmo polinomial para resolver o PPI



PPI



- É difícil resolver um PPI ?
    - Sabemos que resolver um PPL é polinomial
    - Não existe algoritmo polinomial para resolver o PPI

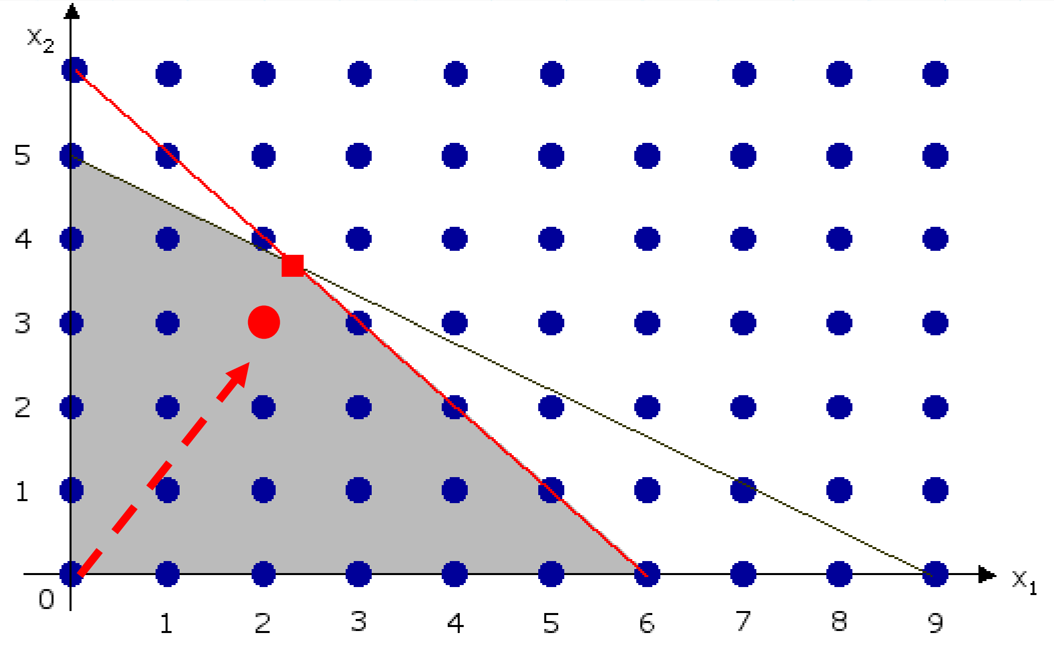
- E se abandonarmos as restrições de integralidade e resolvermos o PPI como um PPL (relaxação pois estamos eliminando restrições), encontraremos boas soluções ?

PPI

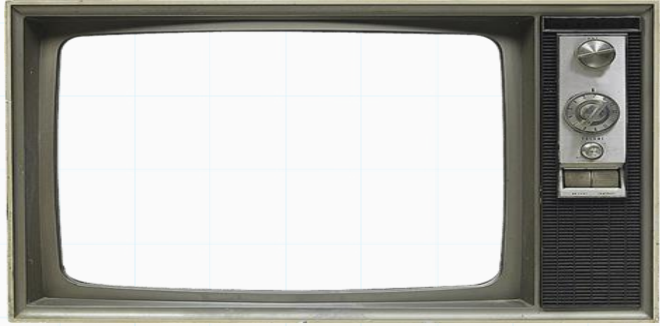
- É difícil resolver um PPI ?
    - Sabemos que resolver um PPL é polinomial
    - Não existe algoritmo polinomial para resolver o PPI

- E se abandonarmos as restrições de integralidade e resolvermos o PPI como um PPL (relaxação pois estamos eliminando restrições), encontraremos boas soluções ?

Não necessariamente, pode ser bem ruim!

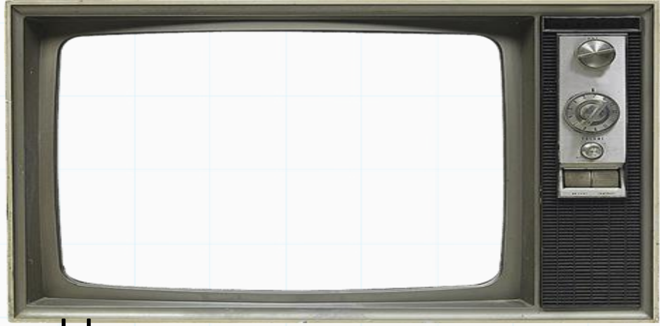


PPI



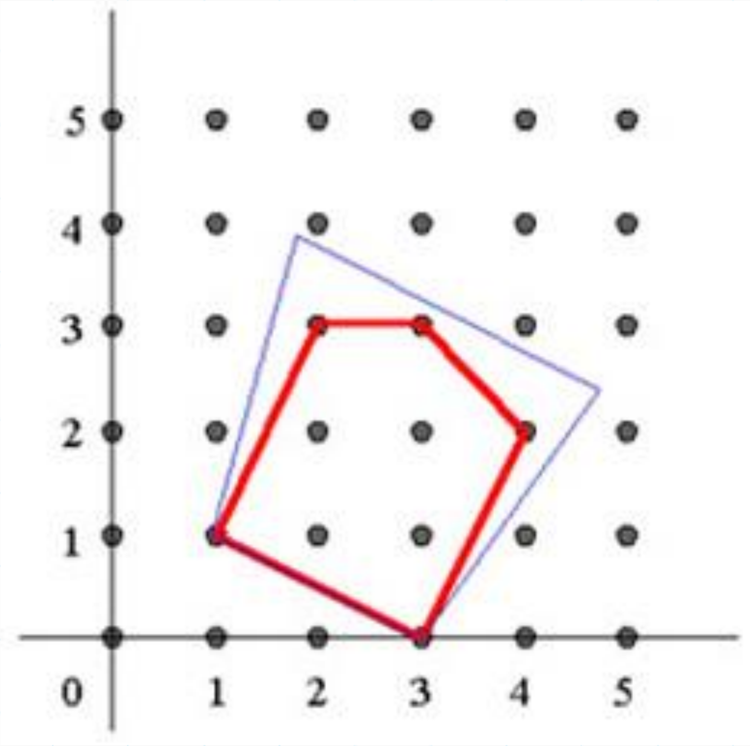
- Então não tem como resolver o PPI como um PPL ?

PPI



- Então não tem como resolver o PPI como um PPL ?

Embora o PPI seja não convexo, existe uma envoltória convexa que cobre os pontos viáveis do problema

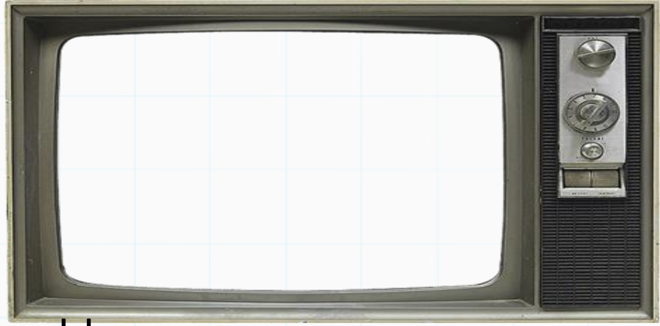


Ax <= b

x ∈ Z

E.C.

PPI

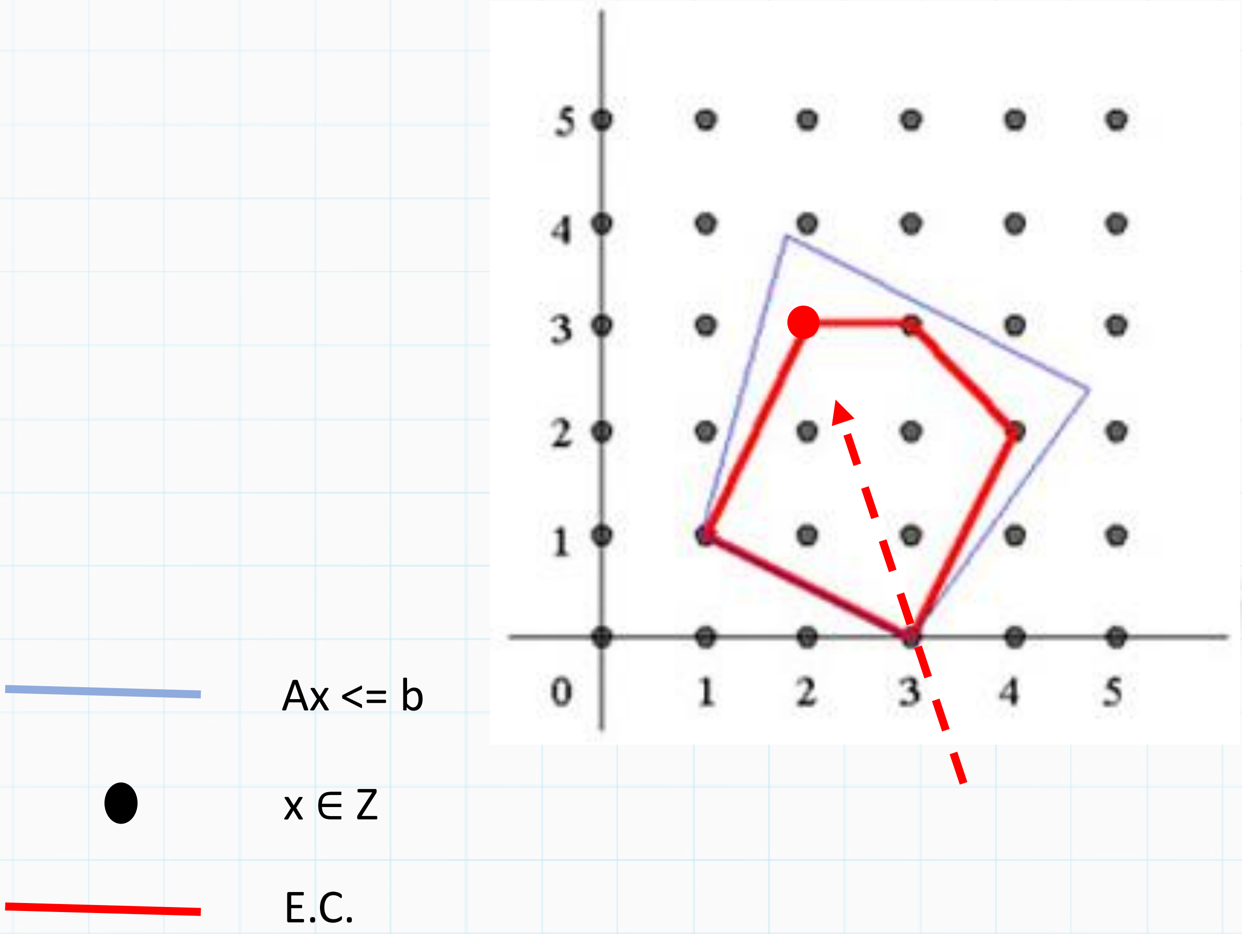


- Então não tem como resolver o PPI como um PPL ?

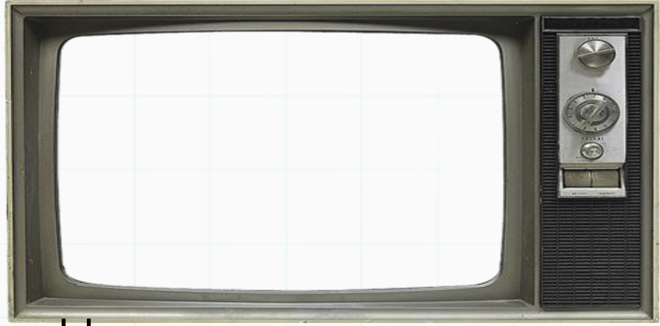
Embora o PPI seja não convexo, existe uma envoltória convexa que cobre os pontos viáveis do problema

Se tivermos essas restrições que definem a envoltória convexa, poderíamos resolver o PPI como um PPL de forma polinomial !

Porque ?



PPI

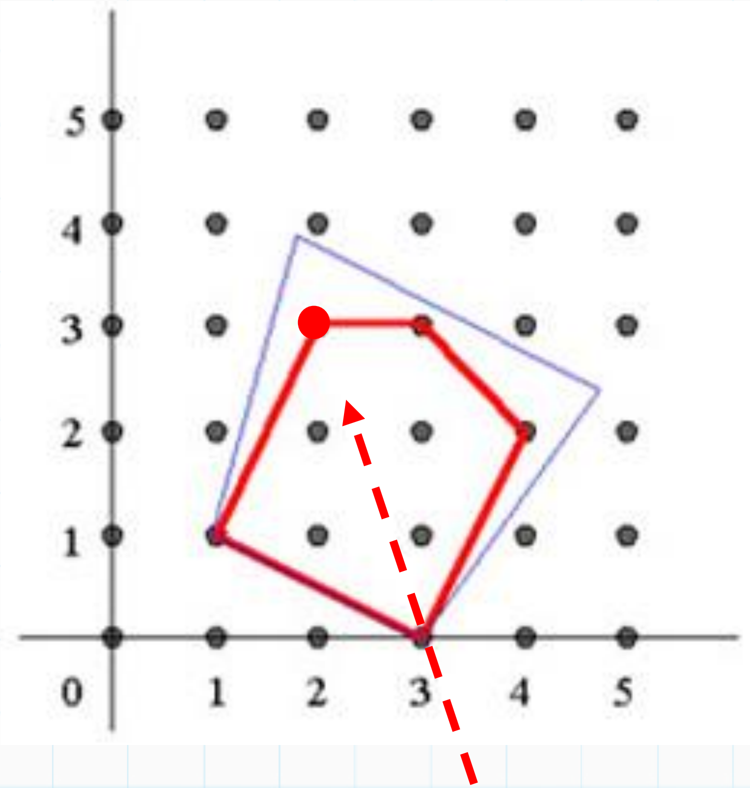


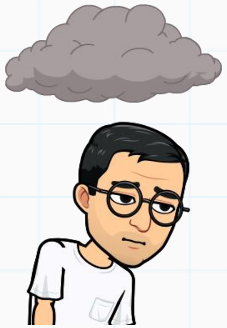
- Então não tem como resolver o PPI como um PPL ?

Embora o PPI seja não convexo, existe uma envoltória convexa que cobre os pontos viáveis do problema

Se tivermos essas restrições que definem a envoltória convexa, poderíamos resolver o PPI como um PPL de forma polinomial !

O problema é que dificilmente a envoltória convexa é conhecida, e quando ela é, é geralmente muito grande para ser expressa !



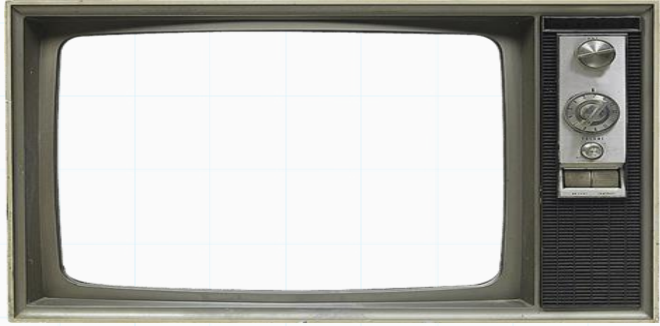


Ax <= b

x ∈ Z

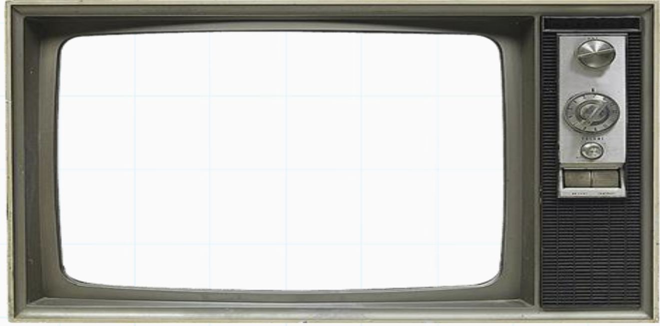
E.C.

Modelando com variáveis Inteiras



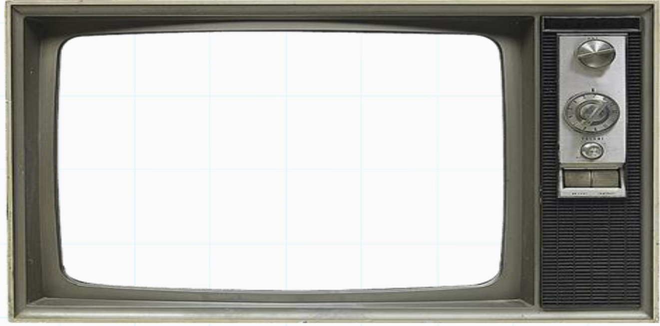
- Vamos agora formalizar algumas técnicas de modelagem de programação inteira que representam decisões e situações a serem modeladas.

Modelando com variáveis Inteiras



- Como usar variáveis inteiras para impor que uma ação x só pode ser feita se a outra ação y for feita também ?

Modelando com variáveis Inteiras

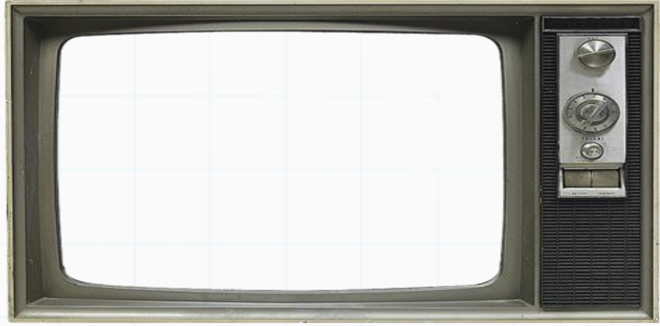


- Como usar variáveis inteiras para impor que uma ação x só pode ser feita se a outra ação y for feita também ?

A restrição x <= y diz que “somente faça x se y também for feito”

x e y binários

Modelando com variáveis Inteiras

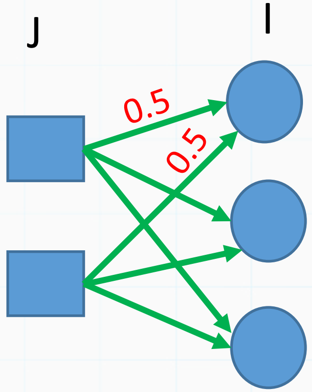


- Como usar variáveis inteiras para impor que uma <u>ação x</u> só pode ser feita se a outra <u>ação y</u> for feita também ?

A restrição x <= y diz que "somente faça x se y também for feito"

x e y binários

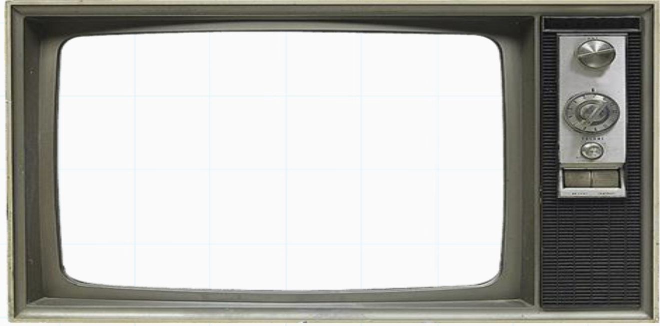
Ex: <u>Problema de Localização não Capacitado</u> (PLF)
Conjunto de facilidades J, Conjunto de Clientes I. Que facilidades tem que ser construídas para atender a demanda ?



c_j = custo de construção da facilidade j
f_{ji} = custo da fração de atendimento de j para i (atendimento unitário)

Variáveis:

Modelando com variáveis Inteiras

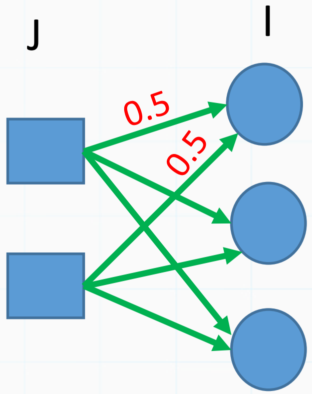


- Como usar variáveis inteiras para impor que uma <u>ação x</u> só pode ser feita se a outra <u>ação y</u> for feita também ?

A restrição x <= y diz que “somente faça x se y também for feito”

x e y binários

Ex: <u>Problema de Localização não Capacitado</u> (PLF)
Conjunto de facilidades J, Conjunto de Clientes I. Que facilidades tem que ser construídas para atender a demanda ?



c_j = custo de construção da facilidade j
f_{ji} = custo da fração de atendimento de j para i (atendimento unitário)

Variáveis:

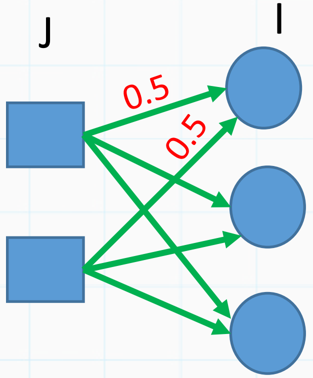
$$x_j \quad \begin{array}{l} 1, \text{se a facilidade j é construída} \\ 0, \text{caso contrário} \end{array}$$

$$y_{ji} \quad \text{fração da demanda do cliente i satisfeita pela a facilidade j}$$

Modelando com variáveis Inteiras

Ex: Problema de Localização não Capacitado (PLF)
Conjunto de facilidades J, Conjunto de Clientes I. Que facilidades tem que ser construídas para atender a demanda ?

c_j = custo de construção da facilidade j
f_{ji} = custo da fração de atendimento de j para i



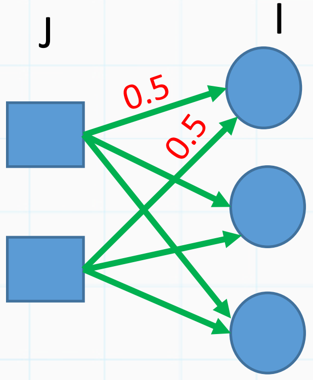
todo cliente tem que ser atendido:

x_j 1, se a facilidade j é construída
0, caso contrário

y_{ji} fração da demanda do cliente i satisfeita pela a facilidade j

Modelando com variáveis Inteiras

Ex: <u>Problema de Localização não Capacitado</u> (PLF)
Conjunto de facilidades J, Conjunto de Clientes I. Que facilidades tem que ser construídas para atender a demanda ?



c_j = custo de construção da facilidade j
f_{ji} = custo da fração de atendimento de j para i

todo cliente tem que ser atendido:

$$\sum_{j \in J} y_{ji} = 1, \ \forall i \in I$$

x_j $\quad$ 1, se a facilidade j é construída
$\quad\quad$ 0, caso contrário
y_{ji} $\quad$ fração da demanda do cliente i satisfeita pela a facilidade j

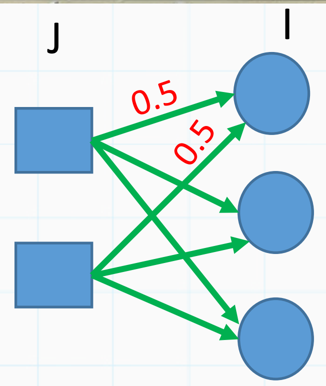
Modelando com variáveis Inteiras

Ex: Problema de Localização não Capacitado (PLF)

Conjunto de facilidades J, Conjunto de Clientes I. Que facilidades tem que ser construídas para atender a demanda ?

c_j = custo de construção da facilidade j

f_{ji} = custo da fração de atendimento de j para i



todo cliente tem que ser atendido:

$$\sum_{j \in J} y_{ji} = 1, \ \forall i \in I$$

se houver atendimento, a facilidade tem que ser construída:

x_j 1, se a facilidade j é construída

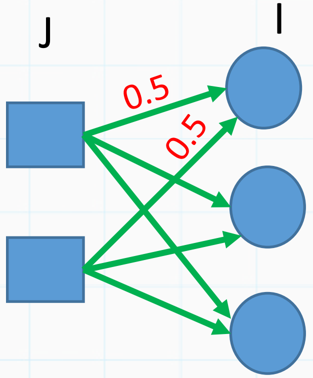
0, caso contrário

y_{ji} fração da demanda do cliente i satisfeita pela a facilidade j

Modelando com variáveis Inteiras

Ex: Problema de Localização não Capacitado (PLF)

Conjunto de facilidades J, Conjunto de Clientes I. Que facilidades tem que ser construídas para atender a demanda ?



c_j = custo de construção da facilidade j

f_{ji} = custo da fração de atendimento de j para i

todo cliente tem que ser atendido:

$$\sum_{j \in J} y_{ji} = 1, \ \forall i \in I$$

se houver atendimento, a facilidade tem que ser construída:

$$\boxed{y_{ji} \leq x_j} \forall j \in J, \forall i \in I$$

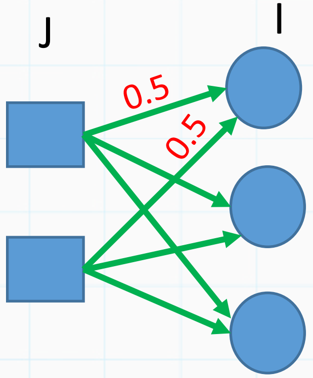
x_j 1, se a facilidade j é construída

0, caso contrário

y_{ji} fração da demanda do cliente i satisfeita pela a facilidade j

Modelando com variáveis Inteiras

Ex: Problema de Localização não Capacitado (PLF)
Conjunto de facilidades J, Conjunto de Clientes I. Que facilidades tem que ser construídas para atender a demanda ?



c_j = custo de construção da facilidade j
f_{ji} = custo da fração de atendimento de j para i

todo cliente tem que ser atendido:

$$\sum_{j \in J} y_{ji} = 1, \ \forall i \in I$$

se houver atendimento, a facilidade tem que ser construída:

$$\boxed{y_{ji} \leq x_j} \forall j \in J, \forall i \in I$$

restrições de não negatividade e integralidade:

$$x \in \mathbb{B}^{|J|}$$

$$y_{ij} \geq 0, \ \forall j \in J, \forall i \in I$$

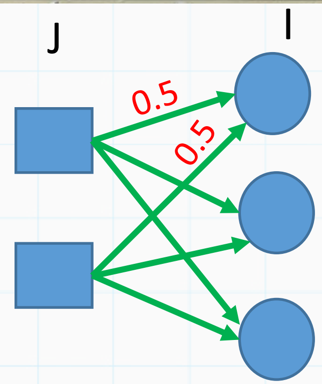
x_j 1, se a facilidade j é construída
0, caso contrário

y_{ji} fração da demanda do cliente i satisfeita pela a facilidade j

Modelando com variáveis Inteiras

Ex: <u>Problema de Localização não Capacitado</u> (PLF)

Conjunto de facilidades J, Conjunto de Clientes I. Que facilidades tem que ser construídas para atender a demanda ?



c_j = custo de construção da facilidade j

f_{ji} = custo da fração de atendimento de j para i

todo cliente tem que ser atendido:

$$\sum_{j \in J} y_{ji} = 1, \ \forall i \in I$$

se houver atendimento, a facilidade tem que ser construída:

$$\boxed{y_{ji} \leq x_j} \forall j \in J, \forall i \in I$$

restrições de não negatividade e integralidade:

$$x \in \mathbb{B}^{|J|}$$

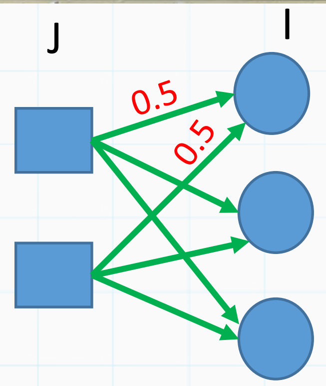
$$y_{ij} \geq 0, \ \forall j \in J, \forall i \in I$$

função objetivo:

Modelando com variáveis Inteiras

Ex: <u>Problema de Localização não Capacitado </u>(PLF)

Conjunto de facilidades J, Conjunto de Clientes I. Que facilidades tem que ser construídas para atender a demanda ?



c_j = custo de construção da facilidade j

f_{ji} = custo da fração de atendimento de j para i

todo cliente tem que ser atendido:

$$\sum_{j\in J} y_{ji} = 1, \ \forall i \in I$$

se houver atendimento, a facilidade tem que ser construída:

$$\boxed{y_{ji} \leq x_j} \forall j \in J, \forall i \in I$$

restrições de não negatividade e integralidade:

$$x \in \mathbb{B}^{|J|}$$

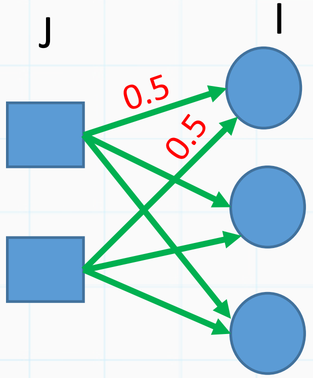
$$y_{ij} \geq 0, \ \forall j \in J, \forall i \in I$$

função objetivo:

$$\max \sum_{j\in J} c_j x_j + \sum_{j\in J}\sum_{i\in I} f_{ji} y_{ji}$$

Modelando com variáveis Inteiras

Ex: <u>Problema de Localização não Capacitado</u> (PLF)
Conjunto de facilidades J, Conjunto de Clientes I. Que facilidades tem que ser construídas para atender a demanda ? c_j = custo de construção da facilidade j e f_{ji} = custo da fração de atendimento de j para i



Modelo completo:

x_j 1, se a facilidade j é construída
0, caso contrário

y_{ji} fração da demanda do cliente i satisfeita pela a facilidade j

$$\max \sum_{j \in J} c_j x_j + \sum_{j \in J} \sum_{i \in I} f_{ji} y_{ji}$$

$$\sum_{j \in J} y_{ji} = 1, \ \forall i \in I$$

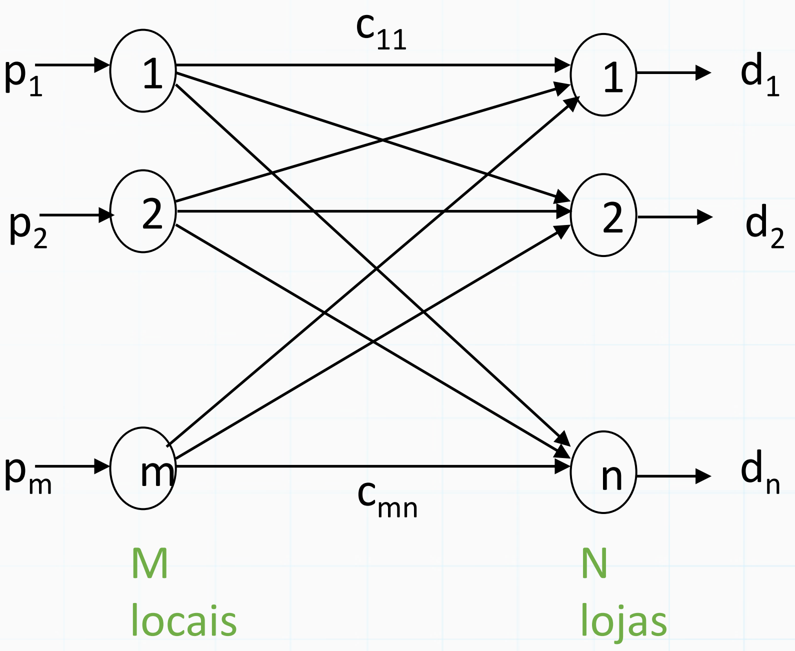
$$y_{ji} \leq x_j \ \forall j \in J, \forall i \in I$$

$$x \in \mathbb{B}^{|J|}$$

$$y_{ij} \geq 0, \ \forall j \in J, \forall i \in I$$

- 1) Problema de Transporte com Localização:

Exercícios



- Uma empresa pode produzir um determinado produto em m possíveis fábricas distintas (não construídas) e afastadas, para atender a demanda de n lojas diferentes (conjunto M e N).
- A capacidade de produção da fábrica $i \in M$ (se construída) é no máximo igual a p_i. A demanda da loja $j \in N$ é igual a d_j.
- Para a empresa realizar um atendimento a partir de uma fábrica $i \in M$, ela tem que ser construída e deve pagar um preço (custo fixo) de f_i.
- Sabendo-se que o custo de envio de uma unidade do produto da fábrica i para a cidade j é igual a c_{ij} (custo variável), determinar quais fábricas a serem construídas e a quantidade que deve ser enviada de cada fábrica para cada cidade, de modo a minimizar os custos de transporte e de construção desta empresa.

1) Variáveis:

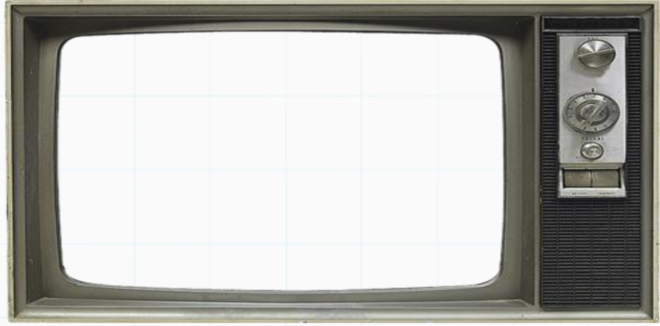
   Yij -> quantidade de produto enviada do local i para loja j (continua)

   Xi -> fabrica é construída no local i

2) Restrições:

   - Para cada loja, a demanda dela tem que ser atendida
   - Para cada fábrica, ela não pode produzir/enviar mais que sua capacidade
   - Para cada fábrica, se ela produz/envia então ela tem que estar construída
   - Não negatividade

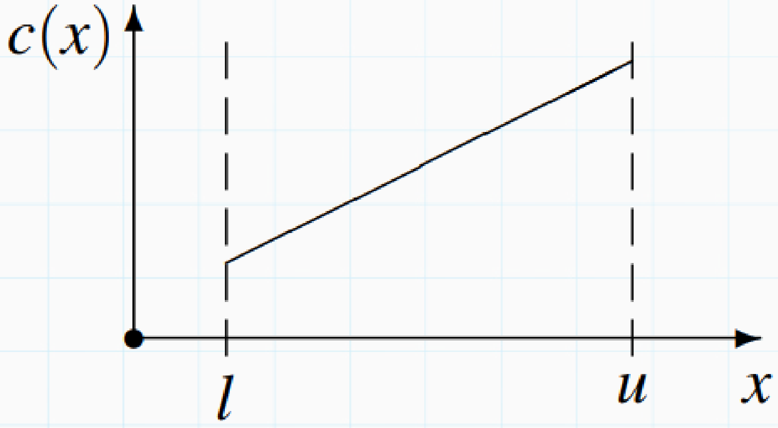
1) F.O. : minimizar custos de construção e transporte



Modelando com variáveis Inteiras

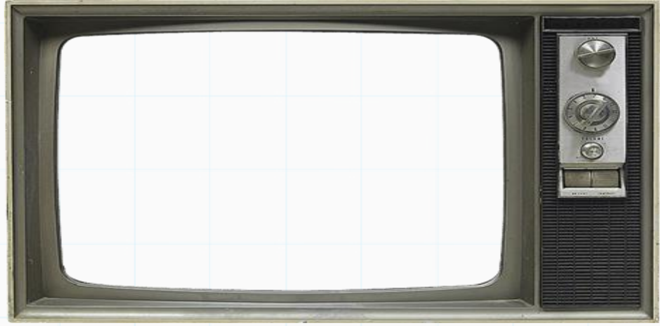
De uma forma geral usamos variáveis inteiras (binárias) para representar esse “custo fixo” de produção. Vamos assumir que o custo de se produzir x unidades de um produto é dado por:

$$c(x) \stackrel{\text{def}}{=} \begin{cases} f + px & \text{if } 0 < l \leq x \leq u, \\ 0 & \text{if } x = 0, \end{cases}$$



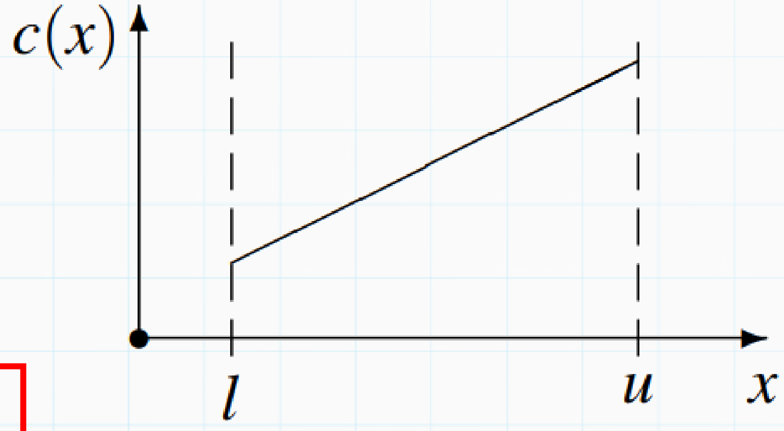
onde f é o custo fixo e p o custo por unidade. E l e u são o mínimo e máximo que podem ser produzidos, se houver produção.

Modelando com variáveis Inteiras



De uma forma geral usamos variáveis inteiras (binárias) para representar esse “custo fixo” de produção. Vamos assumir que o custo de se produzir x unidades de um produto é dado por:

$$c(x) \stackrel{\text{def}}{=} \begin{cases} f + px & \text{if } 0 < l \leq x \leq u, \\ 0 & \text{if } x = 0, \end{cases}$$

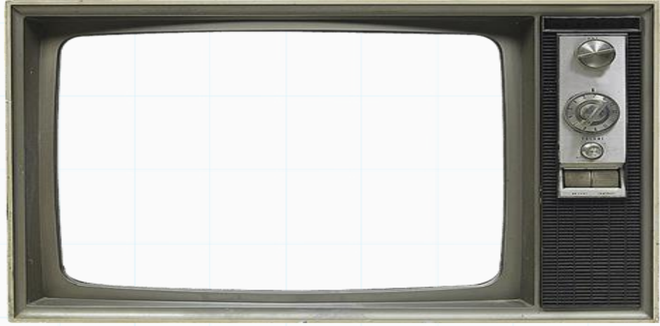


onde f é o custo fixo e p o custo por unidade. E l e u são o mínimo e máximo que podem ser produzidos, se houver produção.

O custo (que não é contínuo) pode ser modelado usando variáveis binárias y:

$$ly \leq x \leq uy.$$

Modelando com variáveis Inteiras



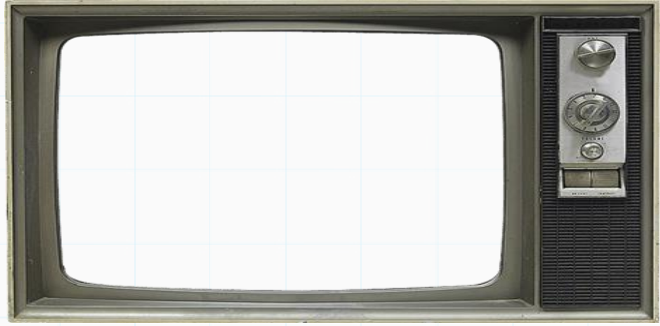
<u>Restrições Disjuntas:</u> Suponha duas restrições

$$a^T x >= b \quad \text{e} \quad c^T x >= d$$

com coeficientes não negativos, e queremos dizer que pelo menos uma delas tem que estar ativa.

Como ?

Modelando com variáveis Inteiras



<u>Restrições Disjuntas:</u> Suponha duas restrições

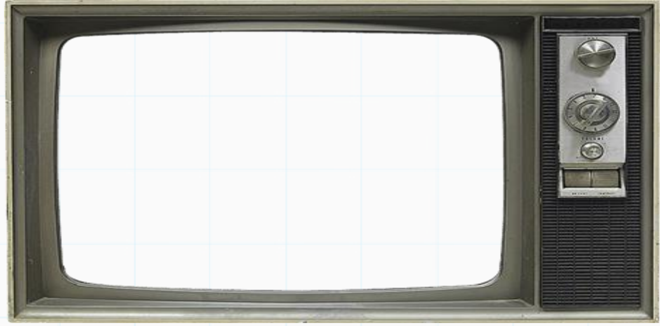
$$a^T x >= b \quad \text{e} \quad c^T x >= d$$

com coeficientes não negativos, e queremos dizer que pelo menos uma delas tem que estar ativa.

Cria variável binária y $\in \{0,1\}$ e:

$$a^T x >= y.b$$
$$c^T x >= (1-y).d$$

Modelando com variáveis Inteiras



<u>Restrições Disjuntas:</u> Suponha duas restrições

$$a^T x >= b \quad e \quad c^T x >= d$$

com coeficientes não negativos, e queremos dizer que pelo menos uma delas tem que estar ativa.

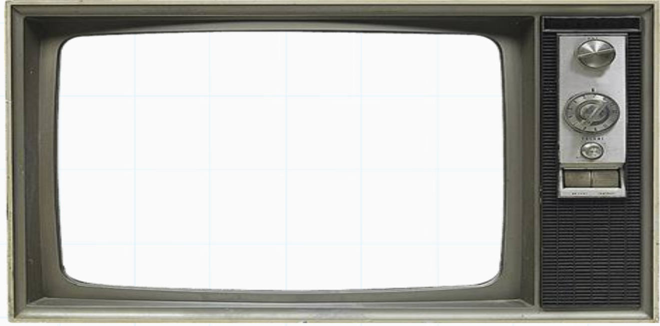
Cria variável binária y ∈ $\{0,1\}$ e:

$$a^T x >= y.b$$
$$c^T x >= (1-y).d$$

De forma geral podemos impor que pelo menos <u>k</u> restrições $a_i^T x >= b_i$ estejam ativas para i=1...<u>m</u> restrições:

Como ?

Modelando com variáveis Inteiras



<u>Restrições Disjuntas:</u> Suponha duas restrições

$$a^T x >= b \quad \text{e} \quad c^T x >= d$$

com coeficientes não negativos, e queremos dizer que pelo menos uma delas tem que estar ativa.

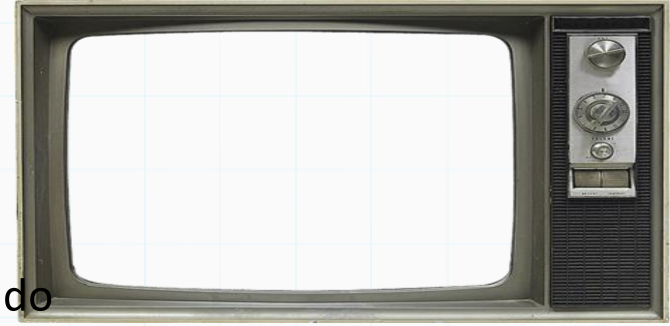
Cria variável binária $y \in \{0,1\}$ e:

$$a^T x >= y.b$$
$$c^T x >= (1-y).d$$

De forma geral podemos impor que pelo menos <u>k</u> restrições $a_i^T x >= b_i$ estejam ativas para i=1...<u>m</u> restrições:

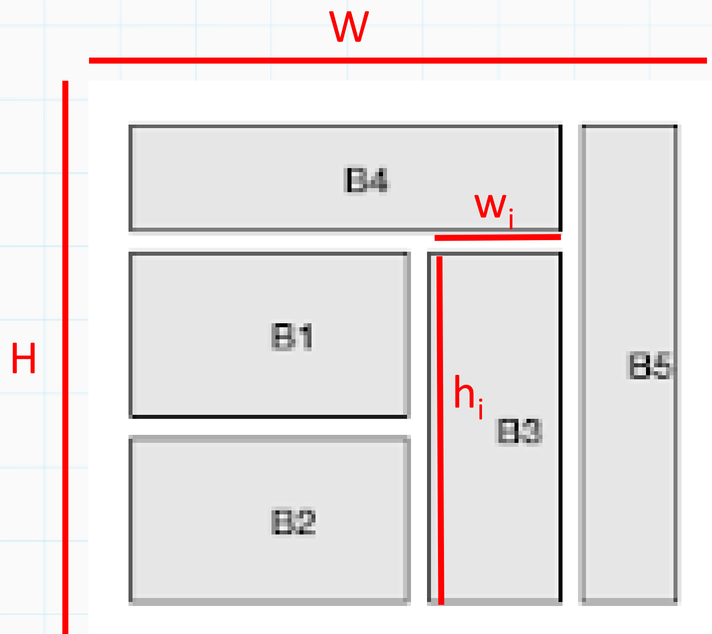
$$a_i^T x >= y_i.b_i, \qquad \forall i=1...m$$
$$\sum_{i=1}^{m} y_i \geq k$$
$$y_i \in \{0,1\} \qquad \forall i=1...m$$

Modelando com variáveis Inteiras

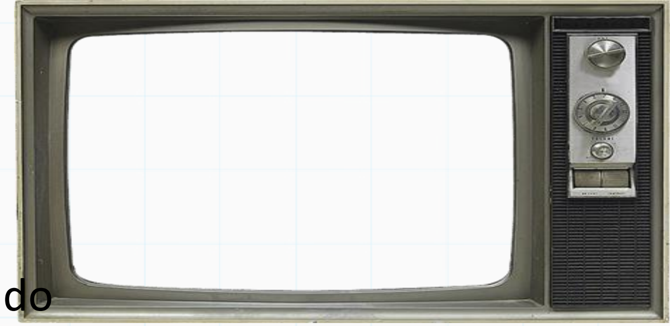


<u>Ex: Problema de colocação de cerâmica:</u> Em um plano retângulo de largura W e altura H, n cerâmicas retangulares devem ser colocadas, cada uma com largura w_i e altura h_i. Esta versão do problema não tem objetivo, buscamos apenas uma solução viável.

As cerâmicas não giram



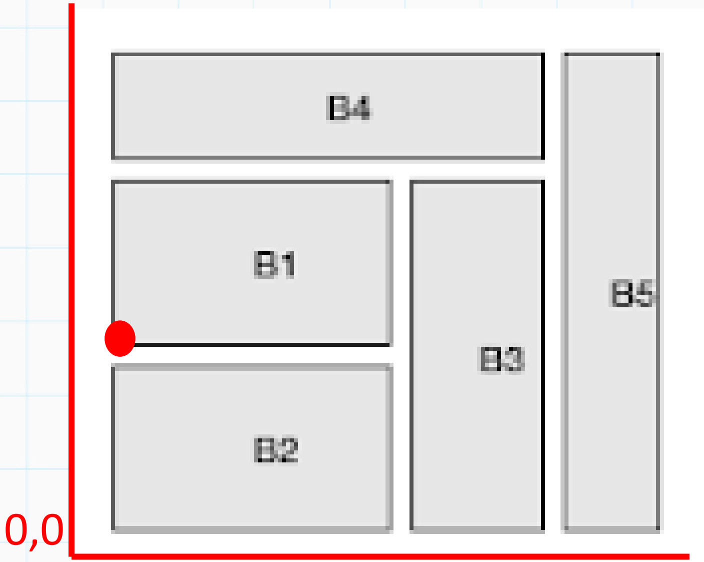
Modelando com variáveis Inteiras



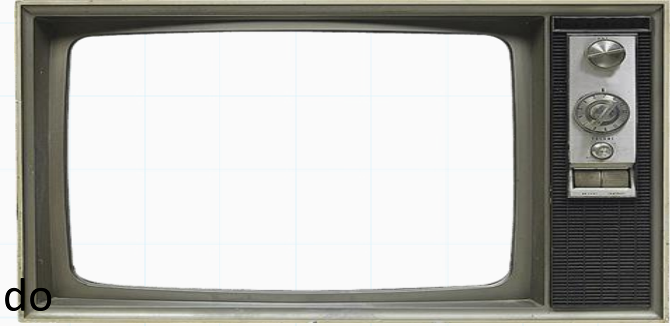
<u>Ex: Problema de colocação de cerâmica:</u> Em um plano retângulo de largura W e altura H, n cerâmicas retangulares devem ser colocadas, cada uma com largura w_i e altura h_i. Esta versão do problema não tem objetivo, buscamos apenas uma solução viável.

As cerâmicas não giram

<u>Variáveis</u>: x_i e y_i >= 0 indicam a posição origem da cerâmica i (canto inferior esquerdo)



Modelando com variáveis Inteiras



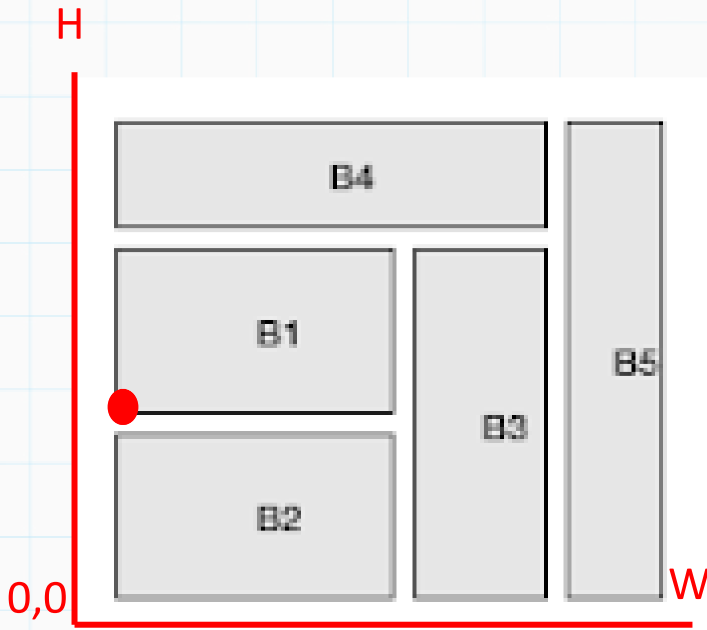
Ex: Problema de colocação de cerâmica: Em um plano retângulo de largura W e altura H, n cerâmicas retangulares devem ser colocadas, cada uma com largura w_i e altura h_i. Esta versão do problema não tem objetivo, buscamos apenas uma solução viável.

As cerâmicas não giram

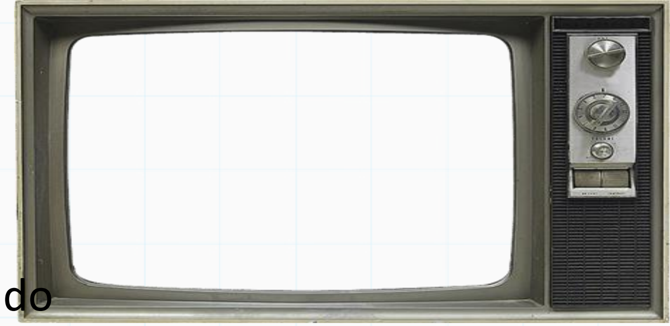
Variáveis: x_i e $y_i >= 0$ indicam a posição origem da cerâmica i (canto inferior esquerdo)

Restrições:

limites do plano



Modelando com variáveis Inteiras



Ex: Problema de colocação de cerâmica: Em um plano retângulo de largura W e altura H, n cerâmicas retangulares devem ser colocadas, cada uma com largura w_i e altura h_i. Esta versão do problema não tem objetivo, buscamos apenas uma solução viável.

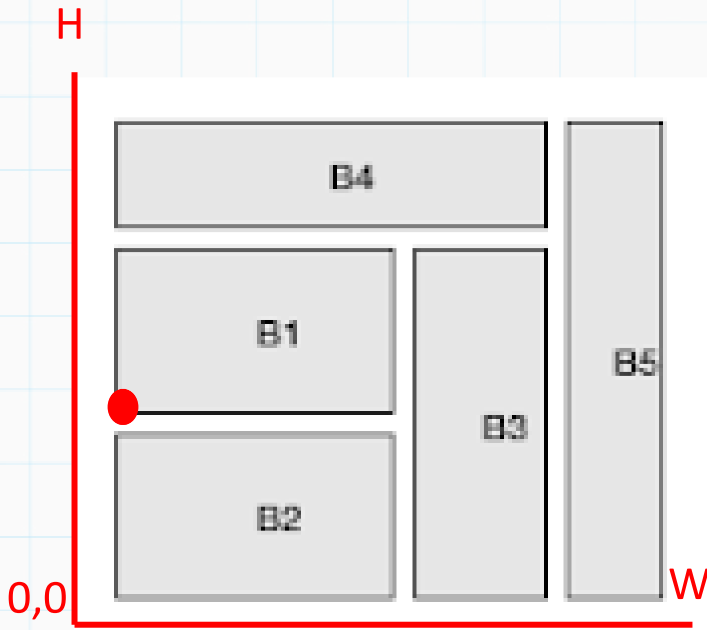
As cerâmicas não giram

Variáveis: x_i e $y_i >= 0$ indicam a posição origem da cerâmica i (canto inferior esquerdo)

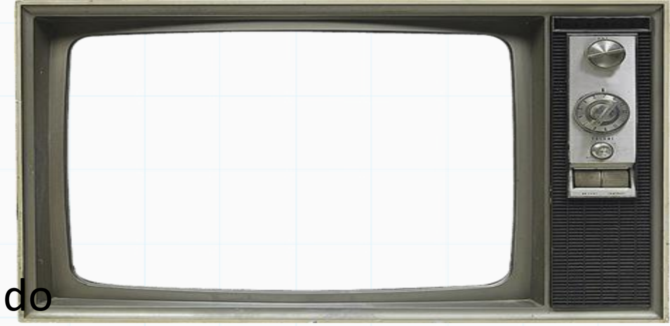
Restrições:

$$0 \leq x_i \leq W - w_i, \;\; 0 \leq y_i \leq H - h_i, \quad i = 1, \ldots, n.$$

limites do plano



Modelando com variáveis Inteiras



Ex: Problema de colocação de cerâmica: Em um plano retângulo de largura W e altura H, n cerâmicas retangulares devem ser colocadas, cada uma com largura w_i e altura h_i. Esta versão do problema não tem objetivo, buscamos apenas uma solução viável.

As cerâmicas não giram

Variáveis: x_i e y_i >= 0 indicam a posição origem da cerâmica i (canto inferior esquerdo)

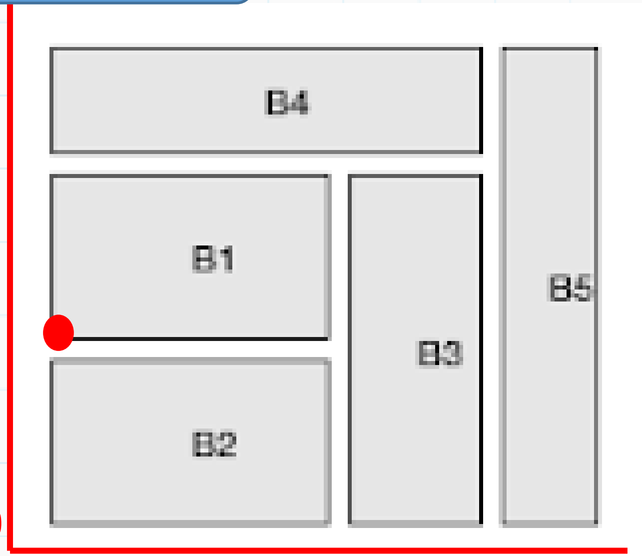
Restrições:

$$0 \le x_i \le W - w_i,\ \ 0 \le y_i \le H - h_i, \quad i = 1, \ldots, n.$$

limites do plano

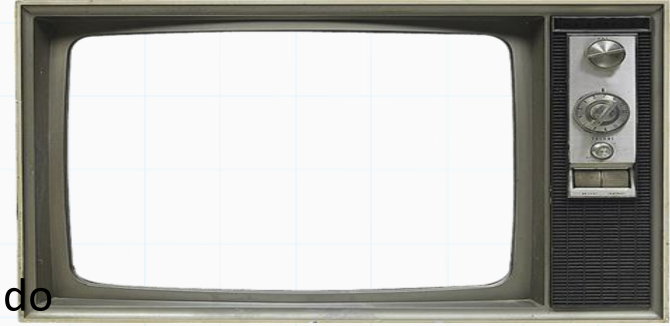
Se soubermos que i está a esquerda de j

Evitar interseção de cerâmicas i e j



0,0

Modelando com variáveis Inteiras



Ex: Problema de colocação de cerâmica: Em um plano retângulo de largura W e altura H, n cerâmicas retangulares devem ser colocadas, cada uma com largura w_i e altura h_i. Esta versão do problema não tem objetivo, buscamos apenas uma solução viável.

As cerâmicas não giram

Variáveis: x_i e y_i >= 0 indicam a posição origem da cerâmica i (canto inferior esquerdo)

Restrições:

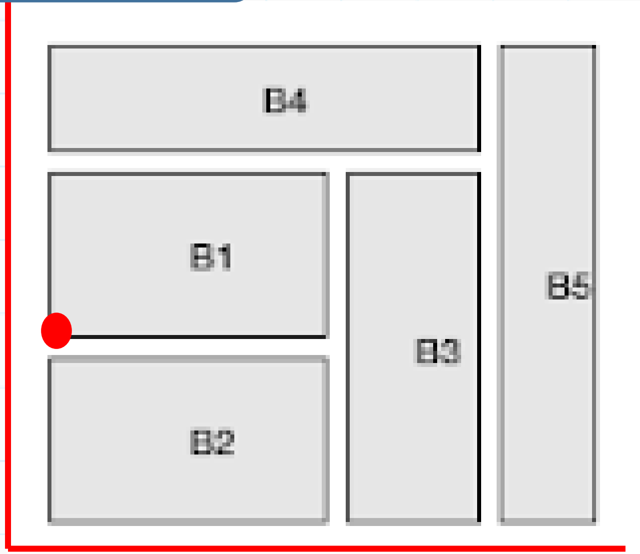
$$0 \le x_i \le W - w_i,\ \ 0 \le y_i \le H - h_i, \quad i = 1, \ldots, n.$$

limites do plano

$$x_i + w_i \le x_j$$

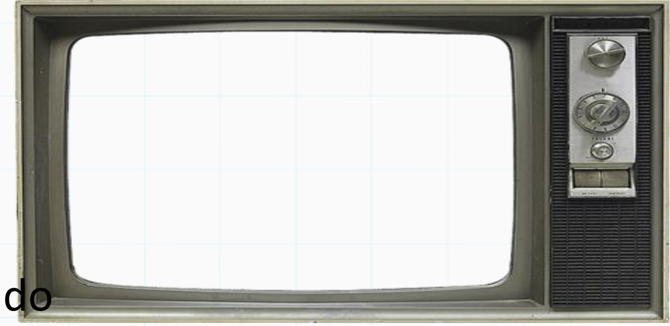
Se soubermos que i está a esquerda de j

Evitar interseção de cerâmicas i e j



0,0

Modelando com variáveis Inteiras



<u>Ex: Problema de colocação de cerâmica:</u> Em um plano retângulo de largura W e altura H, n cerâmicas retangulares devem ser colocadas, cada uma com largura w_i e altura h_i. Esta versão do problema não tem objetivo, buscamos apenas uma solução viável.

As cerâmicas não giram

<u>Variáveis</u>: x_i e y_i >= 0 indicam a posição origem da cerâmica i (canto inferior esquerdo)

<u>Restrições</u>:

$$0 \leq x_i \leq W - w_i, \;\; 0 \leq y_i \leq H - h_i, \quad i = 1, \ldots, n.$$

limites do plano

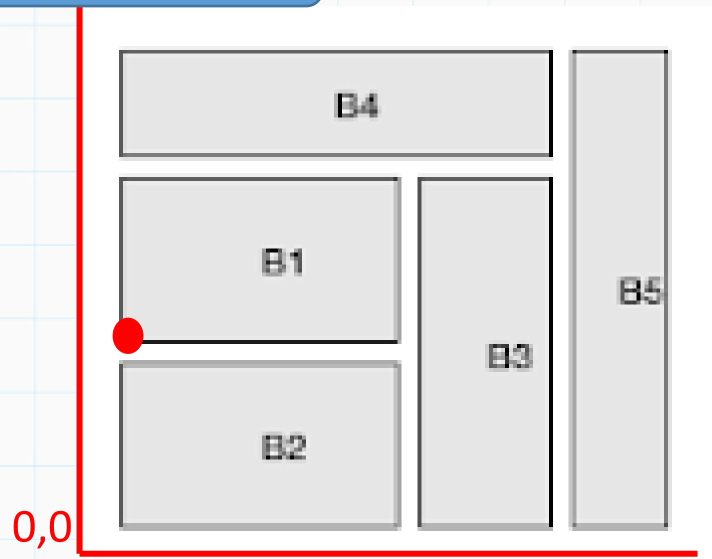
$x_i + w_i \leq x_j$ — i pode estar a esquerda de j

$x_j + w_j \leq x_i$ — i pode estar a direita de j

$y_i + h_i \leq y_j$ — i pode estar abaixo de j

$y_j + h_j \leq y_i$ — i pode estar acima de j

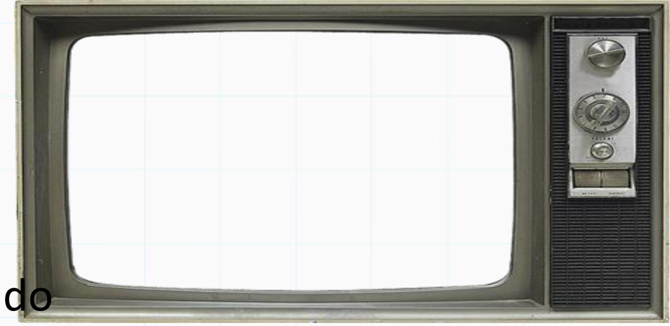
Evitar interseção de cerâmicas i e j



<u>Temos que garantir que pelo menos uma destas restrições seja obedecida</u>

Como ?

Modelando com variáveis Inteiras



<u>Ex: Problema de colocação de cerâmica:</u> Em um plano retângulo de largura W e altura H, n cerâmicas retangulares devem ser colocadas, cada uma com largura w_i e altura h_i. Esta versão do problema não tem objetivo, buscamos apenas uma solução viável.

As cerâmicas não giram

<u>Variáveis</u>: x_i e y_i >= 0 indicam a posição origem da cerâmica i (canto inferior esquerdo)

<u>Restrições</u>:

$$0 \leq x_i \leq W - w_i, \ \ 0 \leq y_i \leq H - h_i, \quad i = 1, \ldots, n.$$

limites do plano

$x_i + w_i \leq x_j$ i pode estar a esquerda de j

$x_j + w_j \leq x_i$ i pode estar a direita de j

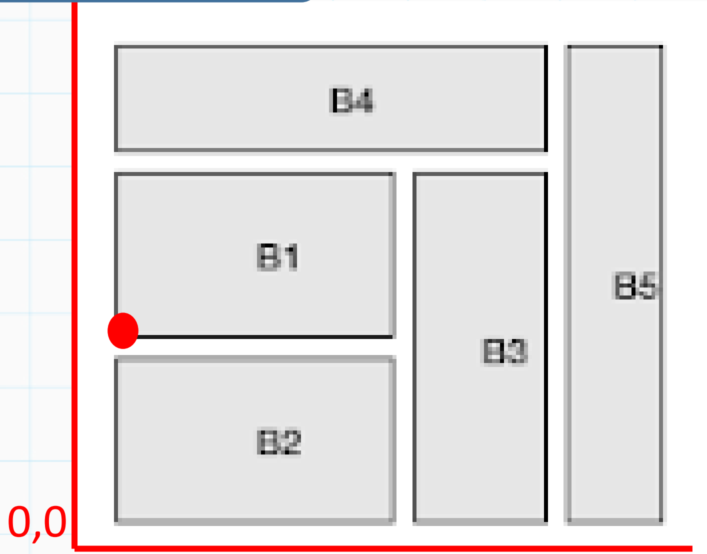
$y_i + h_i \leq y_j$ i pode estar abaixo de j

$y_j + h_j \leq y_i$ i pode estar acima de j

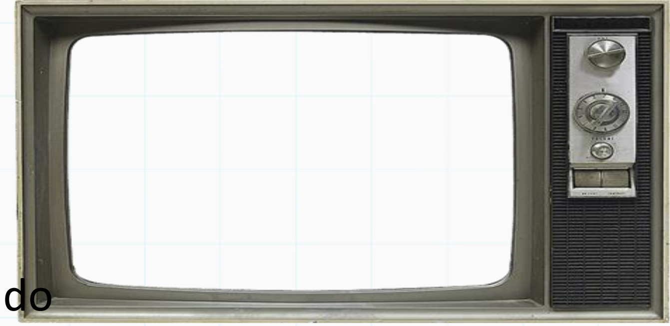
Evitar interseção de cerâmicas i e j

E agora ?

Criamos as variáveis z^l_{ij}, z^r_{ij}, z^b_{ij}, e z^a_{ij}, <u>binárias</u> que vão representar que pelo menos uma das restrições será obedecida.



Modelando com variáveis Inteiras



Ex: Problema de colocação de cerâmica: Em um plano retângulo de largura W e altura H, n cerâmicas retangulares devem ser colocadas, cada uma com largura w_i e altura h_i. Esta versão do problema não tem objetivo, buscamos apenas uma solução viável.

As cerâmicas não giram

Variáveis: x_i e $y_i >= 0$ indicam a posição origem da cerâmica i (canto inferior esquerdo)

Restrições:

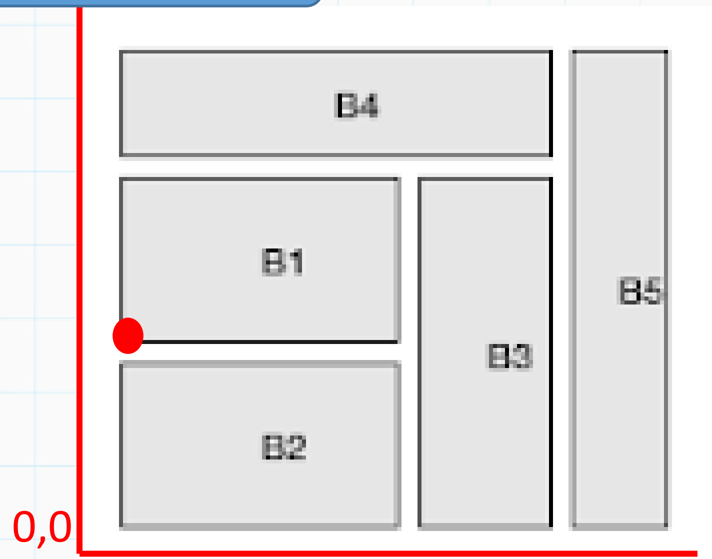
$$0 \le x_i \le W - w_i, \; 0 \le y_i \le H - h_i, \quad i = 1, \ldots, n.$$

limites do plano

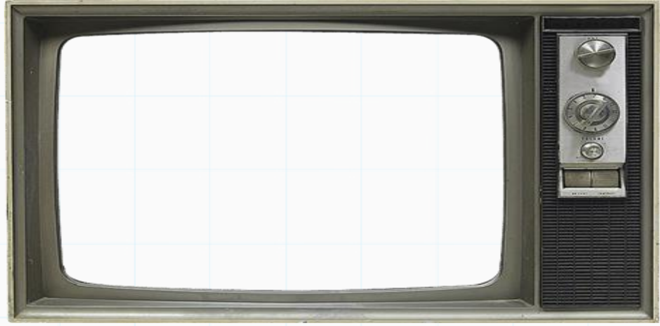
$$\begin{aligned} x_i + w_i &\le x_j \\ x_j + w_j &\le x_i \\ y_i + h_i &\le y_j \\ y_j + h_j &\le y_i \end{aligned}$$

$$\boxed{\begin{aligned} x_i + w_i &\le x_j + W(1 - z^l_{ij}), \\ x_j + w_j &\le x_i + W(1 - z^r_{ij}), \\ y_i + h_i &\le y_j + H(1 - z^b_{ij}), \\ y_j + h_j &\le y_i + H(1 - z^a_{ij}), \\ z^l_{ij} + z^r_{ij} + z^b_{ij} + z^a_{ij} &\ge 1. \end{aligned}}$$

Evitar interseção de cerâmicas i e j



Modelando com variáveis Inteiras

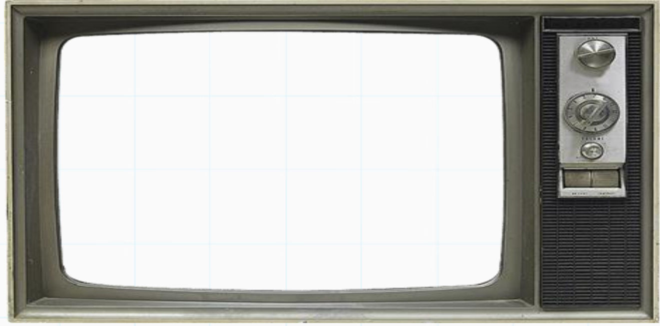


<u>Soft Constraints:</u> Suponha que desejamos indicar com uma variável binária y se a restrição:

$$a^T x <= b$$

está violada (y=1) ou não está violada (y=0). Podemos até permitir a violação, mas ela será penalizada.

Modelando com variáveis Inteiras

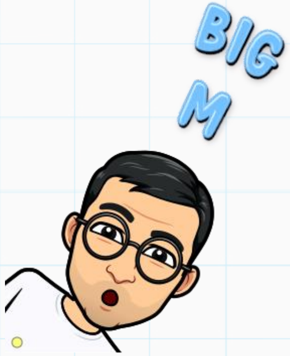


Soft Constraints: Suponha que desejamos indicar com uma variável binária y se a restrição:

$$a^T x <= b$$

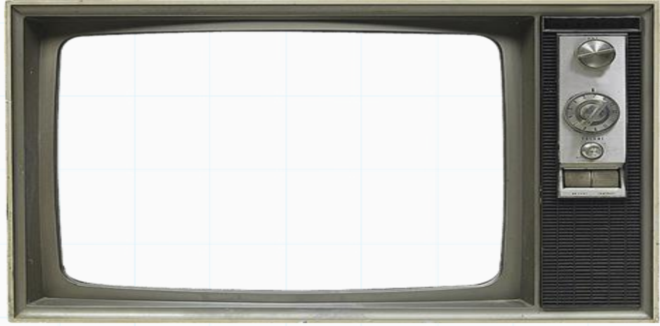
está violada (y=1) ou não está violada (y=0). Podemos até permitir a violação, mas ela será penalizada.

$$a^T x <= b + M.y$$



onde queremos Min y

Modelando com variáveis Inteiras

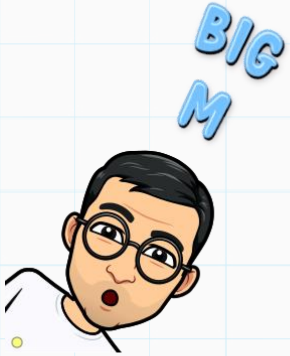


Soft Constraints: Suponha que desejamos indicar com uma variável binária y se a restrição:

$$a^T x <= b$$

está violada (y=1) ou não está violada (y=0). Podemos até permitir a violação, mas ela será penalizada.

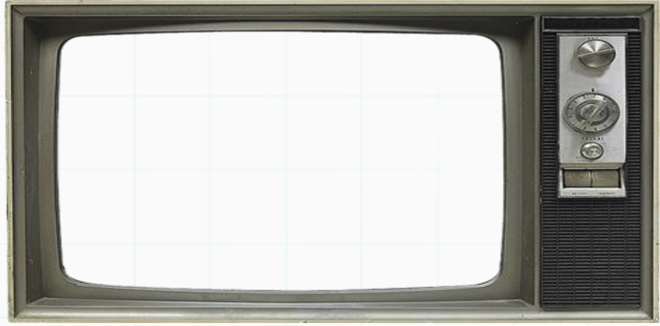
$$a^T x <= b + M.y$$



onde queremos Min y

E se fosse a interpretação de violada (y=0) ou não está violada (y=1)

Modelando com variáveis Inteiras

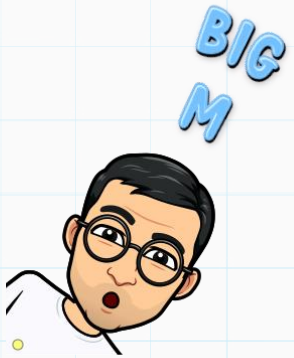


Soft Constraints: Suponha que desejamos indicar com uma variável binária y se a restrição:

$$a^T x <= b$$

está violada (y=1) ou não está violada (y=0). Podemos até permitir a violação, mas ela será penalizada.

$$a^T x <= b + M.y$$



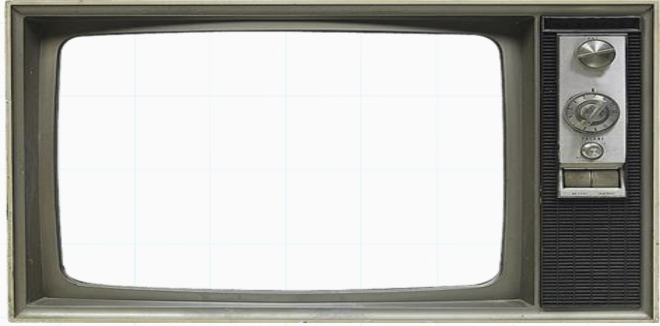
onde queremos Min y

E se fosse a interpretação de violada (y=0) ou não está violada (y=1)

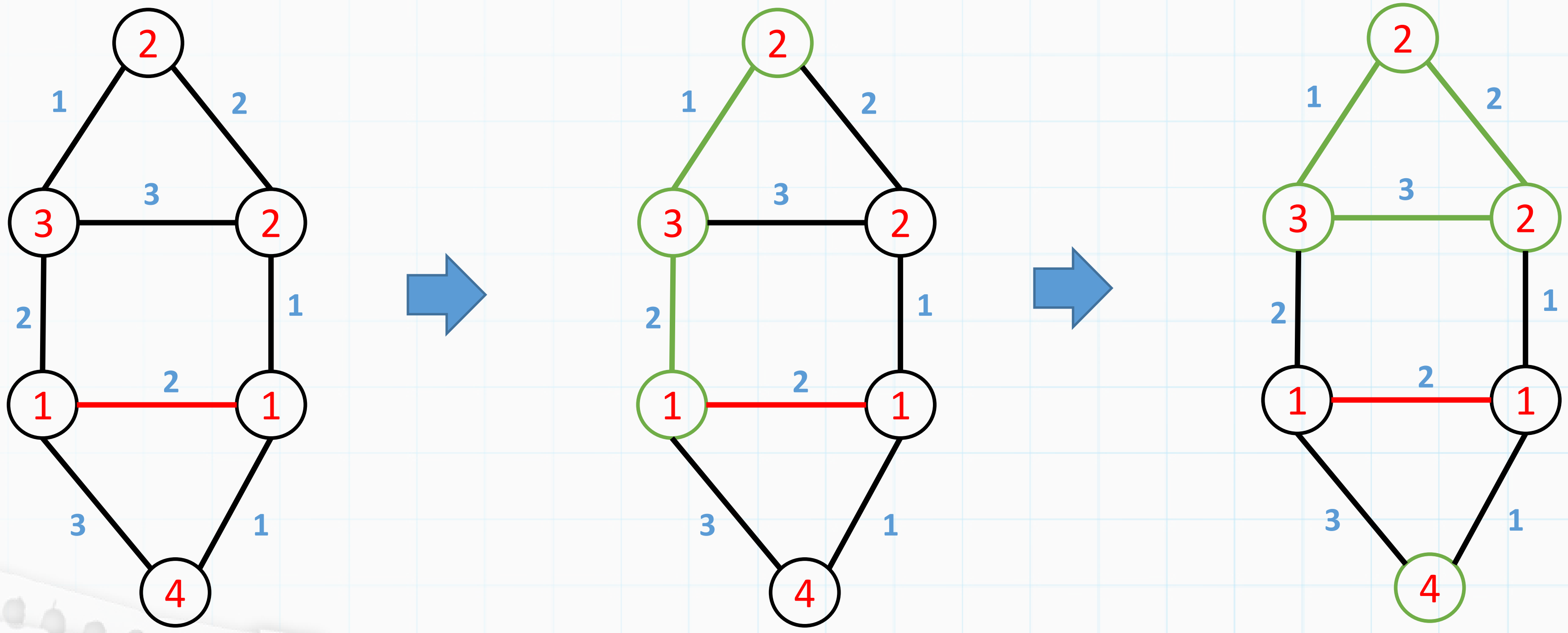
$$a^T x <= b + M(1-y)$$

onde queremos Max y

Modelando com variáveis Inteiras



Ex: Problema do subconjunto de vértices de peso máximo restrito: Seja G=(V, E) um grafo não direcionado, onde cada vértice i possui um prêmio p_i mas toda aresta ij possui um custo c_{ij}.
Porém existe um subconjunto de arestas E' ⊆ E que não podem entrar na solução (uma aresta está na solução se seus vértices extremidades estão).
Queremos encontrar o subconjunto de vértices de peso máximo, proibindo arestas de E' e penalizando arestas de E/E' que entrarem na solução.

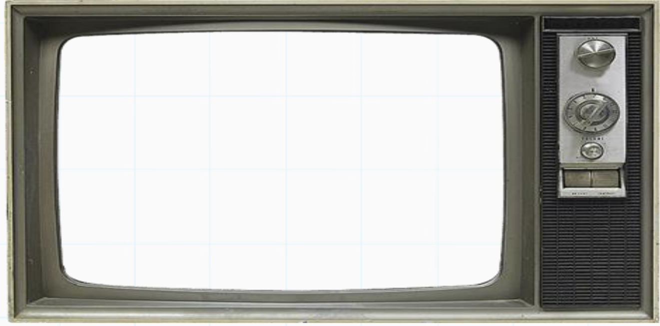


E'

sol = 6-3=3

sol = 11-6=5

Modelando com variáveis Inteiras

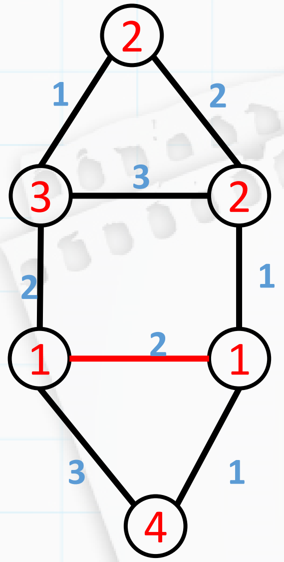


<u>Ex: Problema do subconjunto de vértices de peso máximo restrito:</u> Seja G=(V, E) um grafo não direcionado, onde cada vértice i possui um prêmio p_i mas toda aresta ij possui um custo c_{ij}.
Porém existe um subconjunto de arestas E' ⊆ E que não podem entrar na solução (<u>uma aresta está na solução se seus vértices extremidades estão</u>).
Queremos encontrar o subconjunto de vértices de peso máximo, proibindo arestas de E' e penalizando arestas de E/E' que entrarem na solução.

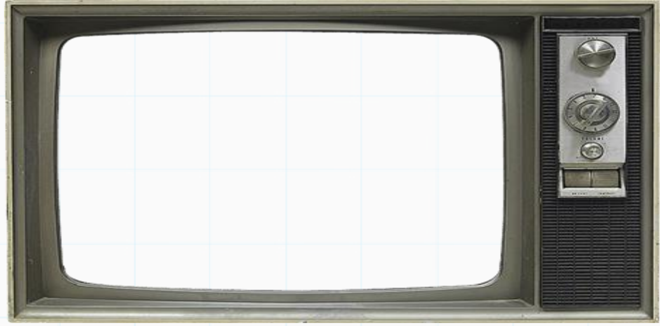
<u>Variáveis</u>:

$$x_i = \begin{cases} 1, & \text{se o vértice i pertence a solução} \\ 0, & \text{caso contrário} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{se a aresta ij será descontada na solução} \\ 0, & \text{caso contrário} \end{cases}$$



Modelando com variáveis Inteiras



Ex: Problema do subconjunto de vértices de peso máximo restrito: Seja G=(V, E) um grafo não direcionado, onde cada vértice i possui um prêmio p_i mas toda aresta ij possui um custo c_{ij}.
Porém existe um subconjunto de arestas $E' \subseteq E$ que não podem entrar na solução (uma aresta está na solução se seus vértices extremidades estão).
Queremos encontrar o subconjunto de vértices de peso máximo, proibindo arestas de E' e penalizando arestas de E/E' que entrarem na solução.

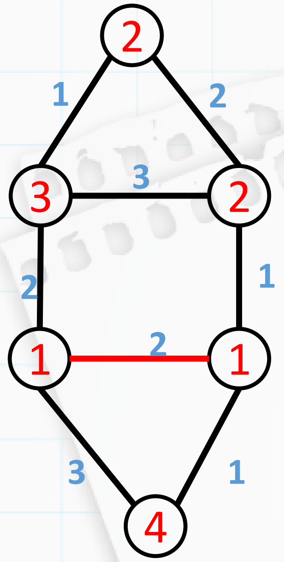
Variáveis:

$$x_i = \begin{cases} 1, & \text{se o vértice i pertence a solução} \\ 0, & \text{caso contrário} \end{cases}$$

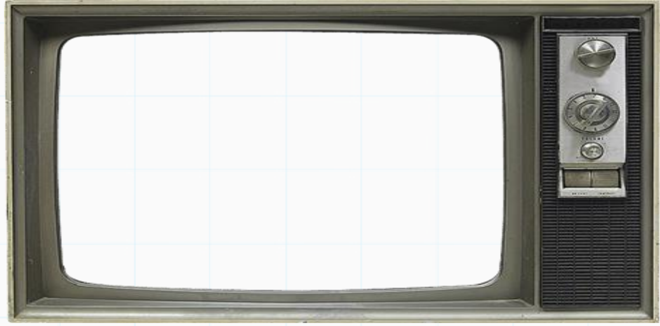
$$y_{ij} = \begin{cases} 1, & \text{se a aresta ij será descontada na solução} \\ 0, & \text{caso contrário} \end{cases}$$

Restrições:

arestas E'



Modelando com variáveis Inteiras



<u>Ex: Problema do subconjunto de vértices de peso máximo restrito:</u> Seja G=(V, E) um grafo não direcionado, onde cada vértice i possui um prêmio p_i mas toda aresta ij possui um custo c_{ij}.
Porém existe um subconjunto de arestas E' ⊆ E que não podem entrar na solução (<u>uma aresta está na solução se seus vértices extremidades estão</u>).
Queremos encontrar o subconjunto de vértices de peso máximo, proibindo arestas de E' e penalizando arestas de E/E' que entrarem na solução.

<u>Variáveis</u>:

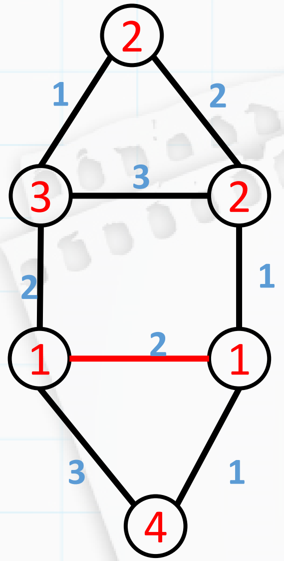
$$x_i = \begin{cases} 1, & \text{se o vértice i pertence a solução} \\ 0, & \text{caso contrário} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{se a aresta ij será descontada na solução} \\ 0, & \text{caso contrário} \end{cases}$$

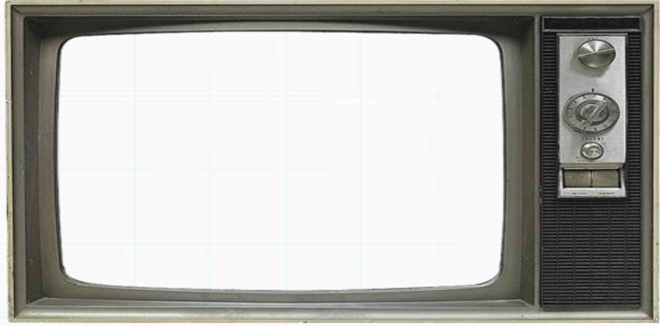
<u>Restrições</u>:

$$x_i + x_j \leq 1, \quad \forall ij \in E'$$

arestas E'



Modelando com variáveis Inteiras



<u>Ex: Problema do subconjunto de vértices de peso máximo restrito:</u> Seja G=(V, E) um grafo não direcionado, onde cada vértice i possui um prêmio p_i mas toda aresta ij possui um custo c_{ij}.
Porém existe um subconjunto de arestas E' ⊆ E que não podem entrar na solução (<u>uma aresta está na solução se seus vértices extremidades estão</u>).
Queremos encontrar o subconjunto de vértices de peso máximo, proibindo arestas de E' e penalizando arestas de E/E' que entrarem na solução.

<u>Variáveis</u>:

$$x_i = \begin{cases} 1, & \text{se o vértice i pertence a solução} \\ 0, & \text{caso contrário} \end{cases}$$

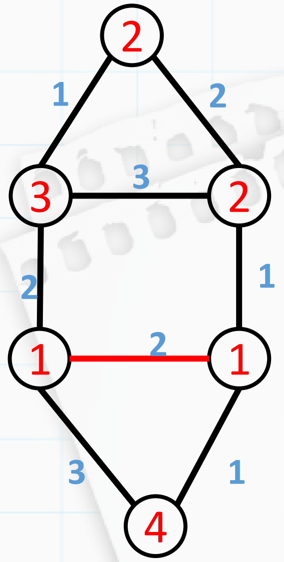
$$y_{ij} = \begin{cases} 1, & \text{se a aresta ij será descontada na solução} \\ 0, & \text{caso contrário} \end{cases}$$

<u>Restrições</u>:

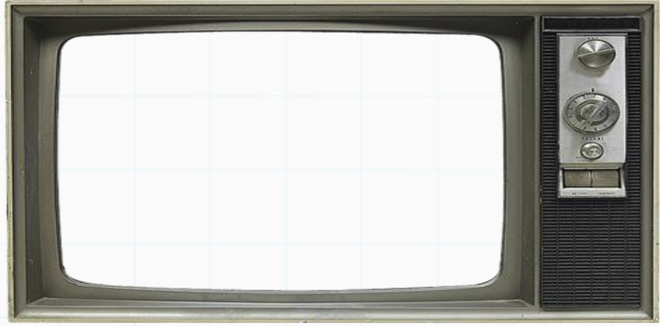
$$x_i + x_j \leq 1, \quad \forall ij \in E'$$

arestas E'

arestas penalizadas E\E'



Modelando com variáveis Inteiras



Ex: Problema do subconjunto de vértices de peso máximo restrito: Seja G=(V, E) um grafo não direcionado, onde cada vértice i possui um prêmio p_i mas toda aresta ij possui um custo c_{ij}.
Porém existe um subconjunto de arestas E' ⊆ E que não podem entrar na solução (uma aresta está na solução se seus vértices extremidades estão).
Queremos encontrar o subconjunto de vértices de peso máximo, proibindo arestas de E' e penalizando arestas de E/E' que entrarem na solução.

Variáveis:

$$x_i = \begin{cases} 1, & \text{se o vértice i pertence a solução} \\ 0, & \text{caso contrário} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{se a aresta ij será descontada na solução} \\ 0, & \text{caso contrário} \end{cases}$$

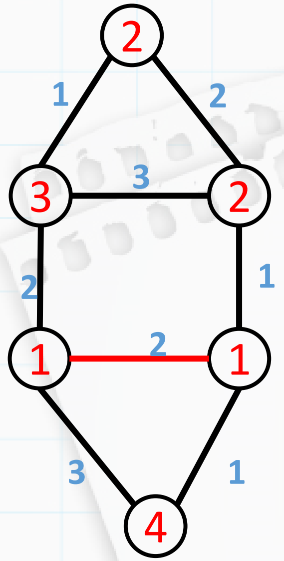
Restrições:

$$x_i + x_j \leq 1, \quad \forall ij \in E'$$

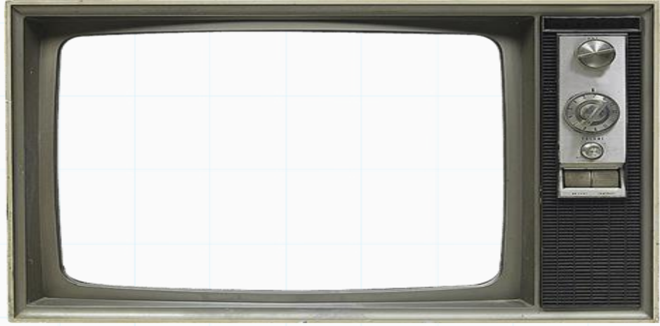
arestas E'

$$x_i + x_j \leq 1 + y_{ij}, \quad \forall ij \in E \backslash E'$$

arestas penalizadas E\E'



Modelando com variáveis Inteiras



<u>Ex: Problema do subconjunto de vértices de peso máximo restrito:</u> Seja G=(V, E) um grafo não direcionado, onde cada vértice i possui um prêmio p_i mas toda aresta ij possui um custo c_{ij}.
Porém existe um subconjunto de arestas E’ ⊆ E que não podem entrar na solução (<u>uma aresta está na solução se seus vértices extremidades estão</u>).
Queremos encontrar o subconjunto de vértices de peso máximo, proibindo arestas de E’ e penalizando arestas de E/E’ que entrarem na solução.

<u>Variáveis</u>:

$$x_i = \begin{cases} 1, & \text{se o vértice i pertence a solução} \\ 0, & \text{caso contrário} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{se a aresta ij será descontada na solução} \\ 0, & \text{caso contrário} \end{cases}$$

<u>Restrições</u>:

$$x_i + x_j \leq 1, \quad \forall ij \in E'$$

arestas E’

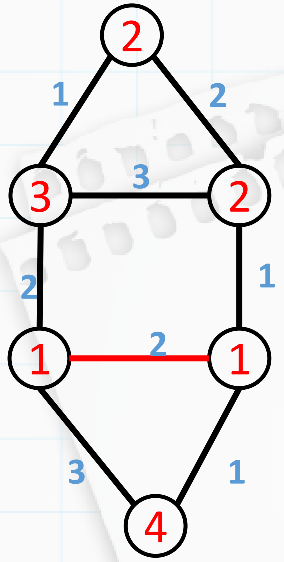
$$x_i + x_j \leq 1 + y_{ij}, \quad \forall ij \in E \backslash E'$$

arestas penalizadas E\E’

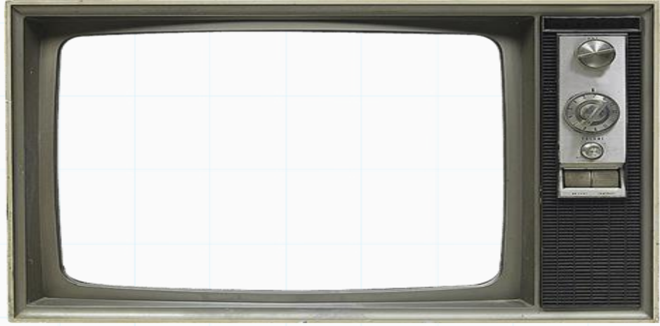
$$x_i \in \{0,1\}, \quad \forall i \in V$$

$$y_{ij} \in \{0,1\}, \quad \forall ij \in E \backslash E'$$

integralidade



Modelando com variáveis Inteiras



Ex: Problema do subconjunto de vértices de peso máximo restrito: Seja G=(V, E) um grafo não direcionado, onde cada vértice i possui um prêmio p_i mas toda aresta ij possui um custo c_{ij}.
Porém existe um subconjunto de arestas E’ ⊆ E que não podem entrar na solução (uma aresta está na solução se seus vértices extremidades estão).
Queremos encontrar o subconjunto de vértices de peso máximo, proibindo arestas de E’ e penalizando arestas de E/E’ que entrarem na solução.

Função objetivo:

Variáveis:

$$x_i = \begin{cases} 1, & \text{se o vértice i pertence a solução} \\ 0, & \text{caso contrário} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{se a aresta ij será descontada na solução} \\ 0, & \text{caso contrário} \end{cases}$$

Restrições:

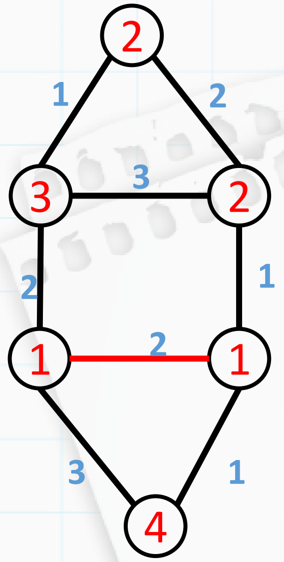
$$x_i + x_j \leq 1, \quad \forall ij \in E'$$ arestas E’

$$x_i + x_j \leq 1 + y_{ij}, \quad \forall ij \in E \backslash E'$$ arestas penalizadas E\E’

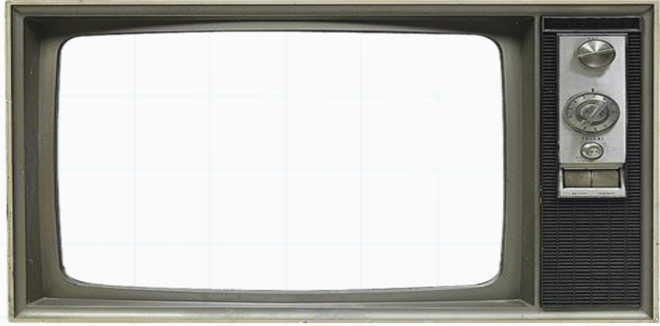
$$x_i \in \{0,1\}, \quad \forall i \in V$$

$$y_{ij} \in \{0,1\}, \quad \forall ij \in E \backslash E'$$

integralidade



Modelando com variáveis Inteiras



Ex: Problema do subconjunto de vértices de peso máximo restrito: Seja G=(V, E) um grafo não direcionado, onde cada vértice i possui um prêmio p_i mas toda aresta ij possui um custo c_{ij}.
Porém existe um subconjunto de arestas E' ⊆ E que não podem entrar na solução (uma aresta está na solução se seus vértices extremidades estão).
Queremos encontrar o subconjunto de vértices de peso máximo, proibindo arestas de E' e penalizando arestas de E/E' que entrarem na solução.

Variáveis:

$$x_i = \begin{cases} 1, & \text{se o vértice i pertence a solução} \\ 0, & \text{caso contrário} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{se a aresta ij será descontada na solução} \\ 0, & \text{caso contrário} \end{cases}$$

Função objetivo:

$$\max \sum_{i \in V} p_i x_i - \sum_{ij \in (E \setminus E')} c_{ij} y_{ij}$$

Restrições:

$$x_i + x_j \leq 1, \quad \forall ij \in E'$$

arestas E'

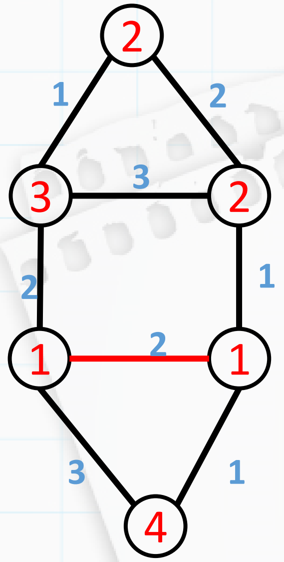
$$x_i + x_j \leq 1 + y_{ij}, \quad \forall ij \in E \setminus E'$$

arestas penalizadas E\E'

$$x_i \in \{0, 1\}, \quad \forall i \in V$$

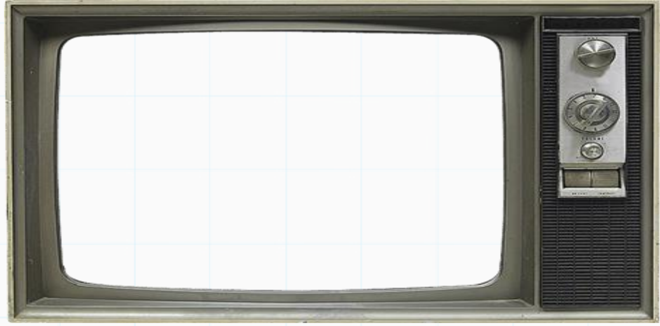
$$y_{ij} \in \{0, 1\}, \quad \forall ij \in E \setminus E'$$

integralidade



Vamos Modelar, Inteiro ?

Vamos ver outro exemplo agora:



- <u>Problema dos padrões das latinhas</u>:
  - Para se fazer uma latinha, precisamos de um corpo e duas tampas

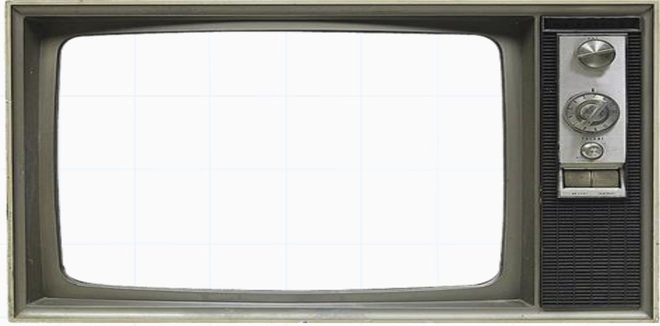
tampa

corpo

tampa

Vamos Modelar, Inteiro ?

Vamos ver outro exemplo agora:



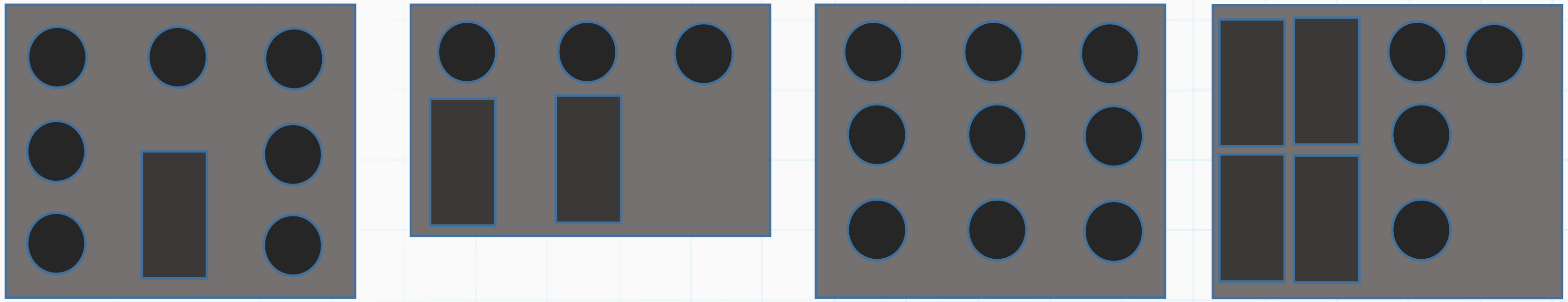
- Problema dos padrões das latinhas:
  - Para se fazer uma latinha, precisamos de um corpo e duas tampas
  - Uma fábrica de latinhas possui 4 padrões de impressão em folhas de metal (2 tamanhos) e um tempo de impressão para cortar a folha

tampa

corpo

tampa

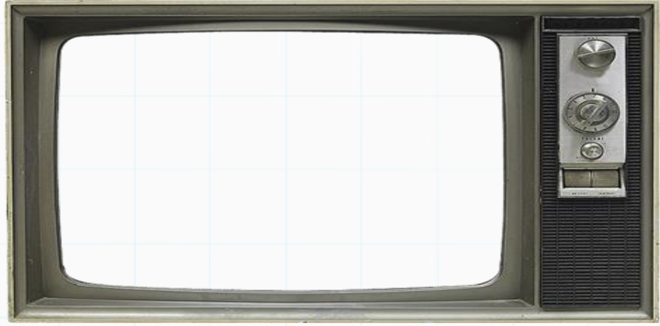
tam1 tam2 tam1 tam1



Padrões	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

Vamos ver outro exemplo agora:



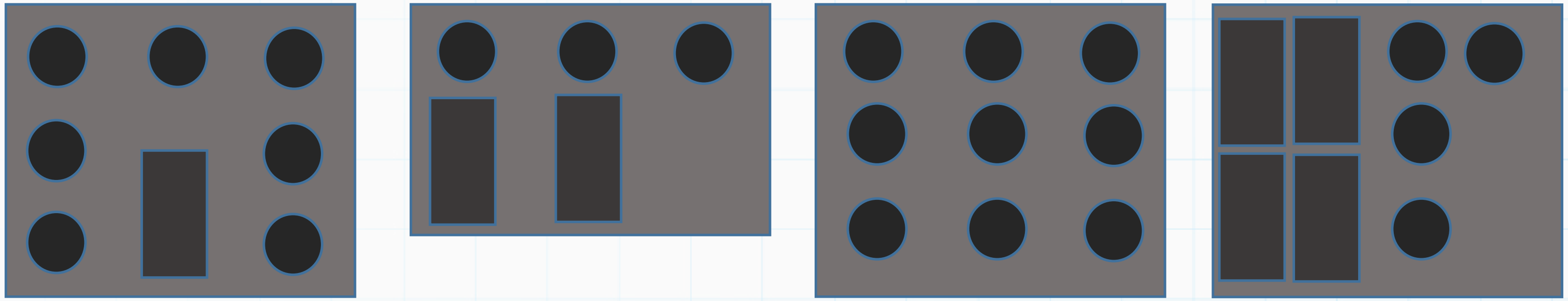
tampa

corpo

tampa

- <u>Problema dos padrões das latinhas</u>:
  - Para se fazer uma latinha, precisamos de um corpo e duas tampas
  - Uma fábrica de latinhas possui 4 padrões de impressão em folhas de metal (2 tamanhos) e um tempo de impressão para cortar a folha

tam1 tam2 tam1 tam1

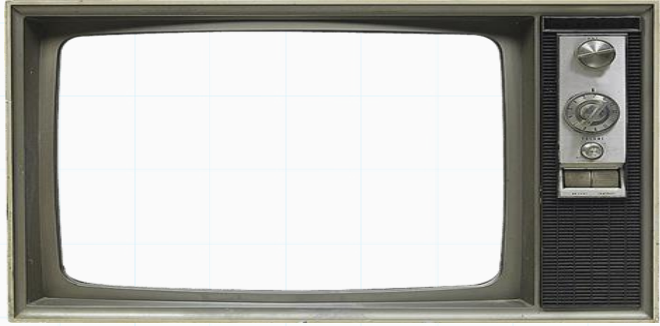


  - A fábrica possui 200 folhas de metal de tam1 e 90 folhas de tam2
  - A fábrica tem um tempo limite de 400 horas para produzir as latinhas

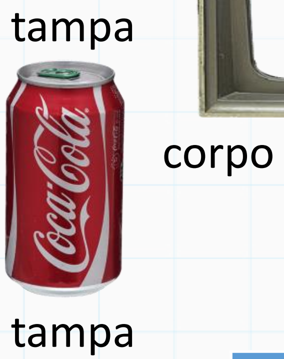
Padrões	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

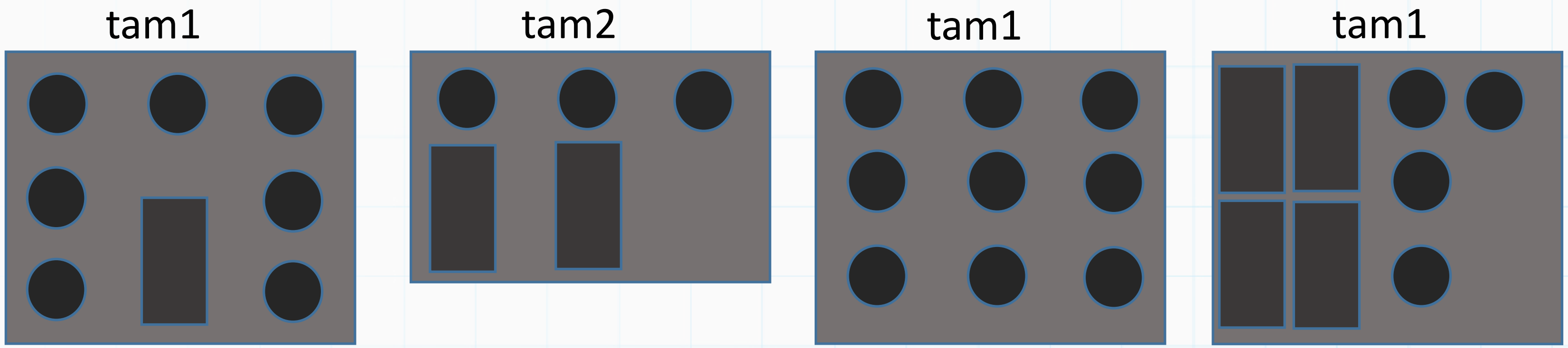
Vamos Modelar, Inteiro ?

Vamos ver outro exemplo agora:



- <u>Problema dos padrões das latinhas</u>:
  - Para se fazer uma latinha, precisamos de um corpo e duas tampas
  - Uma fábrica de latinhas possui 4 padrões de impressão em folhas de metal (2 tamanhos) e um tempo de impressão para cortar a folha



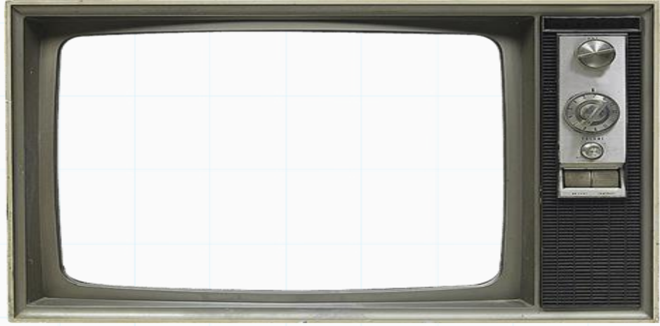


Padrões	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

- A fábrica possui 200 folhas de metal de tam1 e 90 folhas de tam2
- A fábrica tem um tempo limite de 400 horas para produzir as latinhas
- Cada latinha é vendida a 50c
- Cada corpo não utilizado possui custo de estocagem de 5c e cada tampa tem custo de estocagem de 3c

Vamos Modelar, Inteiro ?

Vamos ver outro exemplo agora:



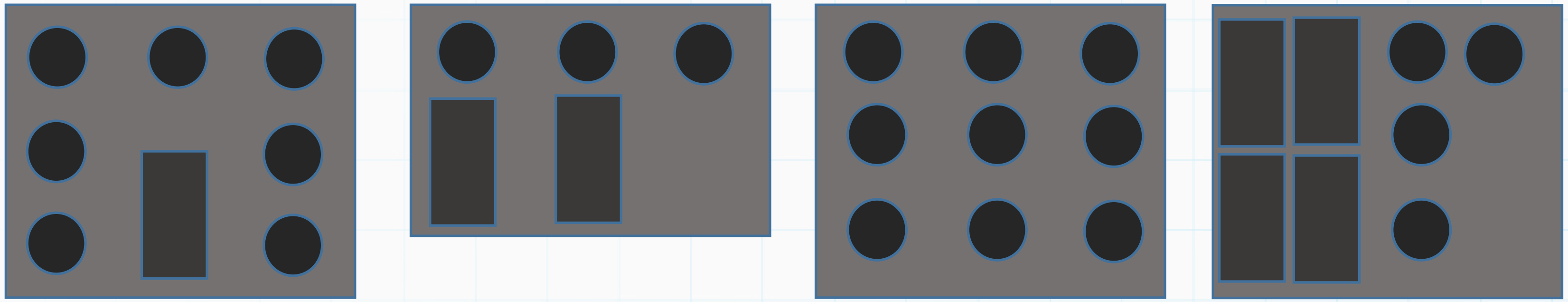
- <u>Problema dos padrões das latinhas</u>:
  - Para se fazer uma latinha, precisamos de um corpo e duas tampas
  - Uma fábrica de latinhas possui 4 padrões de impressão em folhas de metal (2 tamanhos) e um tempo de impressão para cortar a folha

tampa

corpo

tampa

tam1 tam2 tam1 tam1



Padrões	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

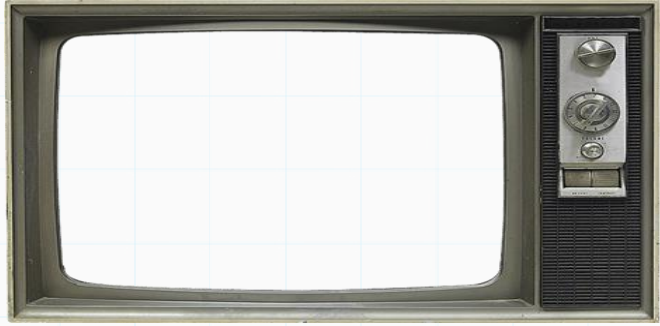
- A fábrica possui 200 folhas de metal de tam1 e 90 folhas de tam2
- A fábrica tem um tempo limite de 400 horas para produzir as latinhas
- Cada latinha é vendida a 50c
- Cada corpo não utilizado possui custo de estocagem de 5c e cada tampa tem custo de estocagem de 3c
- Quantas impressões de cada padrão devem ser feitas para maximizar o lucro da empresa ?

Vamos Modelar, Inteiro ?

tampa

corpo

tampa



- Variáveis de Decisão:

Ao escolher suas variáveis, leve em consideração como a solução deve ser representada !

Vamos Modelar, Inteiro ?

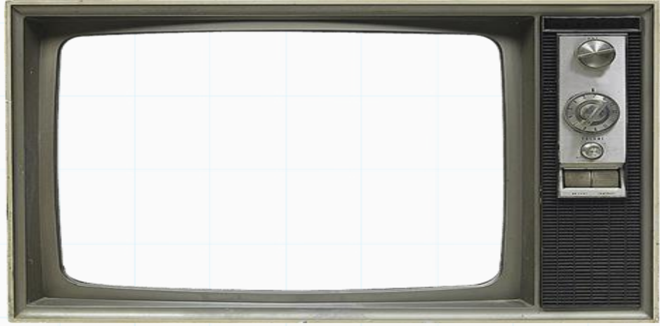
- Variáveis de Decisão:

  x_j: número de padrões impressos do tipo j=1,...,4

tampa

corpo

tampa

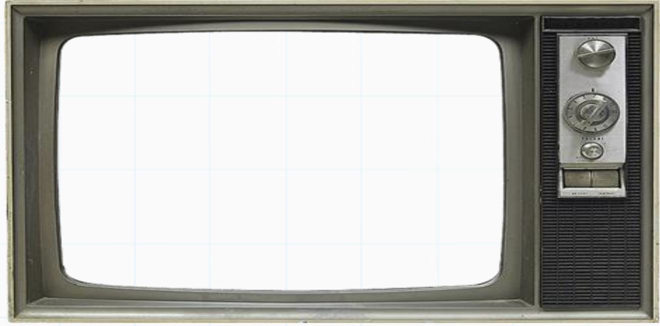


Ao escolher suas variáveis, leve em consideração como a solução deve ser representada !

Vamos Modelar, Inteiro ?

tampa

corpo



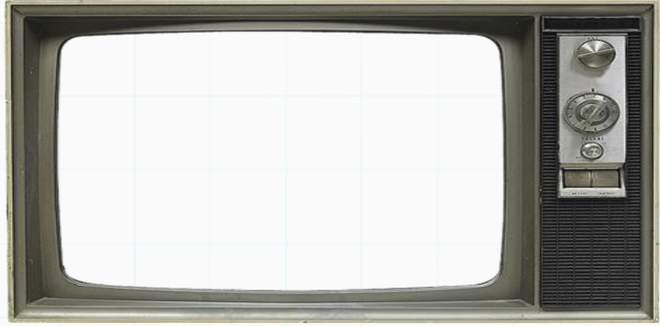
tampa

- Variáveis de Decisão:
  - x_j: número de padrões impressos do tipo i=1,...,4
- Restrições:
  - Tempo

A fábrica tem um tempo limite de 400 horas para produzir as latinhas

	1	**2**	**3**	**4**
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?



- Variáveis de Decisão:
  x_j: número de padrões impressos do tipo i=1,...,4

- Restrições:
  - Tempo

  $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$

A fábrica tem um tempo limite de 400 horas para produzir as latinhas

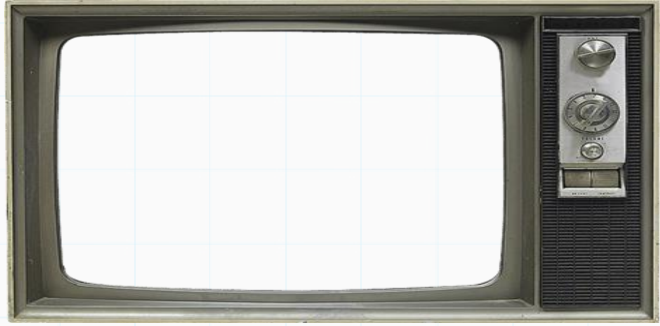
	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

tampa

corpo

tampa



- Variáveis de Decisão:
  x_j: número de padrões impressos do tipo i=1,...,4

- Restrições:
  - Tempo

  $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$
  - Materia Prima

    Folha 1

    Folha 2

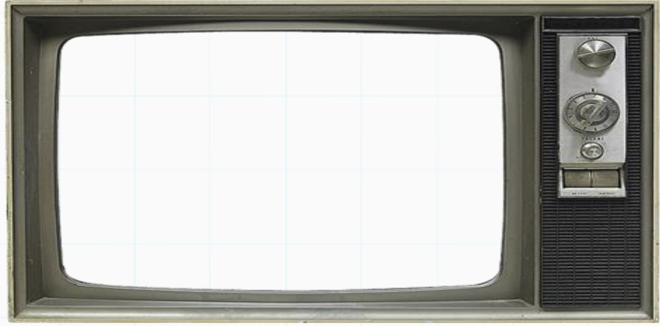
- A fábrica possui 200 folhas de metal de tam1 e 90 folhas de tam2

	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

tampa

corpo



tampa

- Variáveis de Decisão:

  x_j: número de padrões impressos do tipo i=1,...,4

- A fábrica possui 200 folhas de metal de tam1 e 90 folhas de tam2

- Restrições:
  - Tempo

    $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$
  - Materia Prima

    $x_1 + x_3 + x_4 <= 200$ Folha 1

    $x_2 <= 90$ Folha 2

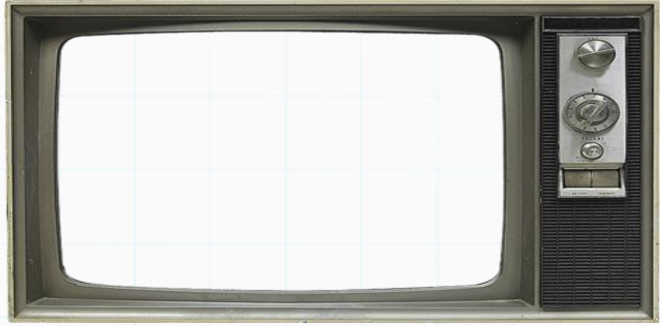
	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

tampa

corpo

tampa



- Cada latinha é vendida a 50c

- Variáveis de Decisão:
  x_j: número de padrões impressos do tipo i=1,...,4

- Restrições:
  - Tempo

    $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$
  - Materia Prima

    $x_1 + x_3 + x_4 <= 200$ Folha 1

    $x_2 <= 90$ Folha 2

- Função Objetivo: (como saber o #latinhas produzidas?)

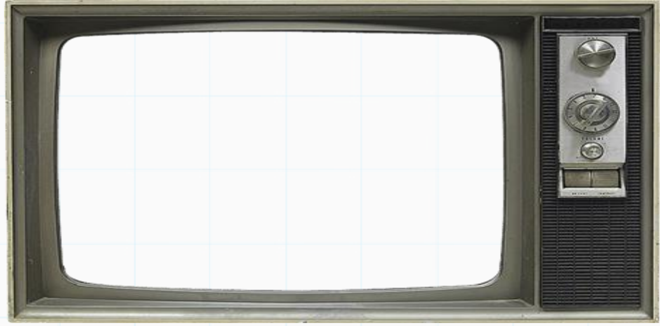
	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

tampa

corpo

tampa



- Variáveis de Decisão:
  - x_j: número de padrões impressos do tipo i=1,...,4
  - t: #tampas e c: #corpos
  - y: #latinhas produzidas

  (var. inteiras)
- Restrições:
  - Tempo

    $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$
  - Materia Prima

    $x_1 + x_3 + x_4 <= 200$ Folha 1

    $x_2 <= 90$ Folha 2

- Cada latinha é vendida a 50c

- Função Objetivo: (como saber o #latinhas produzidas?)

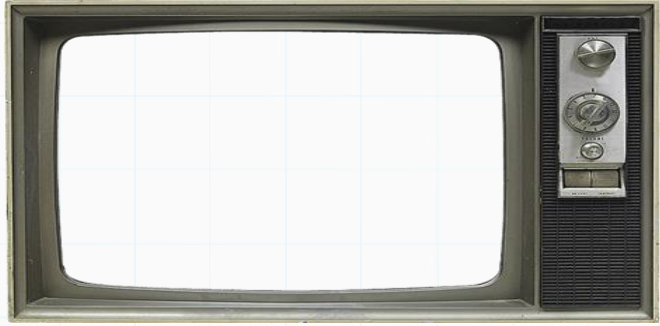
	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

tampa

corpo

tampa



- Variáveis de Decisão:
  - x_j: número de padrões impressos do tipo i=1,...,4
  - t: #tampas e c: #corpos
  - y: #latinhas produzidas

  (var. inteiras)
- Restrições:
  - Tempo

  $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$
  - Materia Prima

  $x_1 + x_3 + x_4 <= 200$ Folha 1

  $x_2 <= 90$ Folha 2

  $t = 7x_1 + 3x_2 + 9x_3 + 4x_4$ e $c = x_1 + 2x_2 + 4x_4$

  $y = ?$

- Função Objetivo: (como saber o #latinhas produzidas?)

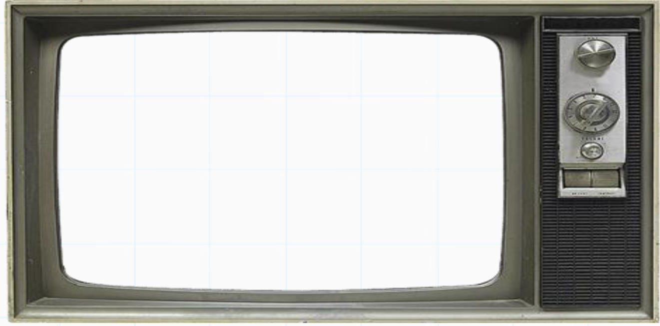
- Cada latinha é vendida a 50c

	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

tampa

corpo



tampa

- Variáveis de Decisão:
  - x_j: número de padrões impressos do tipo i=1,...,4
  - t: #tampas e c: #corpos
  - y: #latinhas produzidas

(var. inteiras)

- Cada latinha é vendida a 50c

- Restrições:
  - Tempo

    $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$
  - Materia Prima

    $x_1 + x_3 + x_4 <= 200$ Folha 1

    $x_2 <= 90$ Folha 2

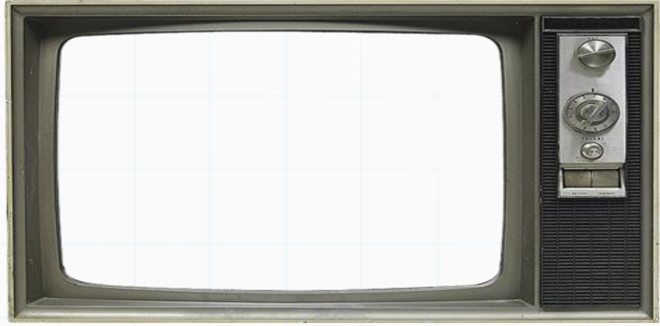
    $t = 7x_1 + 3x_2 + 9x_3 + 4x_4$ e $c = x_1 + 2x_2 + 4x_4$

    $y = \min(t/2 \ , \ c)$

- Função Objetivo: (como saber o #latinhas produzidas?)

	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?



- Variáveis de Decisão:
  - x_j: número de padrões impressos do tipo i=1,...,4
  - t: #tampas e c: #corpos
  - y: #latinhas produzidas

  (var. inteiras)
- Restrições:
  - Tempo

    $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$
  - Materia Prima

    $x_1 + x_3 + x_4 <= 200$ Folha 1

    $x_2 <= 90$ Folha 2

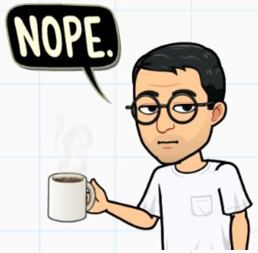
    $t = 7x_1 + 3x_2 + 9x_3 + 4x_4$ e $c = x_1 + 2x_2 + 4x_4$

    $y = \min(t/2 \ , \ c)$

NÃO LINEAR

- Função Objetivo: (como saber o #latinhas produzidas?)

- Cada latinha é vendida a 50c



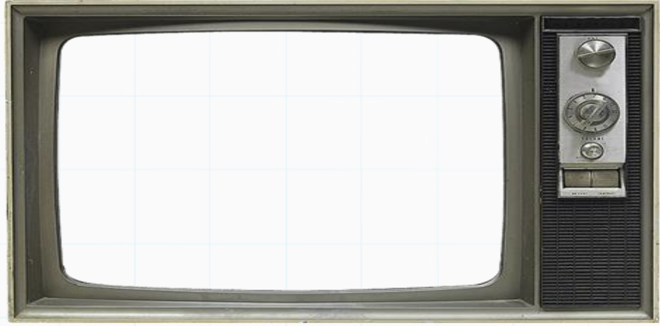
	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

tampa

corpo

tampa



- Variáveis de Decisão:
  - x_j: número de padrões impressos do tipo i=1,...,4
  - t: #tampas e c: #corpos
  - y: #latinhas produzidas

  (var. inteiras)
- Restrições:
  - Tempo

    $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$
  - Materia Prima

    $x_1 + x_3 + x_4 <= 200$ Folha 1

    $x_2 <= 90$ Folha 2

    $t = 7x_1 + 3x_2 + 9x_3 + 4x_4$ e $c = x_1 + 2x_2 + 4x_4$

    $y <= t/2$

    $y <= c$

Limites superiores

- Função Objetivo: (como saber o #latinhas produzidas?)

  MAX 50y

- Cada latinha é vendida a 50c

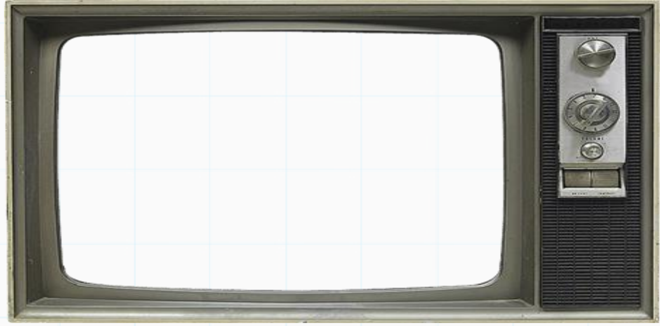
	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

tampa

tampa

corpo



- Variáveis de Decisão:
  - x_j: número de padrões impressos do tipo i=1,...,4
  - t: #tampas e c: #corpos
  - y: #latinhas produzidas

  (var. inteiras)
- Restrições:
  - Tempo

    $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$
  - Materia Prima

    $x_1 + x_3 + x_4 <= 200$ Folha 1

    $x_2 <= 90$ Folha 2

    $t = 7x_1 + 3x_2 + 9x_3 + 4x_4$ e $c = x_1 + 2x_2 + 4x_4$

    $y <= t/2$

    $y <= c$
- Função Objetivo: (como saber o #latinhas produzidas?)

  MAX 50y – (custo de estoque) ?

- Cada corpo não utilizado possui custo de estocagem de 5c e cada tampa tem custo de estocagem de 3c

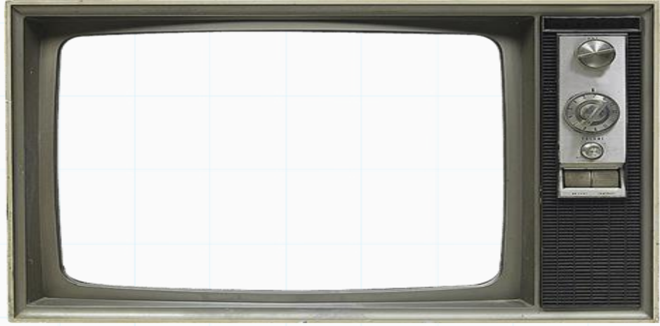
	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

tampa

corpo

tampa



- Variáveis de Decisão:
  - x_j: número de padrões impressos do tipo i=1,...,4
  - t: #tampas e c: #corpos
  - y: #latinhas produzidas

  (var. inteiras)
- Restrições:
  - Tempo

    $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$
  - Materia Prima

    $x_1 + x_3 + x_4 <= 200$ Folha 1

    $x_2 <= 90$ Folha 2

    $t = 7x_1 + 3x_2 + 9x_3 + 4x_4$ e $c = x_1 + 2x_2 + 4x_4$

    $y <= t/2$

    $y <= c$
- Função Objetivo: (como saber o #latinhas produzidas?)

  MAX $50y - 5(c - y) - 3(t - 2y)$

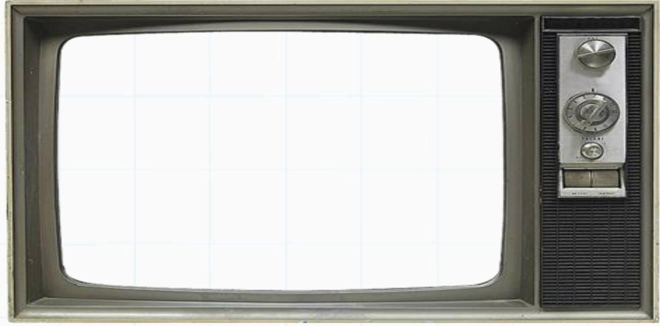
- Cada corpo não utilizado possui custo de estocagem de 5c e cada tampa tem custo de estocagem de 3c

	1	2	3	4
Tam.	1	2	1	1
#corpo	1	2	0	4
#tampa	7	3	9	4
tempo	2	3	2	1

Vamos Modelar, Inteiro ?

tampa

corpo



tampa

- Modelo final:

  MAX $50y - 5(c - y) - 3(t - 2y)$

  sujeito a:

  $2x_1 + 3x_2 + 2x_3 + x_4 <= 400$

  $x_1 + x_3 + x_4 <= 200$
  $x_2 <= 90$

  $t = 7x_1 + 3x_2 + 9x_3 + 4x_4$
  $c = x_1 + 2x_2 + 4x_4$
  $y <= t/2$
  $y <= c$

  x_i, c, t e y é inteiro, para todo i={1,2,3,4}

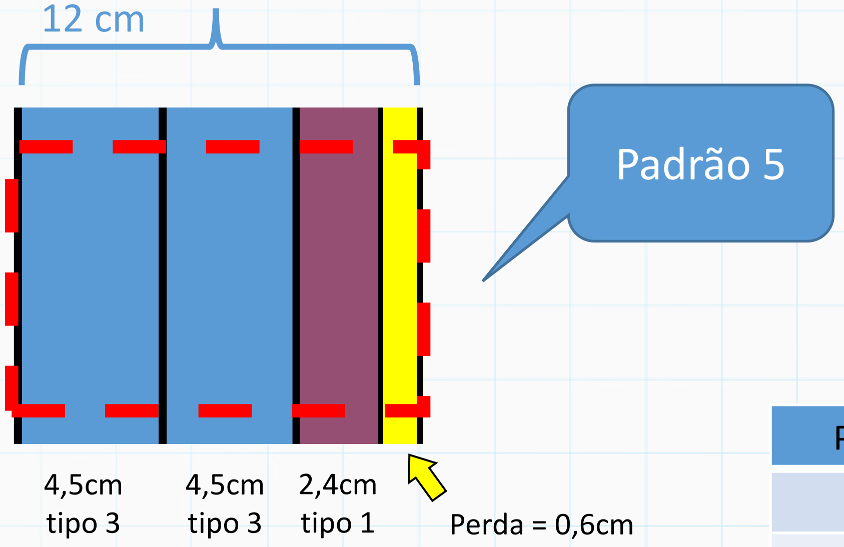
Exercícios

- Problema do corte de fitas:

- Uma fábrica necessita cortar fitas de aço de 12 cm de largura em tiras de 2,4 cm, 3,4 cm e 4,5 cm de largura. As necessidades globais de tiras de cada largura são dadas pela Tabela 1.
- A Tabela 2 indica os 7 possíveis tipos de corte da fita e quantos pedaços de cada tipo conseguimos com o corte. Além disso, cada tipo de corte gera uma perda de material.
- Queremos uma estratégia de corte (isto é, quantos cortes de cada padrão) para atender a demanda, minimizando a largura da perda de material e as fitas desnecessárias geradas (i.e. se uma fita do tipo 1 foi gerada a mais, tivemos uma perda de largura de 2,4 cm).

Ex de corte:

2 un. tipo 3
1 un. tipo 1



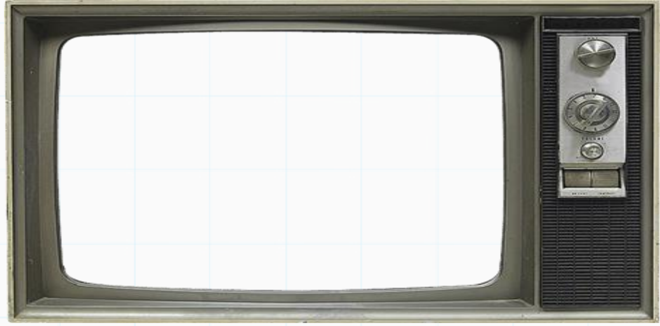
Tipo	Largura (cm)	Demanda (un)
1	2,4	2500
2	3,4	4500
3	4,5	8000

Padrão	Tiras 1	Tiras 2	Tiras 3	Perda (cm)
1	5	0	0	0
2	3	1	0	1,4
3	3	0	1	0,3
4	2	2	0	0,4
5	1	0	2	0,6
6	0	3	0	1,8
7	0	2	1	0,7

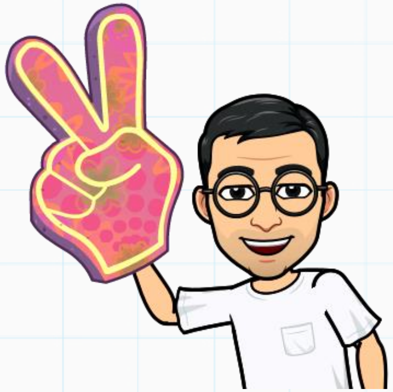
1) Variáveis: qtd de cortes
2) Restrições: uma restrição de demanda para cada tipo, e não negatividade.
3) F.O. : minimizar a largura das perdas dos cortes + largura das fitas cortadas a mais que a demanda.

Vamos Modelar, Inteiro ?

Vamos ver um exemplo de problema clássico:

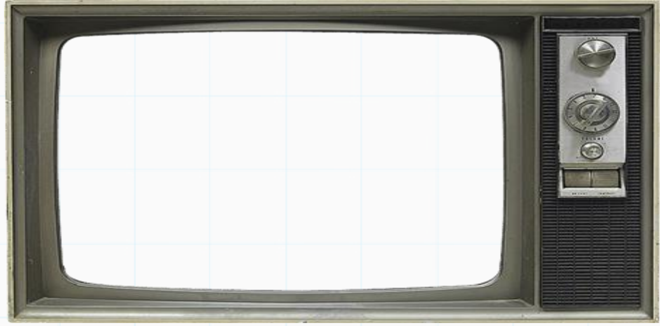


- <u>Problema de Clusterização</u>: Em sua festa de aniversário (tema teletubbies) você convidou n amigos (conjunto N). Na festa tem m mesas (conjunto M) onde seus amigos podem sentar,

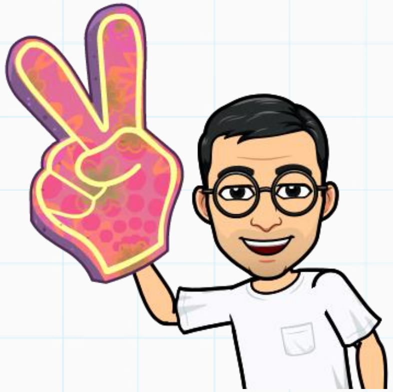


Vamos Modelar, Inteiro ?

Vamos ver um exemplo de problema clássico:

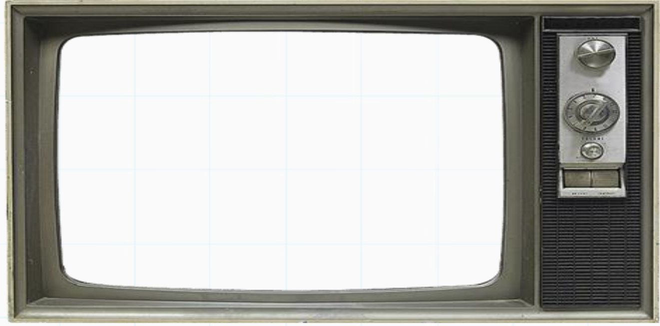


- <u>Problema de Clusterização</u>: Em sua festa de aniversário (tema teletubbies) você convidou n amigos (conjunto N). Na festa tem m mesas (conjunto M) onde seus amigos podem sentar, porém, existe inveja/amargura/rancor entre alguns pares de amigos, onde dado dois amigos i e j, o nível de ódio entre eles é dado por d_{ij} (onde $i<j$). Sua tarefa agora é escolher quais amigos vão sentar em que mesas, de forma a minimizar o nível de ódio da festa (se dois amigos i e j sentam na mesma mesa, será contabilizado o ódio d_{ij})

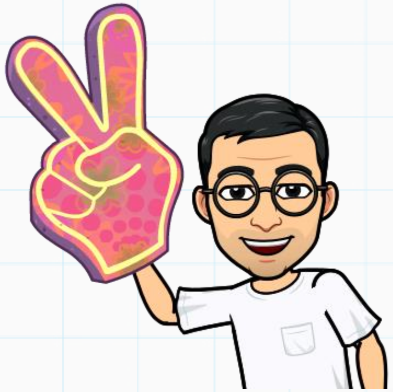


Vamos Modelar, Inteiro ?

Vamos ver um exemplo de problema clássico:

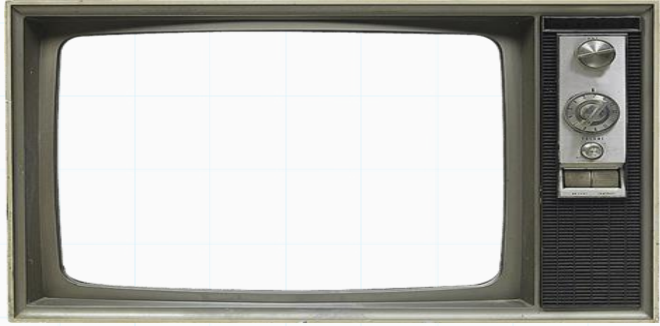


- <u>Problema de Clusterização</u>: Em sua festa de aniversário (tema teletubbies) você convidou n amigos (conjunto N). Na festa tem m mesas (conjunto M) onde seus amigos podem sentar, porém, existe inveja/amargura/rancor entre alguns pares de amigos, onde dado dois amigos i e j, o nível de ódio entre eles é dado por d_{ij} (onde $i<j$). Sua tarefa agora é escolher quais amigos vão sentar em que mesas, de forma a minimizar o nível de ódio da festa (se dois amigos i e j sentam na mesma mesa, será contabilizado o ódio d_{ij})



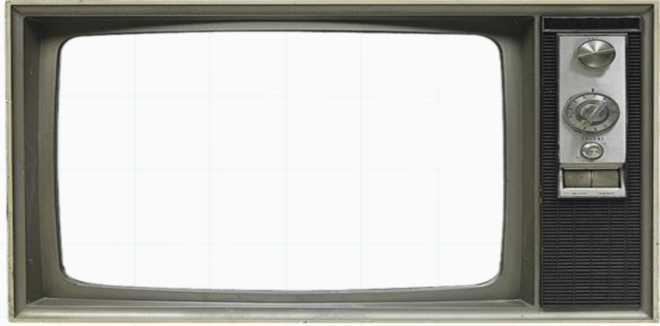
Além disso, vamos querer distribuir as pessoas na festa, então vamos impor que toda mesa tenha pelo menos uma pessoa

Vamos Modelar, Inteiro ?



- <u>Variáveis:</u>

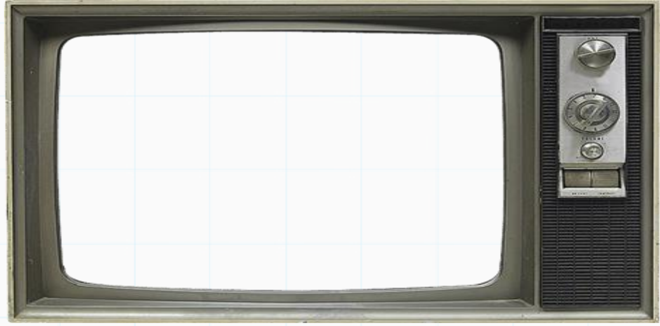
Vamos Modelar, Inteiro ?



- <u>Variáveis:</u>

$$x_{i,k} = \begin{cases} 1 & \text{se o } \overset{\text{pessoa}}{\cancel{\text{vértice}}} \text{ i pertence ao } \overset{\text{mesa}}{\cancel{\text{cluster}}}\ k. \\ 0, & \text{caso contrário.} \end{cases}$$

Vamos Modelar, Inteiro ?



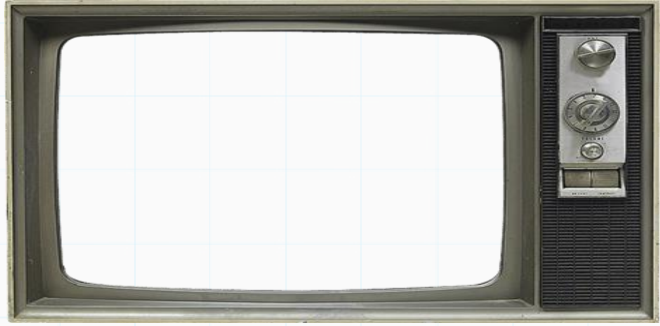
- Variáveis:

$$x_{i,k} = \begin{cases} 1 & \text{se o } \overset{\text{pessoa}}{\cancel{\text{vértice}}} \text{ i pertence ao } \overset{\text{mesa}}{\cancel{\text{cluster}}}\ k. \\ 0, & \text{caso contrário.} \end{cases}$$

- Restrições:

Todo amigo tem que sentar em alguma mesa

Vamos Modelar, Inteiro ?



- <u>Variáveis:</u>

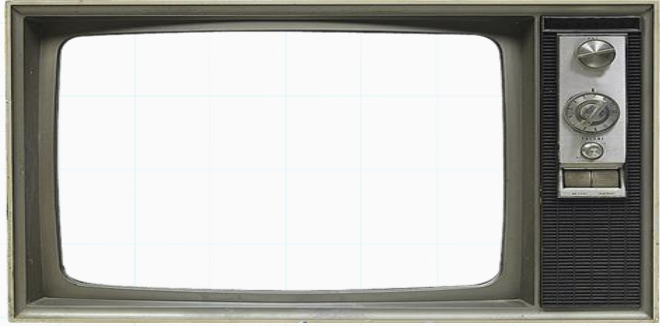
$$x_{i,k} = \begin{cases} 1 & \text{se o } \overset{\text{pessoa}}{\cancel{\text{vértice}}} \text{ i pertence ao } \overset{\text{mesa}}{\cancel{\text{cluster}}}\ k. \\ 0, & \text{caso contrário.} \end{cases}$$

- <u>Restrições:</u>

$$\sum_{k=1}^{M} x_{ik} = 1, \qquad i = 1, \ldots, N$$

Todo amigo tem que sentar em alguma mesa

Vamos Modelar, Inteiro ?



- <u>Variáveis:</u>

$$x_{i,k} = \begin{cases} 1 \;\; \text{se o } \overset{\text{pessoa}}{\cancel{\text{vértice}}} \text{ i pertence ao } \overset{\text{mesa}}{\cancel{\text{cluster}}} \; k. \\ 0, \text{caso contrário.} \end{cases}$$

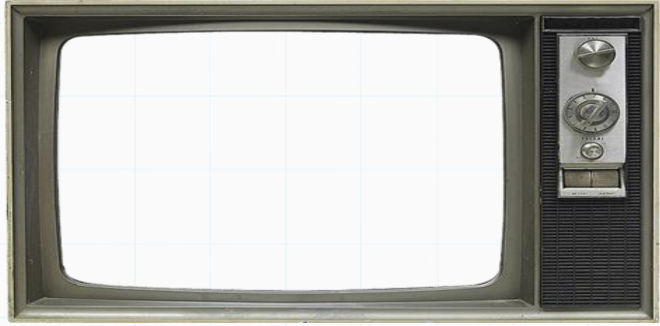
- <u>Restrições:</u>

$$\sum_{k=1}^{M} x_{ik} = 1, \qquad i = 1, \ldots, N$$

Todo amigo tem que sentar em alguma mesa

Toda mesa tem uma pessoa

Vamos Modelar, Inteiro ?



- Variáveis:

$$x_{i,k} = \begin{cases} 1 \;\; \text{se o } \overset{\text{pessoa}}{\cancel{\text{vértice}}} \text{ i pertence ao } \overset{\text{mesa}}{\cancel{\text{cluster}}}\ k. \\ 0, \text{caso contrário.} \end{cases}$$

- Restrições:

$$\sum_{k=1}^{M} x_{ik} = 1, \qquad i = 1, \ldots, N$$

Todo amigo tem que sentar em alguma mesa

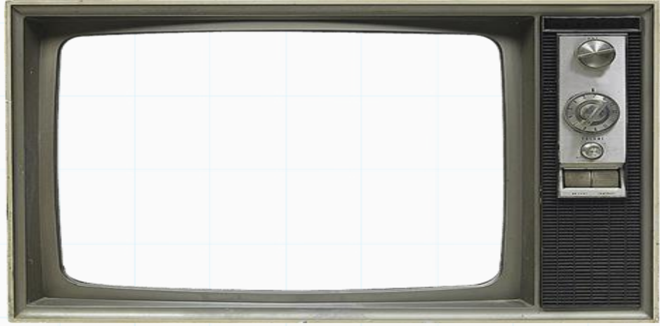
$$\sum_{i=1}^{N} x_{ik} \geq 1, \qquad k = 1, \ldots, M$$

Toda mesa tem uma pessoa

$$x_{ik} \in \{0, 1\}, \qquad i = 1, \ldots, N \qquad k = 1, \ldots, M$$

integralidade

Vamos Modelar, Inteiro ?



- Variáveis:

$$x_{i,k} = \begin{cases} 1 & \text{se o } \cancel{\text{vértice}}^{\text{ pessoa}} \text{ i pertence ao } \cancel{\text{cluster}}^{\text{ mesa}}\ k. \\ 0, \text{caso contrário.} \end{cases}$$

- Restrições:

$$\sum_{k=1}^{M} x_{ik} = 1, \qquad i = 1, \ldots, N$$

Todo amigo tem que sentar em alguma mesa

$$\sum_{i=1}^{N} x_{ik} \geq 1, \qquad k = 1, \ldots, M$$

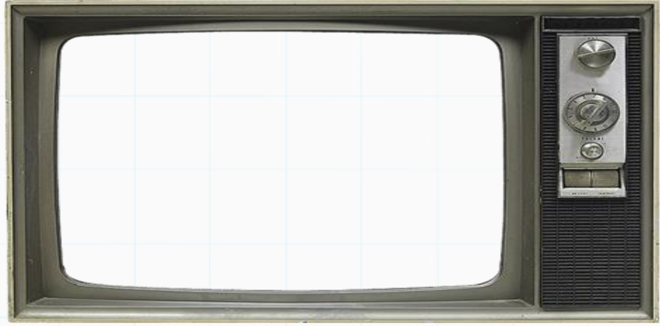
Toda mesa tem uma pessoa

$$x_{ik} \in \{0, 1\}, \qquad i = 1, \ldots, N \qquad k = 1, \ldots, M$$

integralidade

- Função Objetivo:

Vamos Modelar, Inteiro ?



- Variáveis:

$$x_{i,k} = \begin{cases} 1 \ \text{ se o } \overset{\text{pessoa}}{\cancel{\text{vértice}}} \text{ i pertence ao } \overset{\text{mesa}}{\cancel{\text{cluster}}}\ k. \\ 0, \text{caso contrário.} \end{cases}$$

$$\boxed{y_{i,j} = \begin{cases} 1 \ \text{ se os vértices } i \text{ e } j \text{ pertencem ao mesmo cluster.} \\ 0, \text{caso contrário.} \end{cases}}$$

- Restrições:

$$\sum_{k=1}^{M} x_{ik} = 1, \qquad i = 1, \ldots, N$$

Todo amigo tem que sentar em alguma mesa

$$\sum_{i=1}^{N} x_{ik} \geq 1, \qquad k = 1, \ldots, M$$

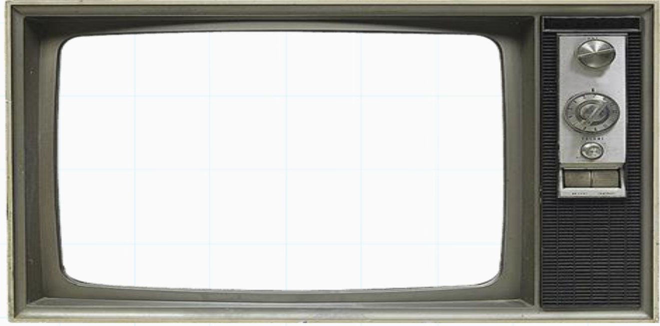
Toda mesa tem uma pessoa

$$x_{ik} \in \{0, 1\}, \qquad i = 1, \ldots, N \qquad k = 1, \ldots, M$$

integralidade

- Função Objetivo:

Vamos Modelar, Inteiro ?



- Variáveis:

$$x_{i,k} = \begin{cases} 1 \ \text{ se o } \overset{\text{pessoa}}{\cancel{\text{vértice}}} \text{ i pertence ao } \overset{\text{mesa}}{\cancel{\text{cluster}}} \ k. \\ 0, \text{caso contrário.} \end{cases}$$

$$\boxed{y_{i,j} = \begin{cases} 1 \ \text{ se os vértices } i \text{ e } j \text{ pertencem ao mesmo cluster.} \\ 0, \text{caso contrário.} \end{cases}}$$

- Restrições:

$$\sum_{k=1}^{M} x_{ik} = 1, \qquad i = 1, \ldots, N$$

Todo amigo tem que sentar em alguma mesa

$$\sum_{i=1}^{N} x_{ik} \geq 1, \qquad k = 1, \ldots, M$$

Toda mesa tem uma pessoa

$$x_{ik} \in \{0, 1\}, \qquad i = 1, \ldots, N \qquad k = 1, \ldots, M$$

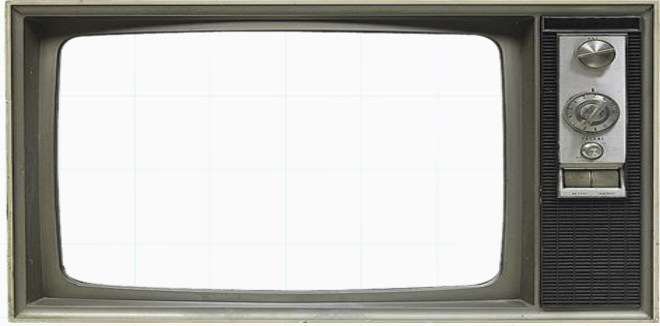
integralidade

- Função Objetivo:

$$Min \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij} \, y_{ij}$$

Mas como impor os valores corretos das variáveis y

Vamos Modelar, Inteiro ?



- Variáveis:

$$x_{i,k} = \begin{cases} 1 \ \text{ se o } \overset{\text{pessoa}}{\cancel{\text{vértice}}} \text{ i pertence ao } \overset{\text{mesa}}{\cancel{\text{cluster}}}\ k. \\ 0, \text{caso contrário.} \end{cases}$$

$$\boxed{y_{i,j} = \begin{cases} 1 \ \text{ se os vértices } i \text{ e } j \text{ pertencem ao mesmo cluster.} \\ 0, \text{caso contrário.} \end{cases}}$$

- Restrições:

$$\sum_{k=1}^{M} x_{ik} = 1, \qquad i = 1, \ldots, N$$

$$\sum_{i=1}^{N} x_{ik} \geq 1, \qquad k = 1, \ldots, M$$

$$\boxed{\begin{aligned} & y_{ij} \geq x_{ik} + x_{jk} - 1, \\ & \qquad i = 1, \ldots N \\ & \qquad j = i+1, \ldots, N \\ & \qquad k = i, \ldots, M \end{aligned}}$$

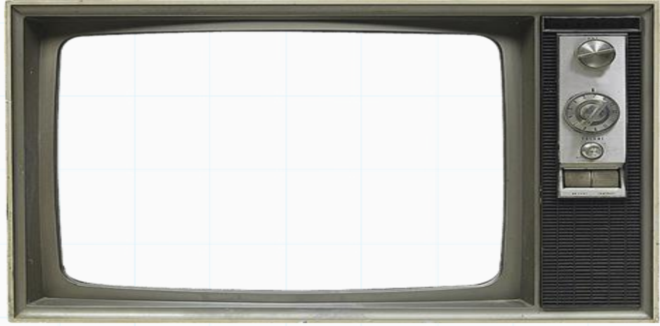
$$x_{ik} \in \{0, 1\}, \qquad i = 1, \ldots, N \qquad k = 1, \ldots, M$$

- Função Objetivo:

$$Min \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij}\, y_{ij}$$

Mas como impor os valores corretos das variáveis y

Vamos Modelar, Inteiro ?



- <u>Modelo completo:</u>

$$Min \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij}\, y_{ij}$$

$$\sum_{k=1}^{M} x_{ik} = 1, \qquad i = 1, \ldots, N$$

$$\sum_{i=1}^{N} x_{ik} \geq 1, \qquad k = 1, \ldots, M$$

$$y_{ij} \geq x_{ik} + x_{jk} - 1, \qquad i = 1, \ldots N \quad j = i+1, \ldots, N \qquad k = i, \ldots, M$$

$$x_{ik} \in \{0, 1\}, \qquad i = 1, \ldots, N \qquad k = 1, \ldots, M$$

Até a próxima