# Programação Inteira

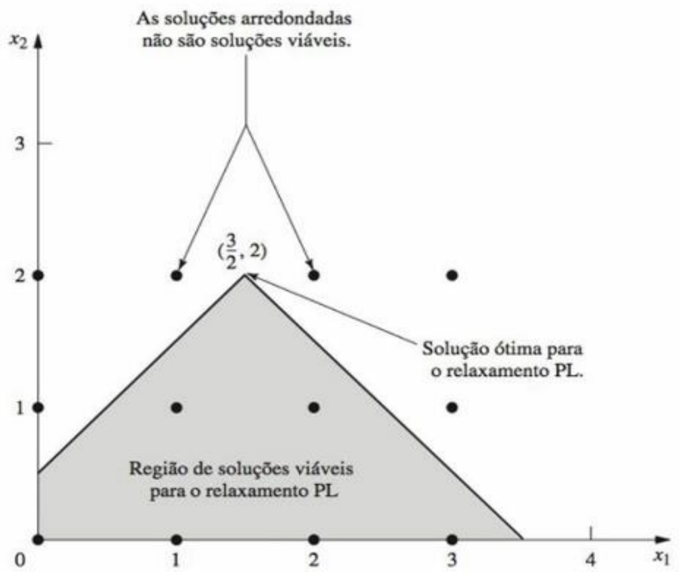
- Diversas classes de problemas exigem que os valores obtidos em um problemas de PL seja inteiros (Hillier , 2013):
  - Seleção entre diversas alternativas
  - Decisões do tipo fazer ou não fazer
  - Geração de portafólios de investimento
  - Localização de facilities (fábricas, armazéns, postos de atendimento, etc)
  - Problemas de partição
  - Problemas de alocação
  - Problemas de seqüenciamento
  - Problemas de roteamento
  - etc

- Programação Linear Inteira
  - Mista : Quando pelo menos uma variável deve ter valor inteiro e as demais são continuas
  - Pura: Quando todas as variáveis devem ser inteiras
  - Binária: as variáveis só podem assumir valores 0 ou 1

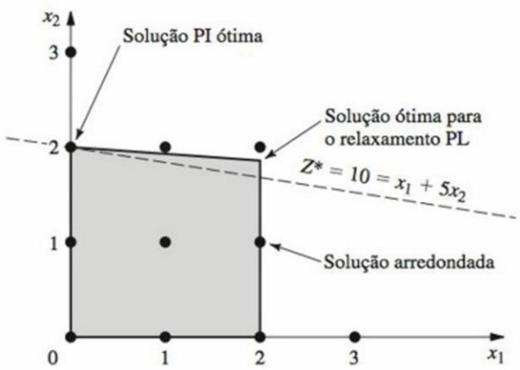
# Programação Inteira

- Quando uma varável deve ser inteira mas a solução fornece um resultado não inteiro não podemos simplesmente arredondar ou truncar o valor
  - A solução obtida pode não ser viável
  - A solução obtida pode não ser ótima

Maximizar $Z = x_1 + 5x_2$,




■ FIGURA 11.2 Um exemplo de um problema de PI em que a solução ótima para o relaxamento PL não pode ser arredondada de nenhuma maneira que possa manter a viabilidade.




■ FIGURA 11.3 Um exemplo no qual o arredondamento da solução ótima para o relaxamento PL está longe de ser uma solução ótima para o problema de PI.

  - Fonte: Hillier, 2005

# Programação Inteira

- Exemplo

max $z = 2x_1 - x_2$

$$
\text{S. A.} \quad
\begin{aligned}
x_1 - x_2 &\leq 1 \\
2x_1 + x_2 &\leq 6 \\
x_1, x_2 &\geq 0 \text{ e } x_2 \text{ inteiro}
\end{aligned}
$$

- Deixamos de lado a necessidade de $x_2$ inteiro (problema relaxado), chamamos este problema de $P_0$ e resolvemos o mesmo (método gráfico ou SIMPLEX):

$x_1 = 2,33$
$x_2 = 1,33$
$z = 3,33$

# Programação Inteira

- Como podemos verificar $x_2 = 1{,}33$, ou seja seu valor está em uma região não inteira (entre 1 e 2)
- Para "forçar" que o valor de $x_2$ seja inteiro, iremos eliminar a possibilidade de que o resultado ocorra nessa faixa de valores, o que nos leva a $(x_2 \leq 1)$ ou $(x_2 \geq 2)$
- Como essas duas restrições não podem ser atendidas simultaneamente iremos "ramificar" o problema $P_0$ em dois novos problemas $P_1$ e $P_2$, um com a restrição $(x_2 \leq 1)$ e outro com a restrição $(x_2 \geq 2)$:

$P_1$:

$$\max \ z = 2x_1 - x_2$$

$$\text{S. A.} \quad \begin{cases} x_1 - x_2 \leq 1 \\ 2x_1 + x_2 \leq 6 \\ x_2 \leq 1 \\ x_1, \ x_2 \geq 0, \ x_2 \text{ inteiro} \end{cases}$$

$P_2$:

$$\max \ z = 2x_1 - x_2$$

$$\text{S. A.} \quad \begin{cases} x_1 - x_2 \leq 1 \\ 2x_1 + x_2 \leq 6 \\ x_2 \geq 2 \\ x_1, \ x_2 \geq 0, \ x_2 \text{ inteiro} \end{cases}$$

# Programação Inteira

- Sempre que adicionamos uma restrição a um problema existente, o valor da função objetivo tende a piorar sendo que na melhor das hipóteses ele poderá ser igual ao anterior mas nunca será melhor.
- Essa propriedade é útil para que possamos limitar a busca de possíveis soluções ótimas na árvore de soluções já que em problemas com mais variáveis inteiras pode haverá um grande número de combinações possíveis de valores o que poderia tornar a busca bastante custosa
- Resolvendo $P_1$ e $P_2$ teremos:

| | |
|---|---|
| $P_1$: | $P_1$: |
| $x_1 = 2$ | $x_1 = 2$ |
| $x_2 = 1$ | $x_2 = 2$ |
| $z = 3$ | $z = 2$ |

# Programação Inteira

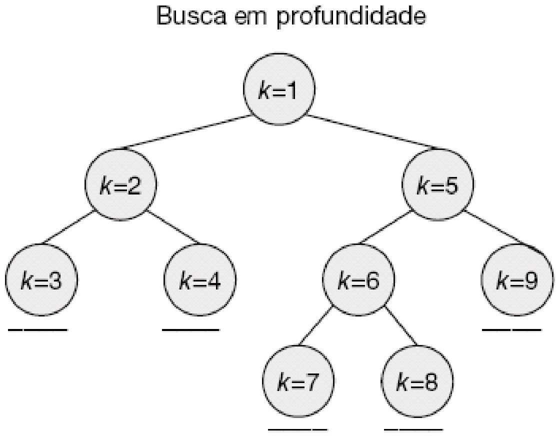
- Como ambos forneceram valores inteiros para $x_2$, escolhemos aquele que da o melhor resultado para a função objetivo, isto é, a resposta 1.

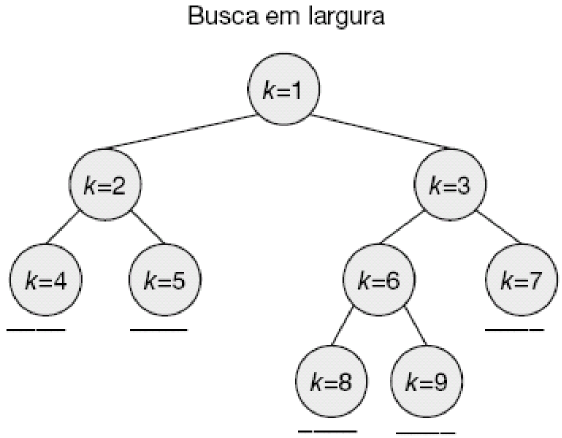
- Se, por hipótese, a resposta 2 fosse $x_2 = 2,5$, também não precisariamos fazer nova ramificação, uma vez que o valor para a FO já é pior do que o da resposta 1.

- Porém,se ainda com $x_2$ não inteiro o valor da FO fosse 3,3 (por exemplo) , valeria a pena fazer mais uma ramificação, pois o valor da FO poderia ser melhor do que a resposta 1

# Programação Inteira

- Branch and Bound – Partição e Avaliação Sucessiva, método de solução mais usado
  - Branch: escolha do subproblema a tratar
  - Bound: estabelecimento do limite inferior( maximização) ou superior (minimização) para busca da melhor solução
- A partição (ramificação) é feita apenas uma variável, se houver mais de uma candidata escolhe-se arbitrariamente
- A busca pode ser em largura ou profundidade




Fonte:

# Programação Inteira

Sejam $x^*_j \notin I$ a solução ótima de um problema de PPL,
$\lfloor x^*_j \rfloor$ o maior inteiro menor que $x^*_j$
$\lceil x^*_j \rceil$ o menor inteiro maior que $x^*_j$

Algoritmo:

Passo 1 Gere uma lista de PPL's inicialmente vazia. Resolva o PPL desconsiderando as restrições de integridade. Se este problema não tem solução viável, então PARE. O PPLIM também não terá solução viável. Em caso contrário, inclua este PPL na lista.

Passo 2 Escolha, entre os PPL's da lista, aquele cujo valor da solução ótima seja o máximo. Se mais de um problema atender esta condição, desempate de qualquer forma, mas sempre a favor de soluções que satisfaças as condições de integridade do PPLIM.

Passo 3 Retire o problema escolhido da lista e, se a sua solução satisfizer as condições de integridade, PARE. A solução ótima do PPLIM foi encontrada. Em caso contrário, tome uma variável inteira que não satisfaz a condição de integridade, isto é $x^*_j \notin Z$, e gere dois novos PPL's, pela agregação das seguintes restrições adicionais:

Problema 1 $x_j \leq \lfloor x^*_j \rfloor$

Problema 2 $x_j \geq \lceil x^*_j \rceil$

Resolva cada PPL gerado, e se tiver solução ótima, inclua na lista.

Passo 4 Se a lista estiver vazia, então PARE, pois o PPLIM não tem solução viável. Em caso contrário volte ao Passo 2.

# Programação Inteira

- Exemplo:

$\max \ 3x_1+3x_2+13x_3$

s.a:

$-3x_1+6x_2+7x_3 \leq 8$

$6x_1-3x_2+7x_3 \leq 8$

$0 \leq x_1, x_2, x_3 \leq 5$

$x_1, x_2, x_3 \in I$

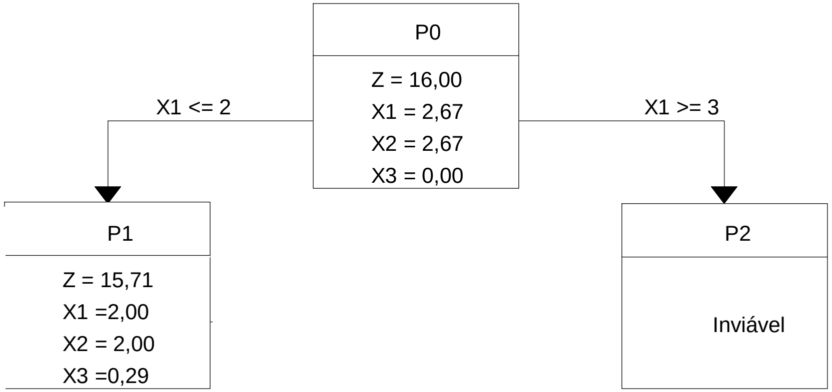
# Programação Inteira

| Problema | | X1 | X2 | X3 | Z | Obs |
|---|---|---|---|---|---|---|
| Núm. | Ant. | Valor | Valor | Valor | | |
| 0 | - | 2,67 | 2,67 | 0,00 | 16,00 | PL original |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

| P0 |
|---|
| Z = 16,00<br>X1 = 2,67<br>X2 = 2,67<br>X3 = 0,00 |

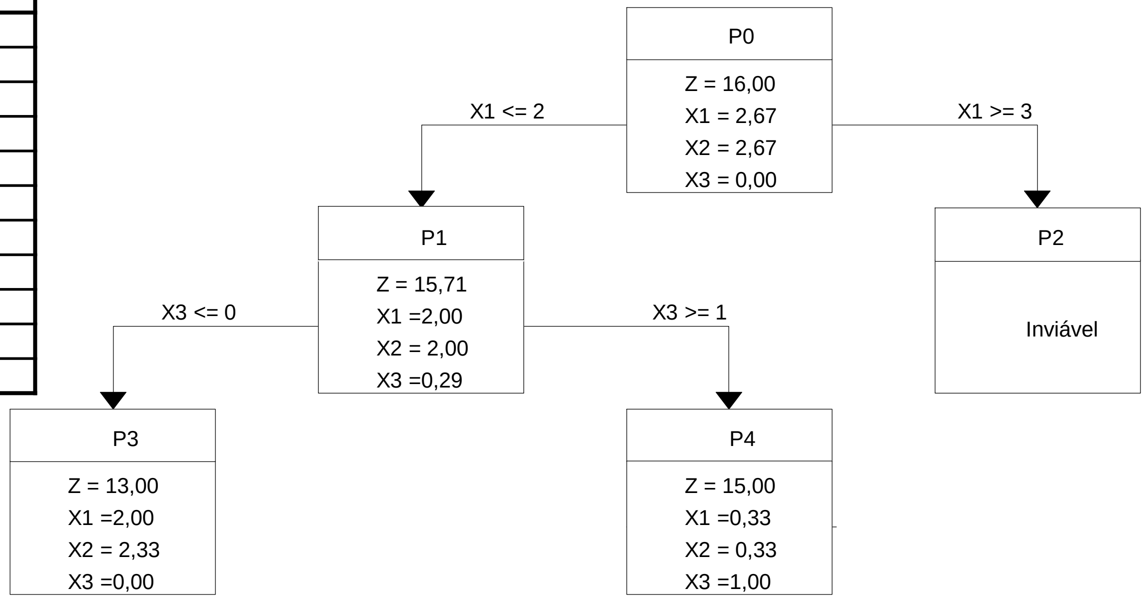
# Programação Inteira

| Problema | | X1 | X2 | X3 | Z | Obs |
|---|---|---|---|---|---|---|
| Núm. | Ant. | Valor | Valor | Valor | | |
| 0 | - | 2,67 | 2,67 | 0,00 | 16,00 | PL original |
| 1 | 0 | 2,00 | 2,00 | 0,29 | 15,71 | Ok |
| 2 | 0 | | | | | Inviavel |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |



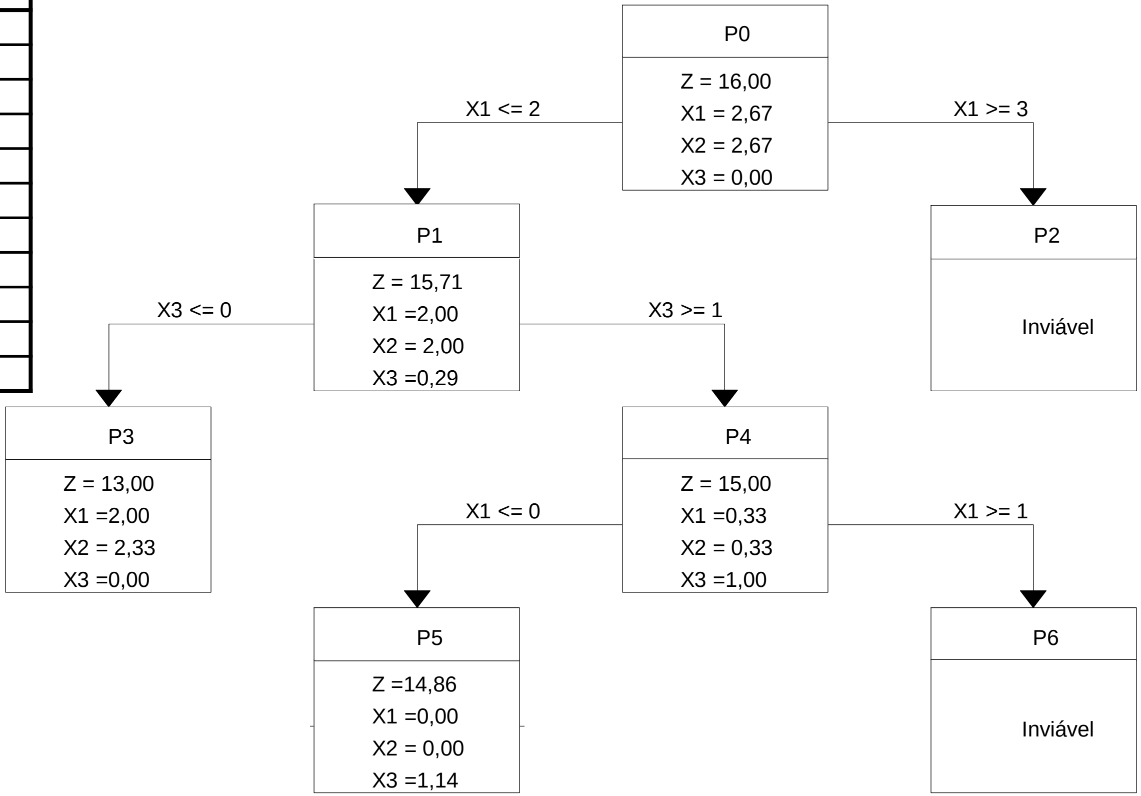
# Programação Inteira

| Problema | | X1 | X2 | X3 | Z | Obs |
|---|---|---|---|---|---|---|
| Núm. | Ant. | Valor | Valor | Valor | | |
| 0 | - | 2,67 | 2,67 | 0,00 | 16,00 | PL original |
| 1 | 0 | 2,00 | 2,00 | 0,29 | 15,71 | Ok |
| 2 | 0 | | | | | Inviavel |
| 3 | 1 | 2,00 | 2,33 | 0,00 | 13,00 | |
| 4 | 1 | 0,33 | 0,33 | 1,00 | 15,00 | Ok |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |



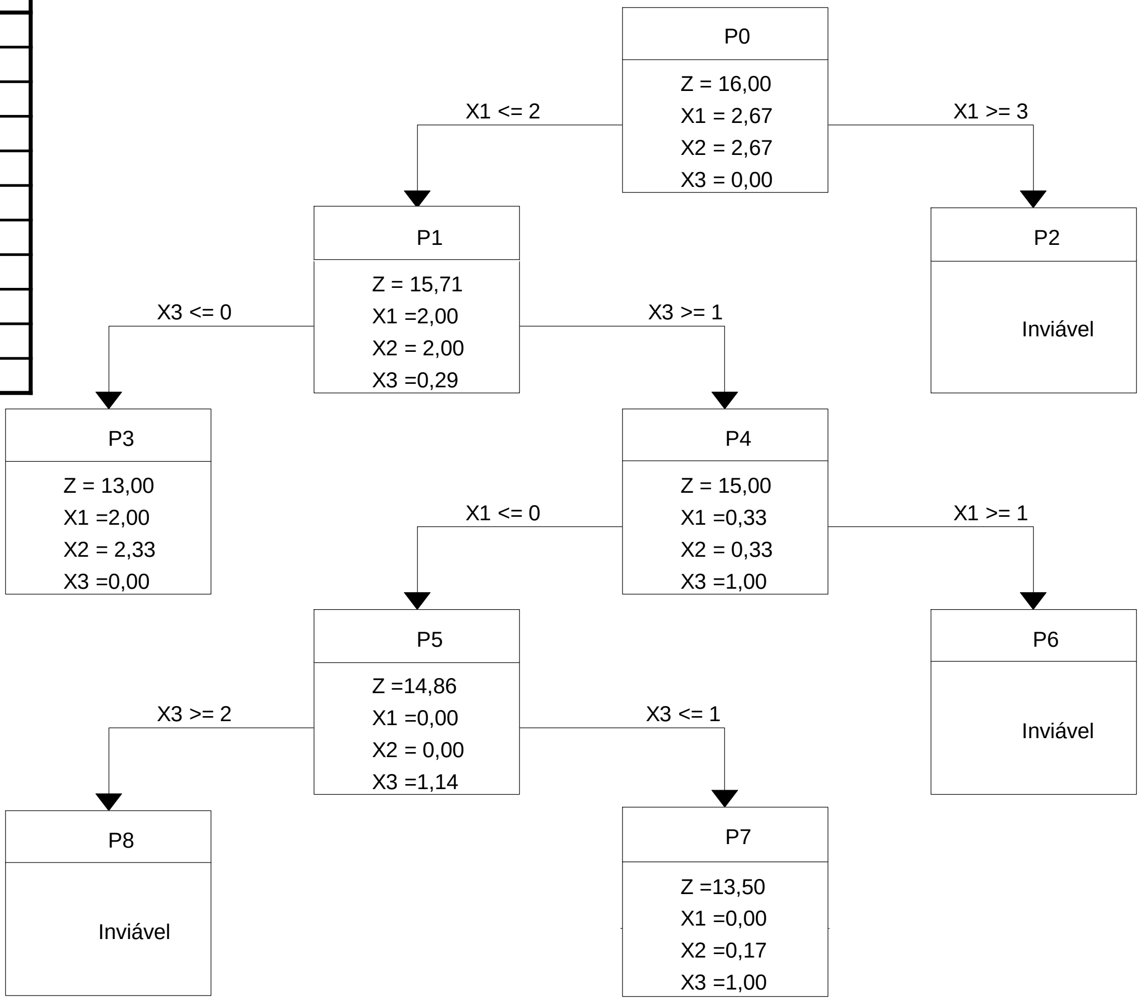
# Programação Inteira

| Problema | | X1 | X2 | X3 | Z | Obs |
|---|---|---|---|---|---|---|
| Núm. | Ant. | Valor | Valor | Valor | | |
| 0 | - | 2,67 | 2,67 | 0,00 | 16,00 | PL original |
| 1 | 0 | 2,00 | 2,00 | 0,29 | 15,71 | Ok |
| 2 | 0 | | | | | Inviavel |
| 3 | 1 | 2,00 | 2,33 | 0,00 | 13,00 | |
| 4 | 1 | 0,33 | 0,33 | 1,00 | 15,00 | Ok |
| 5 | 4 | 0,00 | 0,00 | 1,14 | 14,86 | Ok |
| 6 | 4 | | | | | Inviavel |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |



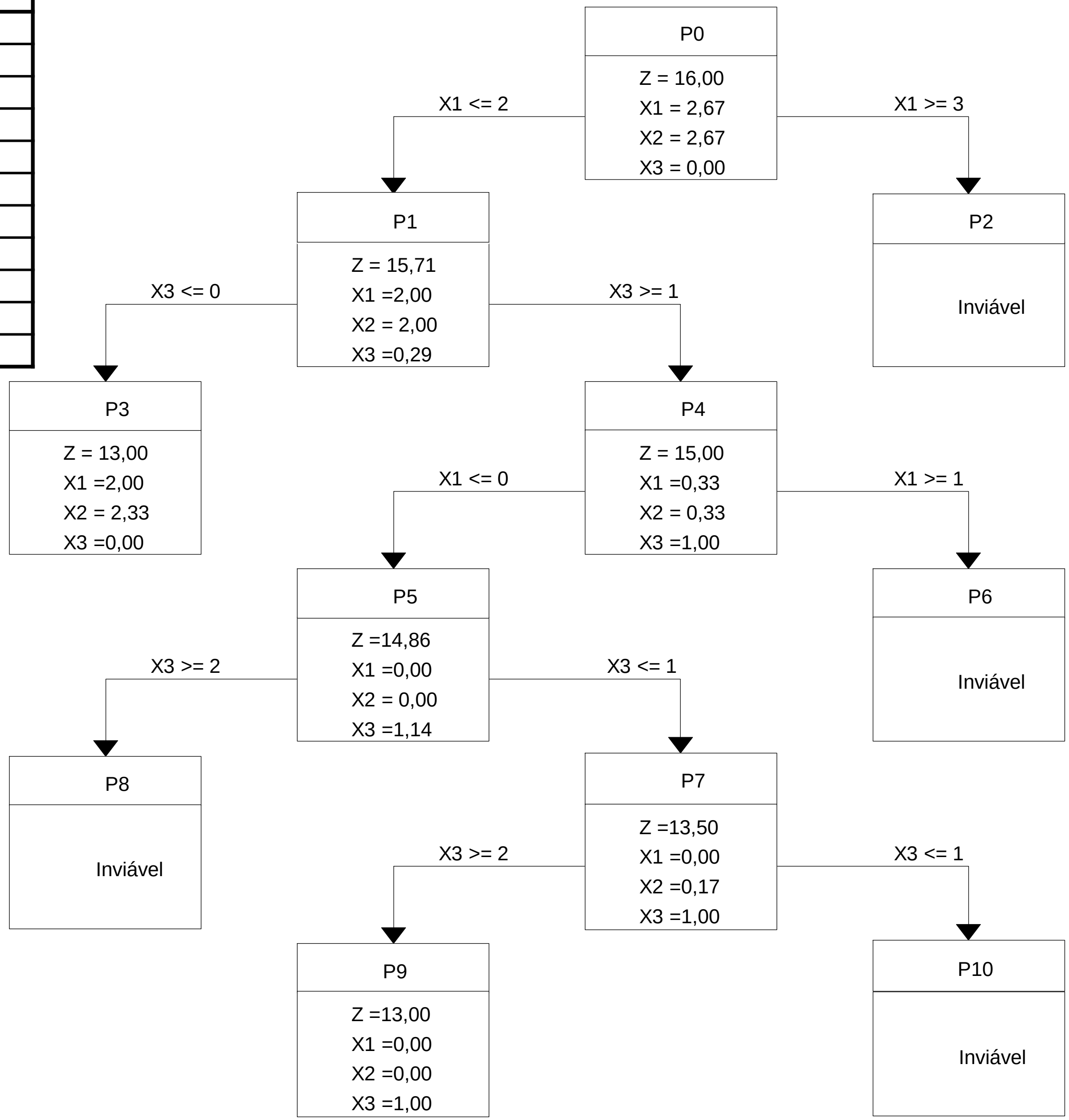
# Programação Inteira

| Problema | | X1 | X2 | X3 | Z | Obs |
|---|---|---|---|---|---|---|
| Núm. | Ant. | Valor | Valor | Valor | | |
| 0 | - | 2,67 | 2,67 | 0,00 | 16,00 | PL original |
| 1 | 0 | 2,00 | 2,00 | 0,29 | 15,71 | Ok |
| 2 | 0 | | | | | Inviavel |
| 3 | 1 | 2,00 | 2,33 | 0,00 | 13,00 | |
| 4 | 1 | 0,33 | 0,33 | 1,00 | 15,00 | Ok |
| 5 | 4 | 0,00 | 0,00 | 1,14 | 14,86 | Ok |
| 6 | 4 | | | | | Inviavel |
| 7 | 5 | 0,00 | 0,17 | 1,00 | 13,50 | Ok |
| 8 | 5 | | | | | Inviavel |
| | | | | | | |
| | | | | | | |



# Programação Inteira

| Problema | | X1 | X2 | X3 | Z | Obs |
|---|---|---|---|---|---|---|
| **Núm.** | **Ant.** | **Valor** | **Valor** | **Valor** | | |
| 0 | - | 2,67 | 2,67 | 0,00 | 16,00 | PL original |
| 1 | 0 | 2,00 | 2,00 | 0,29 | 15,71 | Ok |
| 2 | 0 | | | | | Inviavel |
| 3 | 1 | 2,00 | 2,33 | 0,00 | 13,00 | |
| 4 | 1 | 0,33 | 0,33 | 1,00 | 15,00 | Ok |
| 5 | 4 | 0,00 | 0,00 | 1,14 | 14,86 | Ok |
| 6 | 4 | | | | | Inviavel |
| 7 | 5 | 0,00 | 0,17 | 1,00 | 13,50 | Ok |
| 8 | 5 | | | | | Inviavel |
| 9 | 7 | 0,00 | 0,00 | 1,00 | 13,00 | Sol. ótima |
| 10 | 7 | | | | | Inviavel |



# Programação Inteira

- Bibliografia:
  – HILLIER, Frederick S. Introdução à pesquisa operacional. 9. Porto Alegre AMGH 2013 recurso online ISBN 9788580551198.