Otimização de Sistemas

Prof. Sandro Jerônimo de Almeida, PhD.

Programação Linear Inteira

Definição

- Um problema de Programação Linear Inteira (PLI) é um problema de Programação Linear (PL) em que todas ou alguma(s) das suas variáveis são discretas (têm de assumir valores inteiros)
- Programação Linear Inteira Pura: todas variáveis devem ser inteiras
- Programação Linear Inteira Mista: algumas variáveis devem ser inteiras

Exemplos

Programação Linear

$$\max \ F = 4x_1 - 5x_2$$

suj. a:

$$2x_1 + 3x_2 \leq 8$$
$$5x_1 + 2x_2 \leq 11$$
$$x_1, x_2 \geq 0$$

Programação Linear Inteira

$$\max \ F = 4x_1 - 5x_2$$

suj. a:

$$2x_1 + 3x_2 \leq 8$$
$$5x_1 + 2x_2 \leq 11$$
$$x_1, x_2 \geq 0 \text{ e inteiras}$$

- Exemplos de variáveis de decisão inteiras

  Número de Pessoas, Número de Lotes (indivisíveis)

  Qtde. de Itens, Variáveis binárias, etc.

Problema da Mochila

- Quais itens colocar em uma mochila de forma a levar o maior valor possível e não ultrapassar a capacidade K da mochila?

Item	Valor ($)	Peso (Kg)
1	100	1,2
2	200	2
3	300	3,1
4	400	4,2
5	500	4,8
6	600	5,9
7	700	6,9

Capacidade da Mochila
K = 11 kg

Problema da Mochila

- Caso específico

Item	Valor ($)	Peso (Kg)
1	100	1,2
2	200	2
3	300	3,1
4	400	4,2
5	500	4,8
6	600	5,9
7	700	6,9

Capacidade da Mochila
K = 11 kg

Problema da Mochila

Item	Valor ($)	Peso (Kg)
1	100	1,2
2	200	2
3	300	3,1
4	400	4,2
5	500	4,8
6	600	5,9
7	700	6,9

Capacidade da Mochila
K = 11 kg

- Modelo de PPL

- Variáveis de decisão

X_i = decide se leva o item ou não (0 ou 1), | i={1,2,3 ...7}

- Maximizar $Z = 100\,X_1 + 200\,X_2 + 300\,X_3 + 400\,X_4 + 500\,X_5 + 600\,X_6 + 700\,X_7$

Sujeito a:

$1{,}2\,X_1 + 2\,X_2 + 3{,}1\,X_3 + 4{,}2\,X_4 + 4{,}8\,X_5 + 5{,}9\,X_6 + 6{,}9\,X_7 <= 11$ (kg)

$x1, x2, x3, x4, x5, x6, x7 \in \{0,1\}$

Problema da Mochila

- Formulação geral - *Knapsack problem (PK)*

$$\text{(PK) Maximizar } z = \sum_{j=1}^{n} c_j x_j$$

sujeito a:

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

$$x_j \geq 0 \text{ e inteiro.}$$

Problema da Mochila

- *Apenas 1 item - (PKI)*

Problema da Mochila

- *Apenas 1 item disponível - (PKI)*

$$\text{(PKI) Maximizar } z = \sum_{j=1}^{n} c_j x_j$$

sujeito a:

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

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

Problema da Mochila

- *Quantidade limitada de Itens - (PKL)*

Problema da Mochila

- *Quantidade limitada de Itens - (PKL)*

$$\text{(PKL) Maximizar } z = \sum_{j=1}^{n} c_j x_j$$

sujeito a:

$$\sum_{j=1}^{n} \mathrm{w}_j x_j \leq b$$

$$x_j \leq l_j \qquad j = 1, \ldots, n$$

$$x_j \in \mathfrak{I}^+$$

Exemplo da Confeitaria

Uma confeitaria pode produzir dois tipos de sorvete em lata: chocolate e creme. Cada lata do sorvete de chocolate é vendido com um lucro de $3 e as latas de creme com um lucro de $1. Contratos com várias lojas impõem que sejam produzidas no mínimo 10 latas de sorvete de chocolate por dia e que o total de latas fabricadas por dia nunca seja menor que 20. O mercado só é capaz de consumir até 40 latas de sorvete de creme e 60 de chocolate. As máquinas de preparação do sorvete disponibilizam 180 horas de operação por dia, sendo que cada lata de sorvete de chocolate consome 2 horas de trabalho e cada lata de creme, 3 horas. Determine o esquema de produção que maximiza os lucros com a venda de latas de sorvete.

Exemplo da Confeitaria

- Variáveis de decisão:

x_j = número de latas de sorvete do tipo *j* a serem produzidas por dia, sendo *j* = 1 (chocolate) e *j* = 2 (creme)

- Função objetivo: maximizar lucro obtidos como a venda dos lotes dos bolos tipo creme e chocolate

$$\text{Maximizar } f(x) = x_1 + 3x_2$$

Exemplo da Confeitaria

Modelo PL

Maximizar $z = x_1 + 3x_2$

sujeito a:

$$
\begin{aligned}
x_1 \qquad\quad &\le 40 \\
x_2 &\le 60 \\
x_2 &\ge 10 \\
x_1 + x_2 &\ge 20 \\
3x_1 + 2x_2 &\le 180 \\
x_1 \ge 0,\ x_2 &\ge 0
\end{aligned}
$$

Pergunta

- Sendo x_j = número de latas de sorvete do tipo *j* a serem produzidas por dia.

- É possível produzir um número não inteiro de latas de sorvete?

Exemplo da Confeitaria

Modelo requer restrições de integralidade

$$
\begin{aligned}
&\text{Maximizar } z = x_1 + 3x_2 \\
&\text{sujeito a:} \\
&\qquad x_1 \leq 40 \\
&\qquad x_2 \leq 60 \\
&\qquad x_2 \geq 10 \\
&\qquad x_1 + x_2 \geq 20 \\
&\qquad 3x_1 + 2x_2 \leq 180 \\
&\qquad x_1 \geq 0,\ x_2 \geq 0
\end{aligned}
$$

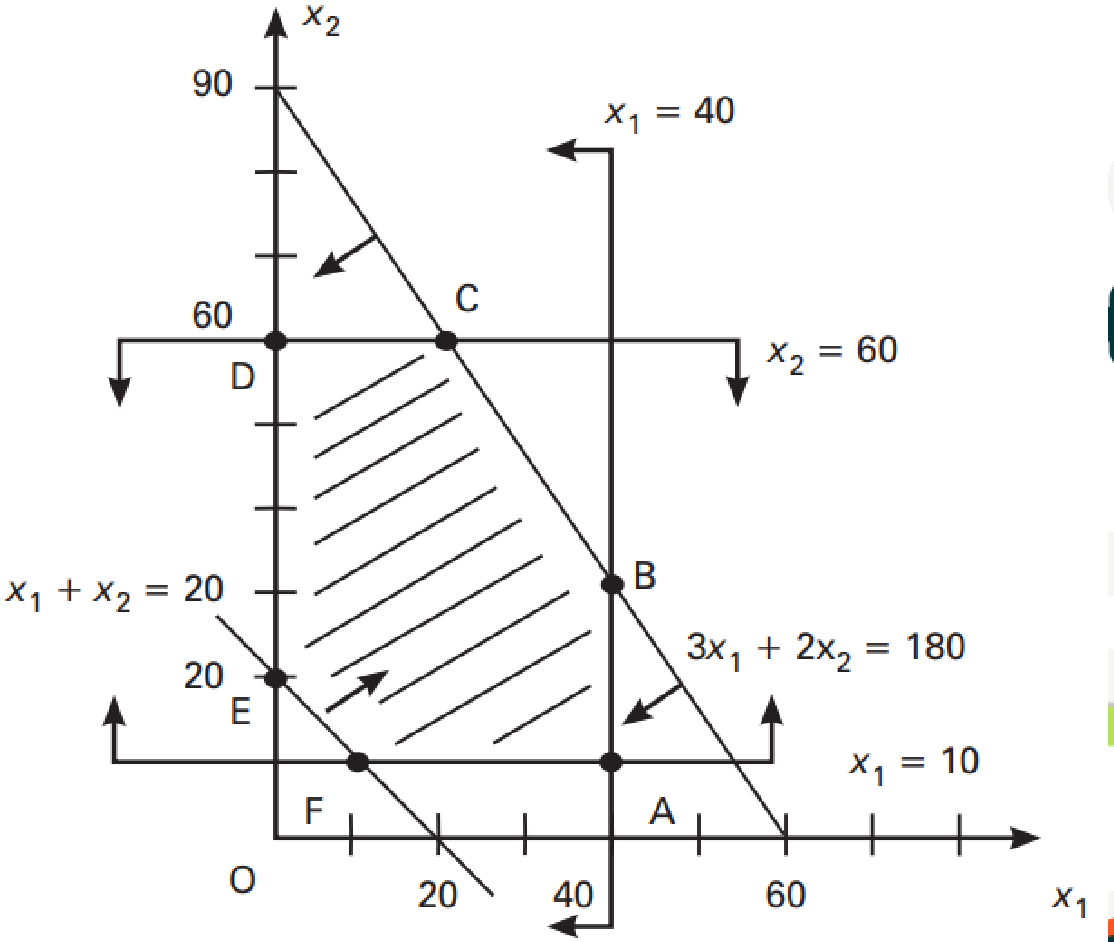
$x_1, x_2 \in Z^+$ (conjunto dos inteiros positivos).

Exemplo da Confeitaria

Representação gráfica

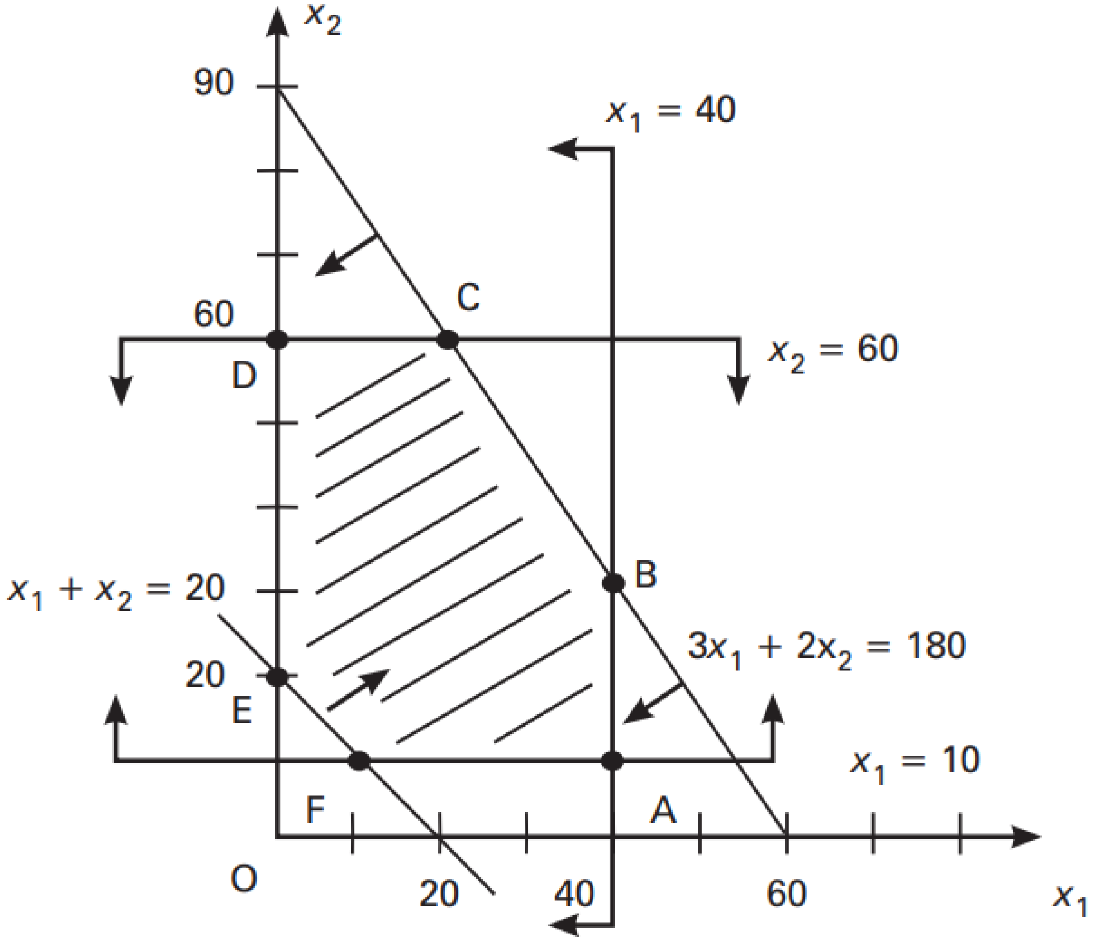
Maximizar $z = x_1 + 3x_2$
sujeito a:

$$
\begin{aligned}
x_1 \quad\quad &\leq 40 \\
x_2 &\leq 60 \\
x_2 &\geq 10 \\
x_1 + x_2 &\geq 20 \\
3x_1 + 2x_2 &\leq 180 \\
x_1 \geq 0,\ x_2 &\geq 0
\end{aligned}
$$



Exemplo da Confeitaria

Representação gráfica



Ponto	Coordenada	Max Z = x1 + 3x2
A	(40, 10)	70
B	(40,30)	130
C	(20,60)	200
D	(0, 60)	180
E	(0, 20)	60
F	(10,10)	40

Pergunta-se:

- E se os pontos não fossem inteiros?
- Qual seria a solução?

Soluções Inteiras

Exemplo

$$
\begin{array}{lccccc}
\max & x_1 & + & 19x_2 & & \\
 & x_1 & + & 20x_2 & \leq & 50 \\
 & x_1 & + & x_2 & \leq & 20 \\
 & x_1 & , & x_2 & \in & Z^+
\end{array}
$$

Solução ótima não inteira

- $X_1 = 18{,}89$
- $X_2 = 1{,}58$
- $Z = 48{,}42$

- Como obter a solução ótima inteira?
- O arredondamento seria uma boa estratégia?

Estratégia de arredondamento

$$
\begin{array}{lccccc}
\max & x_1 & + & 19x_2 & & \\
 & x_1 & + & 20x_2 & \leq & 50 \\
 & x_1 & + & x_2 & \leq & 20 \\
 & x_1 & , & x_2 & \in & Z^+
\end{array}
$$

X_1	X_2	$Z = X_1 + 19\,X_2$
19	2	Inviável
19	1	Z = 38
18	2	Inviável
18	1	Z = 37

Solução ótima

- $X_1 = 18{,}89$
- $X_2 = 1{,}58$
- $Z = 48{,}42$

Existe uma solução melhor?

Estratégia de arredondamento

$$\begin{array}{lcccccc} \max & x_1 & + & 19x_2 & & \\ & x_1 & + & 20x_2 & \leq & 50 \\ & x_1 & + & x_2 & \leq & 20 \\ & x_1 & , & x_2 & \in & Z^+ \end{array}$$

X_1	X_2	$Z = X_1 + 19\,X_2$
19	2	Inviável
19	1	Z = 38
18	2	Inviável
18	1	Z = 37

Erro de 21%

Solução ótima

- $X_1 = 18{,}89$
- $X_2 = 1{,}58$
- $Z = 48{,}42$

Solução ótima Inteira

- $X_1 = 10$
- $X_2 = 2$
- $Z = 48$

Conclusão: Não é uma boa estratégia resolver o PPL (contínuo) e arredondar a solução resultante

Branch-and-Bound

- O método denominado de *Branch-and-Bound (B&B)* baseia-se na ideia de desenvolver uma enumeração inteligente dos pontos candidatos à solução ótima inteira de um problema.
- *Branch:* refere-se ao fato de que o método efetua partições no espaço das soluções.
- Bound: ressalta que a prova da otimalidade da solução utiliza-se de limites calculados ao longo da enumeração

Definições

- Seja o problema (P) com variáveis inteiras e $(\overline{P})$ com variáveis contínuas:

$$(P) = \text{Maximizar } \{cx \mid Ax = b,\ x \geq 0,\ x \in Z^{+}\}$$

$$\text{e}$$

$$(\overline{P}) = \text{Maximizar } \{cx \mid Ax = b,\ x \geq 0,\ x \in R^{+}\}$$

- Temos que a solução ótima de P será sempre menor igual a solução ótima de $\overline{P}$

Branch-and-Bound (B&B)

- Obtém-se a solução do problema P
- Se a solução não é inteira significa que o problema deve ser subdividido *(branch)* em dois novos P1 e P2, com as seguintes restrições:

$$x_j \geq \lfloor \bar{x}_j \rfloor + 1 \text{ ou } x_j \leq \lfloor \bar{x}_j \rfloor$$

- Os problemas P1 e P2 devem ser resolvidos
- Sucessivas subdivisões devem acontecer até se obter uma solução inteira.

Branch-and-Bound (B&B)

- Subproblemas obtidos a partir da inserção das novas restrições

$$
\begin{aligned}
&(P_1) \text{ Maximizar } z = cx \\
&\quad \text{sujeito a:} \\
&\qquad Ax \leq b \\
&\qquad x_i \leq \lfloor x_i^0 \rfloor \\
&\qquad x_i \in Z
\end{aligned}
$$

$$
\begin{aligned}
&(P_2) \text{ Maximizar } z = cx \\
&\quad \text{sujeito a:} \\
&\qquad Ax \leq b \\
&\qquad x_i \leq \lfloor x_i^0 \rfloor + 1 \\
&\qquad x_i \in Z
\end{aligned}
$$

Branch-and-Bound (B&B)

Exemplo

$$\text{Maximizar} \quad z = 5x_1 + 8x_2$$

$$\text{sujeito a}:$$

$$x_1 + x_2 \leq 6$$

$$5x_1 + 9x_2 \leq 45$$

$$x_1, x_2 \in Z^+$$

Solução Contínua: $x_1 = \dfrac{9}{4}$ $\quad x_2 = \dfrac{15}{4}$ $\quad Z = 41\dfrac{1}{4}$

Branch-and-Bound (B&B)

Divisões do problema (*branch*)

- Solução contínua $x_1 = \frac{9}{4}$ $x_2 = \frac{15}{4}$ $Z = 41\frac{1}{4}$

- Restrições a serem adicionadas ao

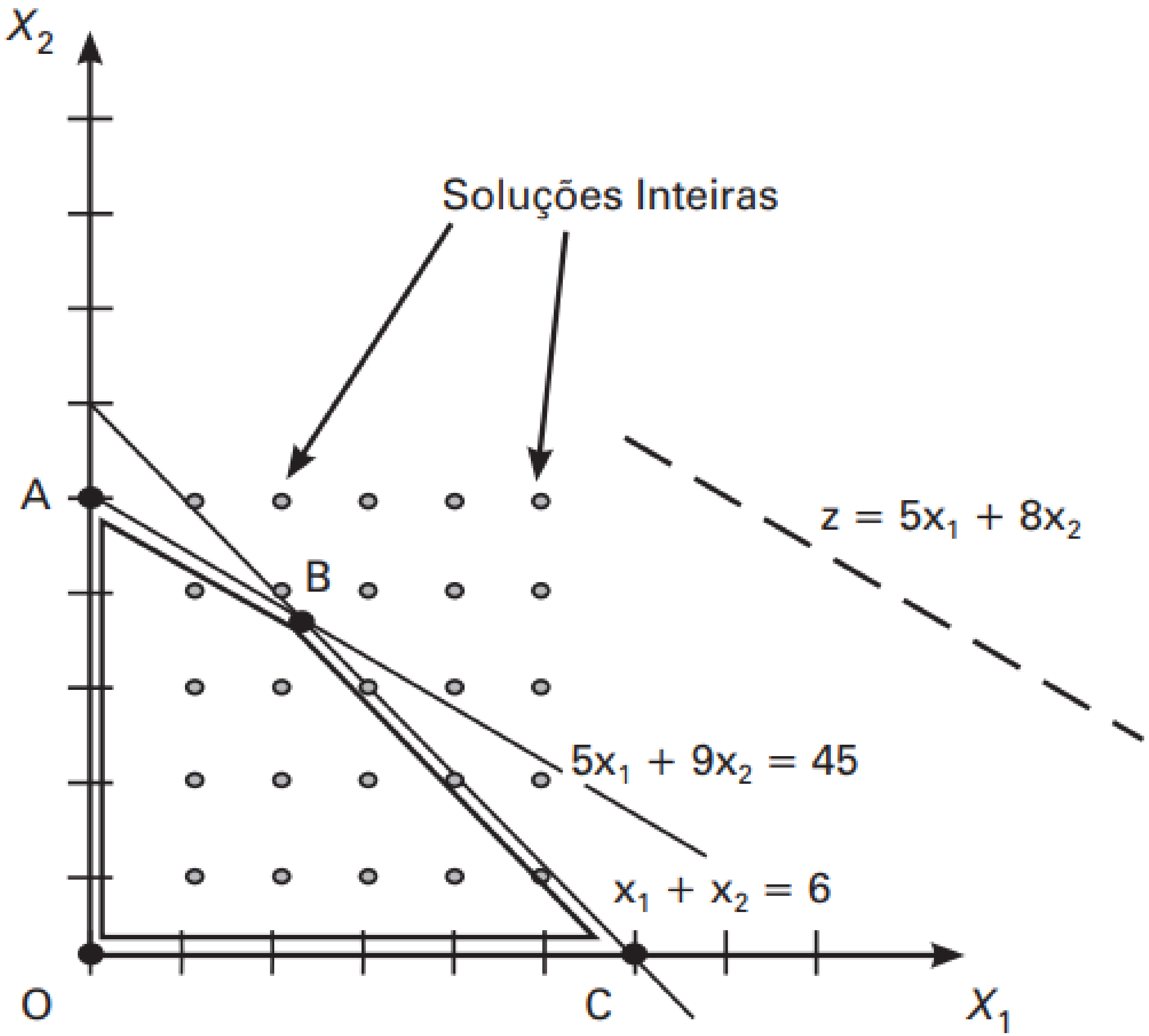
Problema (P1)

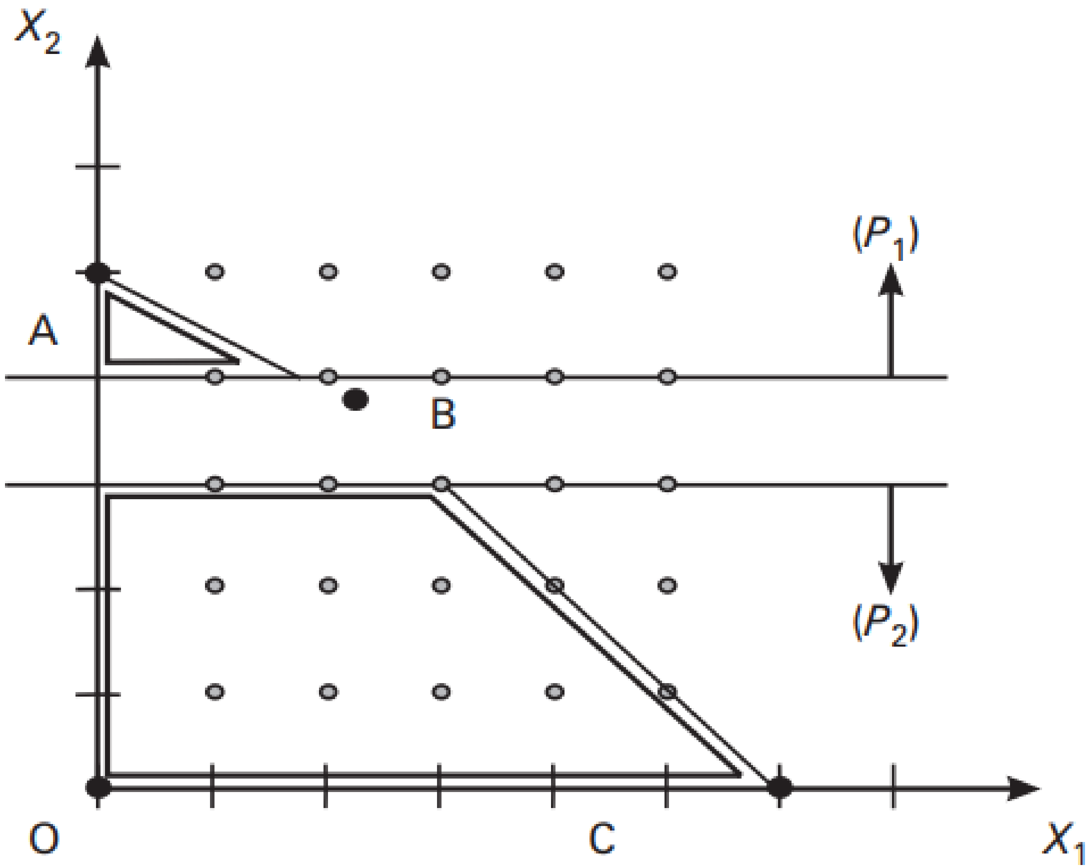
$$x_2 \geq \left\lfloor \frac{15}{4} \right\rfloor + 1 \geq 4$$

Problema (P2)

$$x_2 \leq \left\lfloor \frac{15}{4} \right\rfloor \leq 3$$

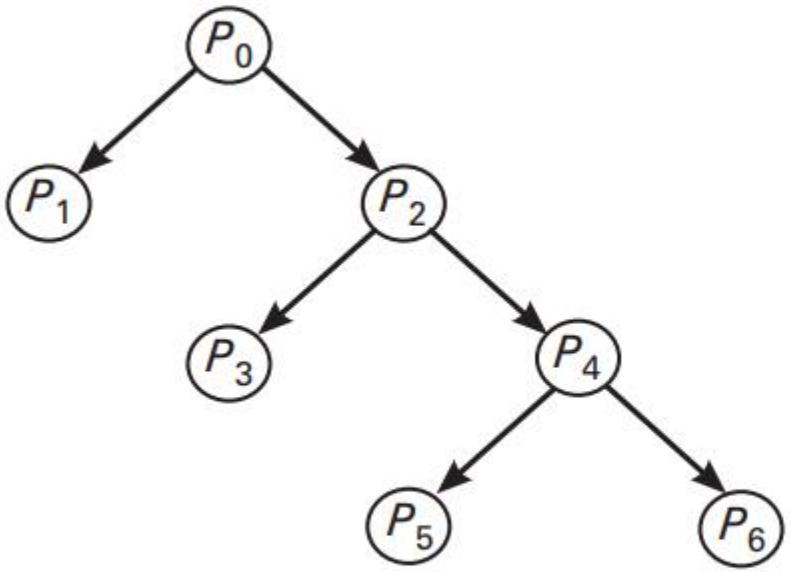
Branch-and-Bound (B&B)



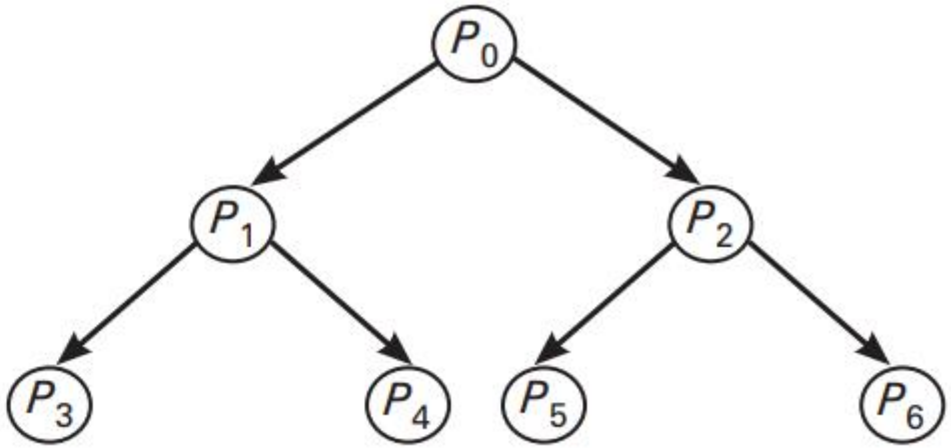


Branch-and-Bound (B&B)

Iterações e estratégias de subdivisões



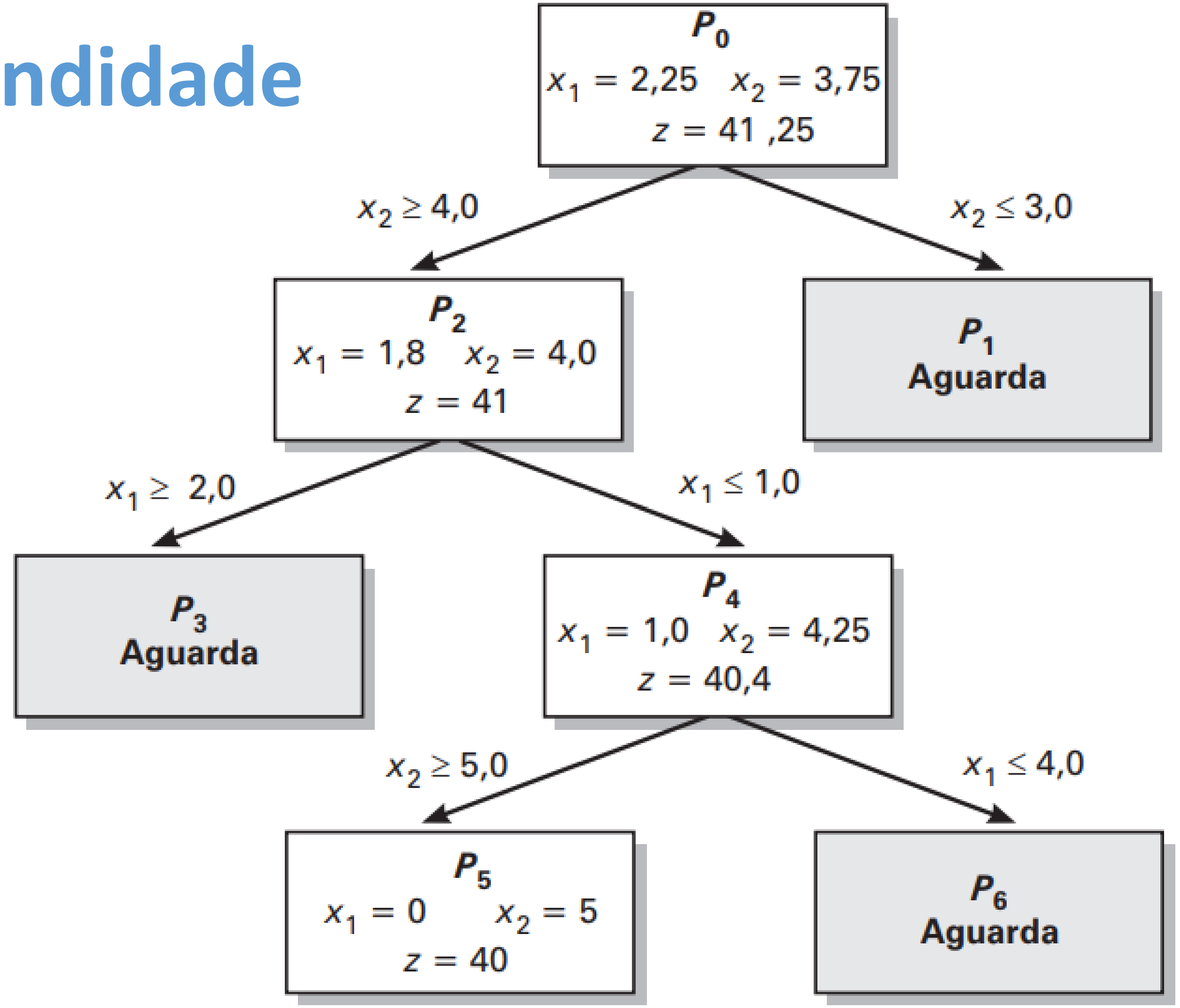
Busca em Profundidade



Busca em Largura

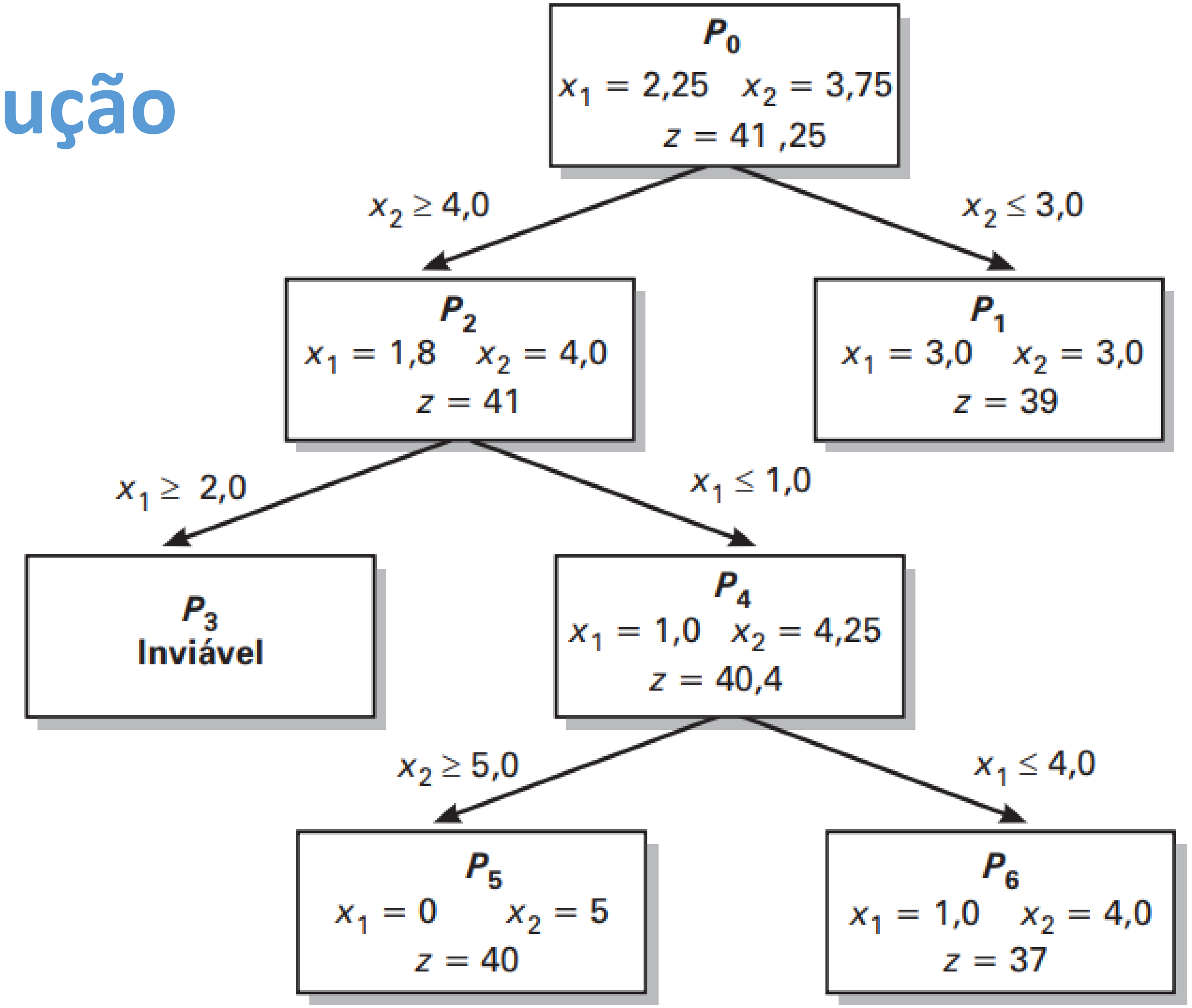
Branch-and-Bound (B&B)

Busca em Profundidade



Branch-and-Bound (B&B)

Resultado da execução



Branch-and-Bound (B&B)

Informações sobre o exemplo

- As soluções contínuas são um limite superior para o valor de z0, sob as condições estabelecidas nos vértices da árvore, enquanto as soluções inteiras geram um limite inferior. Como (P4), um problema com solução contínua, possui Z = 40,4 e (P5), um problema com solução inteira, possui Z = 40, o problema (P6) não precisa mais ser solucionado, uma vez que entre 40, 4 e 40 não existe a possibilidade de uma outra solução inteira melhor que 40 ($40 \leq z0 \leq 40{,}4$). O problema (P2), com z = 41 pode dar origem, contudo, ainda a um problema com uma solução inteira de valor 41 ($40 \leq z \leq 41$), o que obriga ao desenvolvimento de (P3). De modo semelhante, (P0), com z = 41,25 pode dar origem a um problema com a solução também de valor 41 ($40 \leq z0 \leq 41, 25$), o que obriga ao desenvolvimento de (P1).
- A redução pelo limite inferior (*bound*) de apenas um vértice da árvore de enumeração do exemplo pode parecer pequena, mas devemos lembrar que esse problema é pequeno também. Em muitos casos reais, o poder de simplificação do limite inferior (ou superior no problema de minimização) se mostra significativo, sendo extremamente útil no processo de solução.

Branch-and-Bound (B&B)

Variações e implementações

- Técnicas de desenvolvimento da árvore de enumeração:

  Busca em profundidade, Busca em largura, Variantes híbridas

- Técnicas de formação da árvore (escolha da variável de separação):

Variante de Dank (1960), Variante de Land e Doig (1965), Variante de Spielberg (1968), Método das Penalidades (1965), Método de Taha (1971), Estratégias dinâmicas (1976), outras variantes

Branch-and-Bound (B&B)

Variações e implementações

- Técnicas complementares para obtenção dos limites:
  Relaxação linear (ver Pirce [1964], Bagchi et al. [1996], Pardulos et al. [1996]). – Relaxação lagrangeana (ver Fisher [1981], Fisher [1985] e Beasley [1985], Desrosiers et al. [1988], Kohl e Madsen [1997] e Holmberg e Yuan [2000]). – Algoritmos heurísticos e meta-heurísticos (ver Puckinger [2004]). – Cortes.

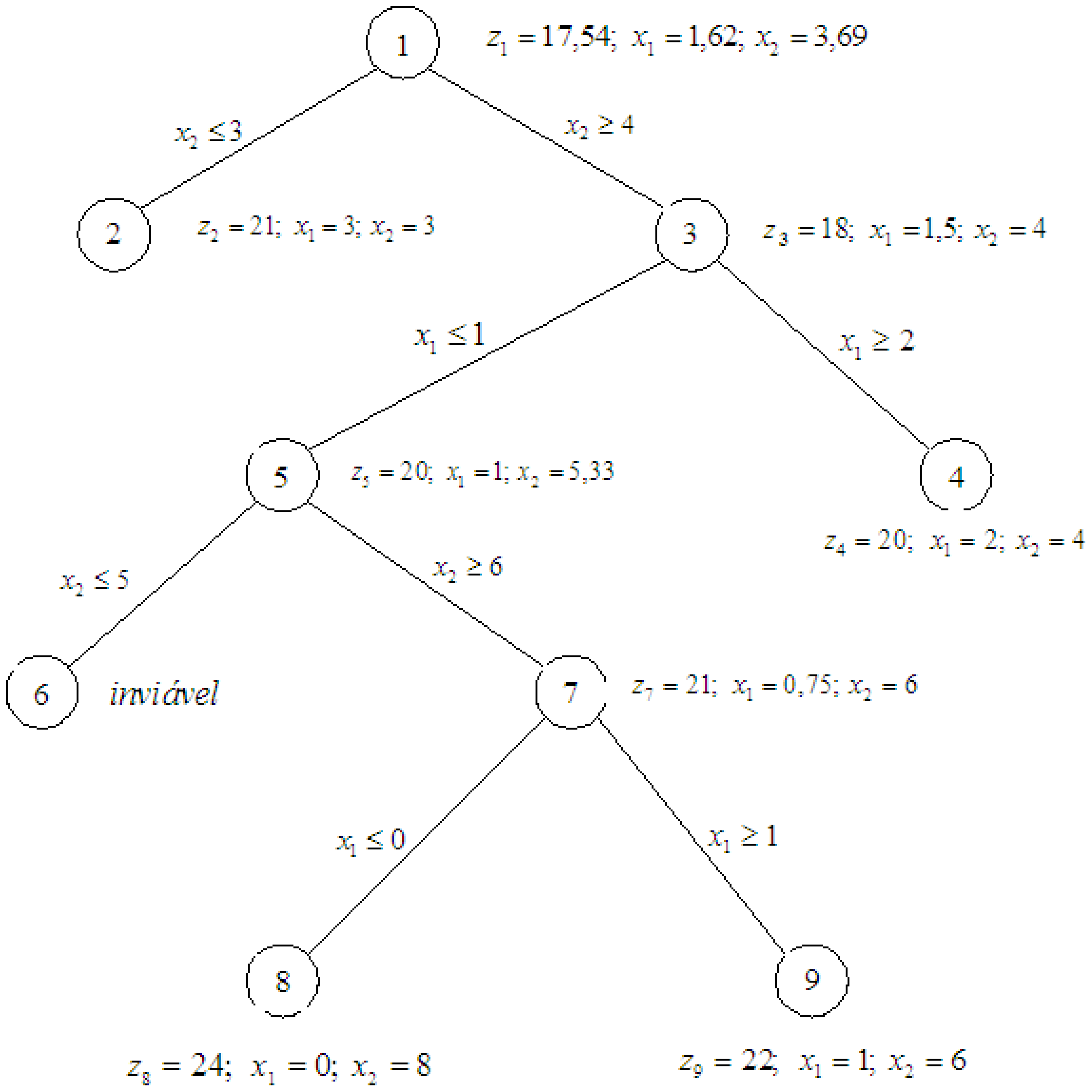
Exercício

- Resolva pelo método *Branch-and-Bound* o PLI abaixo

(P1)
$$\min z = 4x_1 + 3x_2$$
$$8x_1 + 3x_2 \geq 24$$
$$5x_1 + 6x_2 \geq 30$$
$$x_1 + 2x_2 \geq 9$$
$$x_1, x_2 \in Z^+$$

- Use a variante de *Dank* para decidir a variável a ramificar (Nessa variante, a variável a ramificar é aquela cujo valor está mais próximo de um valor inteiro)
- Em caso de empate, escolha a de menor índice
- Use busca em profundidade e analise primeiro o valor maior da variável ramificada, isto é, o valor $x_j \geq \lfloor x_j \rfloor + 1$

Exercício – Árvore de *Branch*



Exercício – Árvore de *Branch*