

Programação Inteira

Solução com o método ***Branch-and-Bound***

Conceitos gerais

- Um conceito fundamental nos métodos de resolução de programação inteira é a relaxação linear
 - Consiste em substituir PIM $y \in \mathbb{Z}_+^p$ por $y \in \mathbb{R}_+^p$
- Problema de programação inteira
$$(P) \quad z = \max \{c^T x : Ax \leq b, x \in \mathbb{Z}_+^n\}$$
- e sua relaxação linear
$$(P) \quad \bar{z} = \max \{c^T x : Ax \leq b, x \in \mathbb{R}_+^n\}$$
- O valor ótimo do problema de PL é um limitante superior do problema de programação inteira P, isto é, $\bar{z} \geq z$

Método *Branch-and-Bound*

- Considere os problemas P e PL: se o poliedro
$$\overline{P} = \{Ax \leq b, x \in R_+^n\}$$
- é fechado, então o número de soluções inteiras factíveis é finito
- Formas de se obter a solução ótima
 - Enumeração completa: calcular o valor de todas as soluções factíveis e escolher o maior
 - Enumeração implícita: subconjuntos de soluções são implicitamente considerados e descartados pois não contêm uma solução ótima ou então contêm soluções ótimas distintas
 - Subconjuntos são obtidos pela estratégia dividir-para-conquistar usada no método *branch-and-bound*

Método *Branch-and-Bound*

- Exemplo:

$$(P) \quad z = \max 5x_1 - x_2$$

$$7x_1 - 5x_2 \leq 13$$

$$3x_1 + 2x_2 \leq 17$$

$$x \in Z_+^2$$

- Solução ótima real do problema relaxado PL,

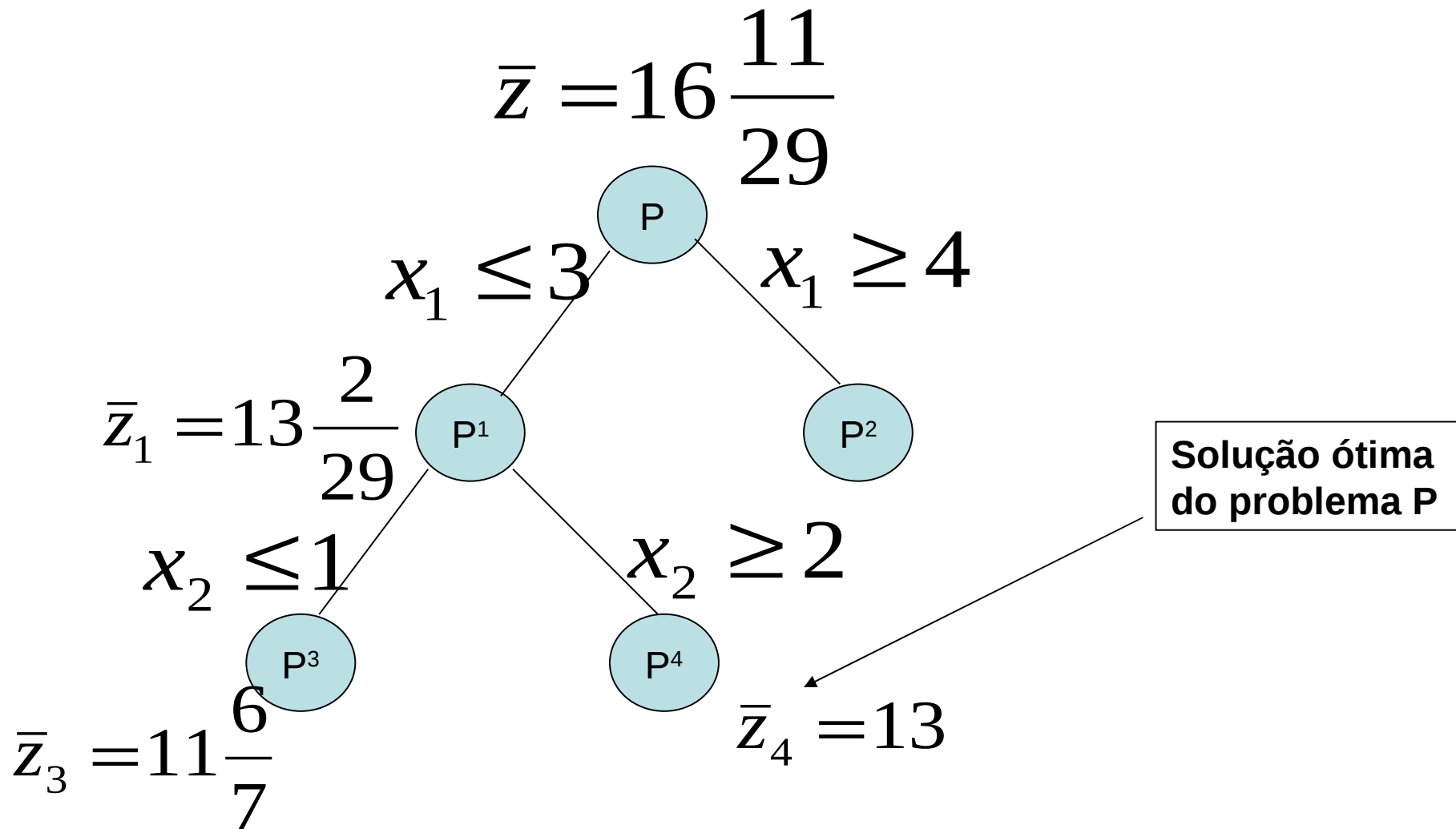
$$\bar{x} = (\bar{x}_1, \bar{x}_2) = \left(3\frac{24}{29}, 2\frac{22}{29}\right)$$

- com valor $\bar{z} = 16\frac{11}{29}$ que é um limitante superior para o valor ótimo z

Método *Branch-and-Bound*

- Considere a notação $\lfloor x \rfloor = \max\{y \in Z : y \leq x\}$, o maior inteiro contido em x , e o menor inteiro que contém x $\lceil x \rceil = \min\{y \in Z : y \geq x\}$,
- Considere que P é dividido em dois problemas P^1 e P^2
 - Adicione a restrição $x_1 \leq \lfloor \bar{x}_1 \rfloor = 3$ para gerar o problema P^1
 - Adicione a restrição $x_1 \geq \lceil \bar{x}_1 \rceil = 4$ para gerar o problema P^2
- Essa partição pode ser representada por meio de uma árvore em que cada nó corresponde a um problema
 - Essa árvore é denominada de árvore B&B

Método *Branch-and-Bound*



Método *Branch-and-Bound*

- Testes para Eliminação dos Nós na Árvore *Branch-and-Bound*
 - Um problema P^i correspondente a um nó na árvore B&B é eliminado ou descartado por testes que utilizam sua relaxação linear PL^i
 - Para formalizar, considere a seguinte notação:
 - $F(P^i)$: região factível do problema P^i
 - $F(PL^i)$: região factível do problema PL^i
 - z^i : valor ótimo do problema P^i (limitante inferior de z)
 - $\bar{z}^i$: valor ótimo do problema PL^i (limitante superior de z)
 - x^* : melhor solução encontrada até o momento (solução incumbente)
 - z^* : valor de x^*

Método *Branch-and-Bound*

- O problema P^i é eliminado se satisfazer um dos seguintes testes de eliminação:
 - $F(PL^i) = \emptyset$: P^i é eliminado por infactibilidade. O resultado decorre da relação:
 - $\bar{z}^i \leq z^* : P^i$ é eliminado por qualidade. Como $z^i \leq \bar{z}^i \leq z^*$, segue-se o resultado
 - Se a solução ótima de PL^i é inteira: P^i é eliminado por otimalidade. Como a solução ótima de PL^i é inteira, segue-se que $\bar{z}^i \leq z^i$. Além disso, $z^i \leq \bar{z}^i$, portanto, $z^i = \bar{z}^i$,

Método *Branch-and-Bound*

- Algoritmo *Branch-and-Bound*
 - O nó 0 da árvore B&B corresponde ao problema original de programação inteira P
 - Um nó não eliminado pelos testes de eliminação de nós e que ainda não foi dividido é chamado de nó ativo
 - Nós ativos são armazenados em uma lista L
 - Passo 0 (inicialização): Faça:

$$\bar{z} = \infty, z^* = -\infty, x^* = \emptyset, L = \{P\}$$

Método *Branch-and-Bound*

- Algoritmo *Branch-and-Bound*
 - Passo 1 (Seleção do nó). Selecione o nó ativo i , associado ao problema P^i , da lista de nós ativos. Se a lista estiver vazia, vá para o Passo 6.
 - Passo 2 (Teste de eliminação 1). Se a região factível de PL^i for vazia, vá para o Passo 1.
 - Passo 3 (Teste de eliminação 2). Se o valor de $\bar{z}^i$ da solução ótima de PL^i é tal que $\bar{z}^i \leq z^*$ vá para o Passo 1
 - Passo 4 (Teste de eliminação 3). Se a solução ótima $\bar{x}_i$ de PL^i é inteira com valor $\bar{z}^i$, e se $\bar{z}^i > z^*$, atualize x^* e z^*

Método *Branch-and-Bound*

- Algoritmo *Branch-and-Bound*
 - Passo 5 (Ramificação). Selecione uma variável da solução ótima $\bar{x}_i$ de PL^i com valor não inteiro e divida P^i em dois problemas. Adicione os dois problemas à lista L e vá para o Passo 1
 - Passo 6 (Fim). Se $Z^* = -\infty$, não existe solução factível; caso contrário, a solução incumbente x^* é uma solução ótima

Seleção de Nós na Árvore B&B

- Regra *a priori*
 - Busca em profundidade com backtracking
 - *last-in first-out*: último nó inserido na lista L é o primeiro a ser examinado
 - Se o nó corrente não é eliminado então o próximo a ser examinado é um de seus “filhos”
 - Vantagens:
 - Na prática, soluções factíveis são mais prováveis de serem encontradas em níveis profundos em relação ao nó raiz
 - O tamanho máximo da lista L não é grande
 - Desvantagem:
 - Árvore com muitos nós

Seleção de Nós na Árvore B&B

- Regra adaptativa
 - Selecionar nó com maior limitante superior
 - A idéia por traz dessa escolha é a expectativa otimista em relação à obtenção de uma nova solução factível com valor máximo em relação a solucoes factiveis geradas pelos outros nos
 - Vantagem:
 - Em geral, produz arvore B&B com menos nos do que a abordagem em profundidade
 - Desvantagem:
 - Armazenamento de mais nos ativos