O método *branch-and-bound* para o problema da mochila binária

Prof. Me. Alex Paranahyba de Abreu

Pesquisa Operacional para a Engenharia de Produção 1
04/07/2025

DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
ufscar

Objetivos da aula

- Revisar os conceitos de programação inteira;
- Explorar o problema da mochila binária;
- Fixar os conceitos por meio da resolução de exercícios.

Bibliografia

- ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. Pesquisa Operacional: Para cursos de engenharia. 2a ed. Rio de Janeiro: Elsevier, 2015.

Revisão: programação inteira

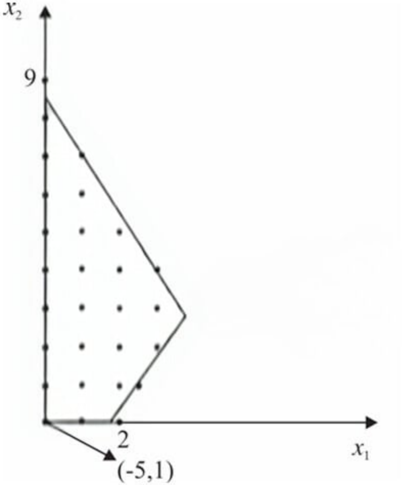
- Modelagem de problemas com programação linear inteira.

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.a} \quad & Ax = b \\
& x \in \mathbb{Z}^n_+
\end{aligned}
$$

Revisão: exemplo

- Exemplo de um problema de programação inteira.

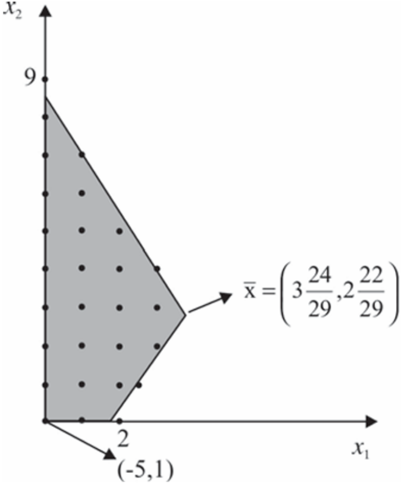
$$
\begin{aligned}
\max \quad & 5x_1 - 1x_2 \\
\text{s.a} \quad & 7x_1 - 5x_2 \leq 13 \\
& 3x_1 + 2x_2 \leq 17 \\
& x_1, x_2 \in \mathbb{Z}_+
\end{aligned}
$$



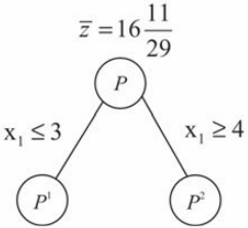
Revisão: relaxação linear

- Relaxação linear do problema de programação inteira.

$$
\begin{aligned}
P = \max \quad & 5x_1 - 1x_2 \\
\text{s.a} \quad & 7x_1 - 5x_2 \leq 13 \\
& 3x_1 + 2x_2 \leq 17 \\
& \boldsymbol{x_1, x_2} \in \mathbb{R}_+
\end{aligned}
$$



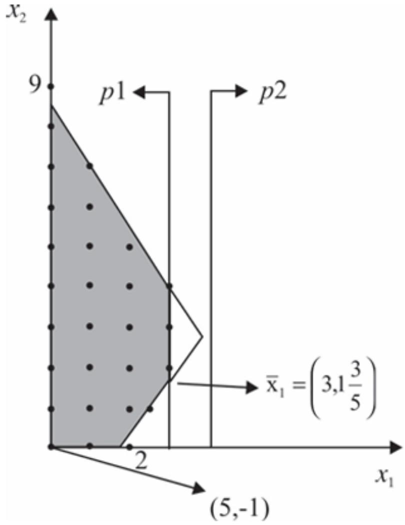
Revisão: ramificação



Revisão: solução dos problemas

$$P_1 = \max \quad 5x_1 - 1x_2$$
$$\text{s.a} \quad 7x_1 - 5x_2 \leq 13$$
$$3x_1 + 2x_2 \leq 17$$
$$x_1 \leq 3$$
$$x_1, x_2 \in \mathbb{R}_+$$

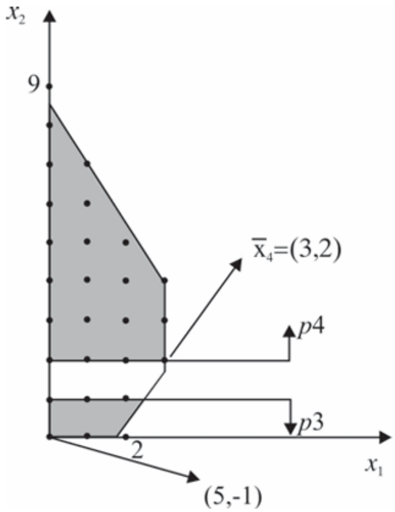
$$P_2 = \max \quad 5x_1 - 1x_2$$
$$\text{s.a} \quad 7x_1 - 5x_2 \leq 13$$
$$3x_1 + 2x_2 \leq 17$$
$$x_1 \geq 4$$
$$x_1, x_2 \in \mathbb{R}_+$$



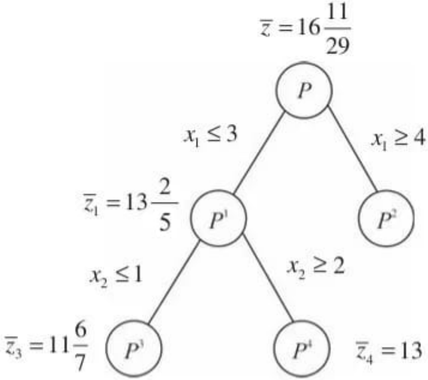
Revisão: solução dos problemas

$$
\begin{aligned}
P_3 = \max \quad & 5x_1 - 1x_2 \\
\text{s.a} \quad & 7x_1 - 5x_2 \leq 13 \\
& 3x_1 + 2x_2 \leq 17 \\
& x_1 \leq 3 \\
& x_2 \leq 1 \\
& x_1, x_2 \in \mathbb{R}_+
\end{aligned}
$$

$$
\begin{aligned}
P_4 = \max \quad & 5x_1 - 1x_2 \\
\text{s.a} \quad & 7x_1 - 5x_2 \leq 13 \\
& 3x_1 + 2x_2 \leq 17 \\
& x_1 \leq 3 \\
& x_2 \geq 2 \\
& x_1, x_2 \in \mathbb{R}_+
\end{aligned}
$$



Revisão: ramificação



Revisão: principais estratégias

- Estratégias de busca:
  em largura ou em profundidade;

- Estratégias de ramificação:
  "mais" fracionária ou melhor potencial;

- Motivos de "poda":
  infactibilidade, integralidade, qualidade.

O problema da mochila binária

Definição

O problema da mochila binária (0-1) envolve escolher itens únicos a serem colocados em uma mochila de forma a maximizar uma função de utilidade sem exceder a sua capacidade.

O problema da mochila binária

Definição

O problema da mochila binária (0-1) envolve escolher itens únicos a serem colocados em uma mochila de forma a maximizar uma função de utilidade sem exceder a sua capacidade.

Contexto de exemplo:

- Uma empresa quer selecionar projetos para serem realizados;
- Exitem n projetos candidatos;
- Cada projeto i possui um custo a_i e retorno esperado p_i;
- A empresa possui um orçamento b.

Modelagem do problema da mochila 0-1

- Identificar as **variáveis** do problema:
  x_i é igual a 1 se o projeto i é selecionado, ou 0 caso contrário.

Modelagem do problema da mochila 0-1

- Identificar as **variáveis** do problema:
  x_i é igual a 1 se o projeto i é selecionado, ou 0 caso contrário.

- Identificar o **objetivo** do problema:
  Maximizar o retorno esperado dos projeto selecionados
  ($\max f(x) = p_1x_1 + p_2x_2 + ... + p_nx_n$).

Modelagem do problema da mochila 0-1

- Identificar as **variáveis** do problema:
  x_i é igual a 1 se o projeto i é selecionado, ou 0 caso contrário.

- Identificar o **objetivo** do problema:
  Maximizar o retorno esperado dos projeto selecionados
  $(\max f(x) = p_1x_1 + p_2x_2 + ... + p_nx_n)$.

- Identificar as **restrições** do problema:
  O custo dos projetos selecionados devem atender o orçamento
  $(a_1x_1 + a_2x_2 + ... + a_nx_n \leq b)$.

Modelagem do problema da mochila 0-1

$$
\begin{aligned}
\max \quad & f(x) = \sum_{i=1}^{n} p_i x_i \\
\text{s.a} \quad & \sum_{i=1}^{n} a_i x_i \leq b, \\
& x_i \in \{0, 1\}, \qquad\qquad\qquad i \in \{1, 2, ..., n\}.
\end{aligned}
$$

A resolução desse problema pode ser feita por meio do método *branch-and-bound*.

O problema da mochila binária: relaxação linear

$$\begin{aligned}
\max \quad & f(x) = \sum_{i=1}^{n} p_i x_i \\
\text{s.a} \quad & \sum_{i=1}^{n} a_i x_i \leq b, \\
& 0 \leq x_i \leq 1, \qquad\qquad\qquad i \in \{1, 2, ..., n\}.
\end{aligned}$$

Ou seja, resolve-se a relaxação linear e:

- ramifique, se a solução é fracionária; ou
- não ramifique, se a solução é inteira, pior que a atual ou não existe.

O problema da mochila binária: relaxação linear

$$
\begin{aligned}
\max \quad & f(x) = \sum_{i=1}^{n} p_i x_i \\
\text{s.a} \quad & \sum_{i=1}^{n} a_i x_i \leq b, \\
& 0 \leq x_i \leq 1, \qquad\qquad\qquad i \in \{1, 2, ..., n\}.
\end{aligned}
$$

Ou seja, resolve-se a relaxação linear e:

- ramifique, se a solução é fracionária; ou
- não ramifique, se a solução é inteira, pior que a atual ou não existe.

A relaxação linear do problema da mochila binária pode ser resolvido por inspeção. [1]

[1] KELLERER, H.; PFERSCHY, U.; PISINGER, D. *Knapsack Problems*. 1a ed. Berlin: Springer, 2004.

Relaxação linear por inspeção

1. Calcule a "utilidade relativa" $u_i = p_i/a_i$ de cada item i;

Relaxação linear por inspeção

1. Calcule a "utilidade relativa" $u_i = p_i/a_i$ de cada item i;
2. Ordene os itens de forma decrescente da utilidade relativa ($u_{[1]} \geq u_{[2]} \geq \ldots \geq u_{[n]}$), onde $[i]$ é o item que está na posição i;

Relaxação linear por inspeção

1. Calcule a "utilidade relativa" $u_i = p_i/a_i$ de cada item i;
2. Ordene os itens de forma decrescente da utilidade relativa ($u_{[1]} \geq u_{[2]} \geq \ldots \geq u_{[n]}$), onde $[i]$ é o item que está na posição i;
3. Na ordem definida, selecione os itens (fazendo $x_i = 1$) até que atingir a capacidade da mochila (sem violá-la);

Relaxação linear por inspeção

1. Calcule a "utilidade relativa" $u_i = p_i/a_i$ de cada item i;
2. Ordene os itens de forma decrescente da utilidade relativa ($u_{[1]} \geq u_{[2]} \geq \ldots \geq u_{[n]}$), onde $[i]$ é o item que está na posição i;
3. Na ordem definida, selecione os itens (fazendo $x_i = 1$) até que atingir a capacidade da mochila (sem violá-la);
4. Se todos os itens foram escolhidos, solução ótima **inteira** encontrada.

Relaxação linear por inspeção

1. Calcule a "utilidade relativa" $u_i = p_i/a_i$ de cada item i;
2. Ordene os itens de forma decrescente da utilidade relativa ($u_{[1]} \geq u_{[2]} \geq \ldots \geq u_{[n]}$), onde $[i]$ é o item que está na posição i;
3. Na ordem definida, selecione os itens (fazendo $x_i = 1$) até que atingir a capacidade da mochila (sem violá-la);
4. Se todos os itens foram escolhidos, solução ótima **inteira** encontrada.
5. Senão, defina a variável do primeiro item que não coube como a fração necessária para completar a capacidade da mochila.

Relaxação linear por inspeção: exemplo

Exemplo. Uma companhia de extração de petróleo tem disponível 5 bombas de submersão e deseja usá-las para aumentar a quantidade extraída de petróleo. A energia para o funcionamento das bombas é fornecida por um único gerador, cuja potência máxima é 8 kVA. A potência consumida e a eficiência de extração de cada bomba são apresentadas na tabela a seguir. Deseja-se determinar quais bombas devem ser usadas, de modo a maximizar a eficiência total da extração.

Bomba	1	2	3	4	5
Eficiência	4	15	12	16	18
Consumo (kVA)	2	3	2	4	6

Relaxação linear por inspeção: exemplo

i	1	2	3	4	5
p_i	4	15	12	16	18
a_i	2	3	2	4	6
u_i					

Relaxação linear por inspeção: exemplo

i	1	2	3	4	5
p_i	4	15	12	16	18
a_i	2	3	2	4	6
u_i					

$$\begin{aligned} \max \quad & f(x) = 4x_1 + 15x_2 + 12x_3 + 16x_4 + 18x_5 \\ \text{s.a} \quad & 2x_1 + 3x_2 + 2x_3 + 4x_4 + 6x_5 \leq 8, \\ & x_1, x_2, x_3, x_4, x_5 \in \{0, 1\}. \end{aligned}$$

Relaxação linear por inspeção: exemplo

i	1	2	3	4	5
p_i	4	15	12	16	18
a_i	2	3	2	4	6
u_i					

$$
\begin{aligned}
\max \quad & f(x) = 4x_1 + 15x_2 + 12x_3 + 16x_4 + 18x_5 \\
\text{s.a} \quad & 2x_1 + 3x_2 + 2x_3 + 4x_4 + 6x_5 \leq 8, \\
& 0 \leq x_1, x_2, x_3, x_4, x_5 \leq 1.
\end{aligned}
$$

Relaxação linear por inspeção: exemplo

i	1	2	3	4	5
p_i	4	15	12	16	18
a_i	2	3	2	4	6
u_i	2	5	6	4	3

Relaxação linear por inspeção: exemplo

i	1	2	3	4	5
p_i	4	15	12	16	18
a_i	2	3	2	4	6
u_i	2	5	6	4	3

$$3 \to 2 \to 4 \to 5 \to 1$$

Relaxação linear por inspeção: exemplo

i	1	2	3	4	5
p_i	4	15	12	16	18
a_i	2	3	2	4	6
u_i	2	5	6	4	3

$$3 \to 2 \to 4 \to 5 \to 1$$

- Se fizermos $x_3 = 1$ e $x_2 = 1$, temos um consumo acumulado de $a_3 + a_2 = 5$;
- Com a folga de 3, a capacidade será violada se adicionarmos o item 4;
- Assim, apenas uma fração desse item será adicionada;
- Para completar a capacidade, faça $x_4 = (b - (a_3 + a_2))/a_4 = (8 - 5)/4 = 0.75$;
- A solução ótima da relaxação linear é $x = (0, 1, 1, 0.75, 0)$ com $f(x) = 39$.

Como incorporar isso no *branch-and-bound*

- A partir de uma solução fracionária $x = (0, 1, 1, 0.75, 0)$;
- Com a solução da relaxação linear, ramifica-se uma variável fracionária;
- Em um nó fixa-se $x_4 = 0$ e em outro $x_4 = 1$;
- Resolva cada nó por inspeção com/sem o item 4;

- Nesse exemplo vamos resolver usando busca em largura e ramificação na variável mais fracionária.

Árvore *branch-and-bound*: ramificação no nível 0

Nó 0 (raiz)
$x = (0, 1, 1, 0.75, 0)$
$f(x) = 39$

$x_4 = 0$ | $x_4 = 1$

Nó 1
$x = (*, *, *, 0, *)$
$f(x) = *$

Nó 2
$x = (*, *, *, 1, *)$
$f(x) = *$

Resolvendo o nó 1 ($x_4 = 0$)

i	1	2	3	4	5
p_i	4	15	12	16	18
a_i	2	3	2	4	6
u_i	2	5	6	4	3

$$3 \rightarrow 2 \rightarrow \cancel{4} \rightarrow 5 \rightarrow 1$$

- Com $x_4 = 0$, $x_3 = 1$ e $x_2 = 1$ temos um consumo acumulado de $a_3 + a_2 = 5$;
- Com a folga de 3, a capacidade será violada se adicionarmos o item 5;
- Assim, apenas uma fração desse item será adicionada;
- Para completar a capacidade, faça $x_5 = (b - (a_3 + a_2))/a_5 = (8 - 5)/6 = 0.5$;
- A solução ótima desse nó é $x = (0, 1, 1, 0, 0.5)$ com $f(x) = 36$.

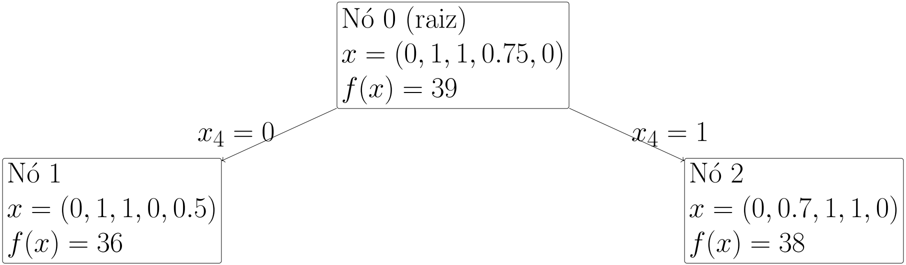
Resolvendo o nó 2 ($x_4 = 1$)

i	1	2	3	4	5
p_i	4	15	12	16	18
a_i	2	3	2	4	6
u_i	2	5	6	4	3

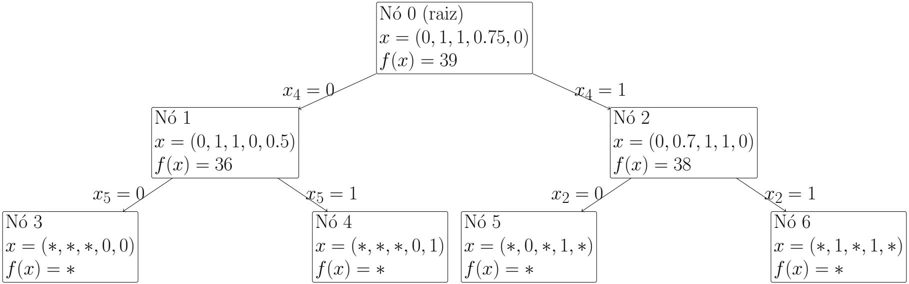
$$3 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 1$$

- Com $x_4 = 1$ e $x_3 = 1$ temos um consumo acumulado de $a_4 + a_3 = 6$;
- Com a folga de 2, a capacidade será violada se adicionarmos o item 2;
- Assim, apenas uma fração desse item será adicionada;
- Para completar a capacidade, faça $x_2 = (b - (a_4 + a_3))/a_2 = (8 - 6)/3 = 0.7$;
- A solução ótima desse nó é $x = (0, 0.7, 1, 1, 0)$ com $f(x) = 38$.

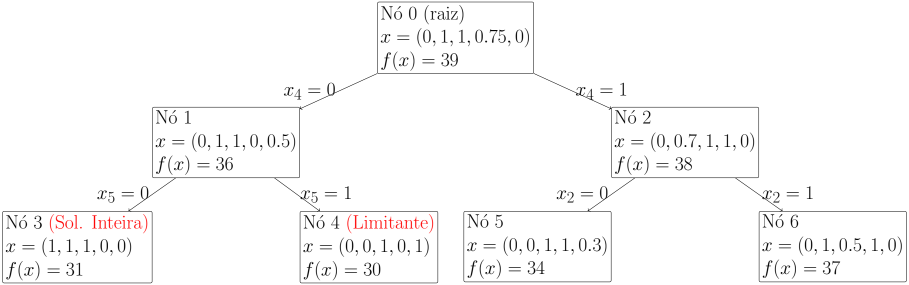
Árvore *branch-and-bound*: ramificação no nível 1



Árvore *branch-and-bound*: ramificação no nível 1

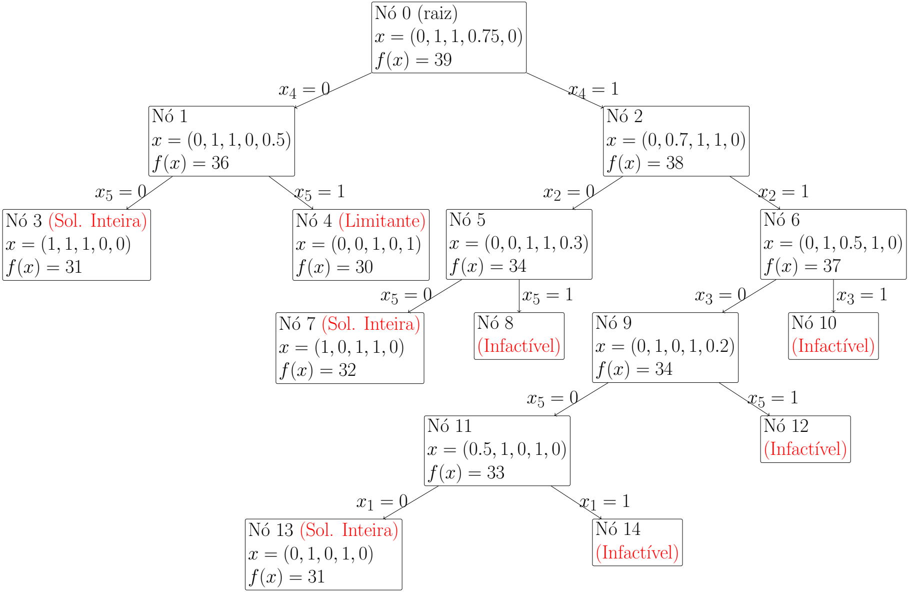


Árvore *branch-and-bound*: até o nível 2



- Atualize a melhor solução inteira encontrada e seu valor objetivo:
- $\bar{x} = (1, 1, 1, 0, 0)$ e $f(\bar{x}) = 31$.

Árvore *branch-and-bound* resultante



Exemplo resolvido

- Fim do método *branch-and-bound*: todos os nós foram explorados;
- Solução ótima $x^* = (1, 0, 1, 1, 0)$, com valor ótimo $f(x^*) = 32$;
- Utiliza-se as bombas 1, 3 e 4;
- Toda a capacidade do gerador é consumida;
- A eficiência máxima de extração é 32.

Conclusão da aula

- Revisamos a modelagem de problemas com programação inteira;
- Revisamos os conceitos do método *branch-and-bound*;
- Modelamos o problema da mochila binária;
- Exploramos uma forma de resolver a relaxação linear do problema;
- Utilizamos esse método para resolver o problema de programação inteira.

Materiais

- Slides e exercícios:
- `https://abreualexp.github.io/ufscar/tema4.html`

- Obrigado pela atenção!
- Dúvidas?