

Template LaTeX para Apresentações de TCC

Aluno ¹ Orientador ²

¹Curso / UFOP

²Departamento / UFOP

Junho de 2020



UFOP

Universidade Federal
de Ouro Preto



Seção 1

Texto em Tópicos



Texto em Tópicos

- ▶ Esse slide apresenta o texto organizado em tópicos e subtópicos.
- ▶ O restante do modelo apresenta os seguintes exemplos:
 - ▶ Figuras.
 - ▶ Tabelas.
 - ▶ Algoritmos.
 - ▶ Código Fonte.
 - ▶ Modelos Matemáticos.
 - ▶ Agradecimentos.
 - ▶ Referências.



Texto em Tópicos - overlays

- ▶ Esse slide apresenta o texto organizado em tópicos e subtópicos.
- ▶ O restante do modelo apresenta os seguintes exemplos:
 - ▶ Figuras.



Texto em Tópicos - overlays

- ▶ Esse slide apresenta o texto organizado em tópicos e subtópicos.
- ▶ O restante do modelo apresenta os seguintes exemplos:
 - ▶ Figuras.
 - ▶ Tabelas.



Texto em Tópicos - overlays

- ▶ Esse slide apresenta o texto organizado em tópicos e subtópicos.
- ▶ O restante do modelo apresenta os seguintes exemplos:
 - ▶ Figuras.
 - ▶ Tabelas.
 - ▶ Algoritmos.



Texto em Tópicos - overlays

- ▶ Esse slide apresenta o texto organizado em tópicos e subtópicos.
- ▶ O restante do modelo apresenta os seguintes exemplos:
 - ▶ Figuras.
 - ▶ Tabelas.
 - ▶ Algoritmos.
 - ▶ Código Fonte.
 - ▶ Modelos Matemáticos.



Texto em Tópicos - overlays

- ▶ Esse slide apresenta o texto organizado em tópicos e subtópicos.
- ▶ O restante do modelo apresenta os seguintes exemplos:
 - ▶ Figuras.
 - ▶ Tabelas.
 - ▶ Algoritmos.
 - ▶ Modelos Matemáticos.



Texto em Tópicos - overlays

- ▶ Esse slide apresenta o texto organizado em tópicos e subtópicos.
- ▶ O restante do modelo apresenta os seguintes exemplos:
 - ▶ Figuras.
 - ▶ Tabelas.
 - ▶ Algoritmos.
 - ▶ Modelos Matemáticos.



Texto em Tópicos - overlays

- ▶ Esse slide apresenta o texto organizado em tópicos e subtópicos.
- ▶ O restante do modelo apresenta os seguintes exemplos:
 - ▶ Figuras.
 - ▶ Tabelas.
 - ▶ Algoritmos.
- ▶ Agradecimentos.
- ▶ Referências.



Seção 2

Figuras



Figuras

- ▶ Figuras, tabelas e elementos não textuais são trabalhados exatamente como em documentos comuns LaTeX.
- ▶ Pode ser interessante incluir um espaçamento entre o texto e a imagem pelo comando `\vspace{3mm}`



Figura: É possível inserir legendas também.



Duas colunas: Figura e texto

Deste lado fica o texto, do outro
a(s) figura(s)

- ▶ item 1
- ▶ item 2

Observe ue neste slide, as figuras
estão utilizando *overlays*

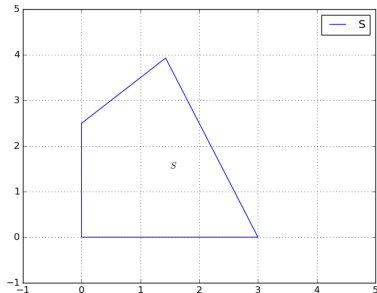


Duas colunas: Figura e texto

Deste lado fica o texto, do outro
a(s) figura(s)

- ▶ item 1
- ▶ item 2

Observe ue neste slide, as figuras
estão utilizando *overlays*

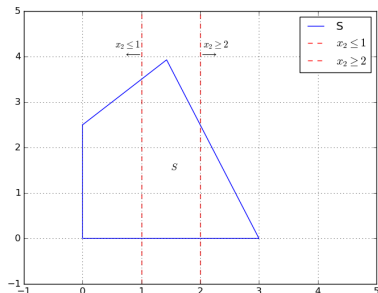


Duas colunas: Figura e texto

Deste lado fica o texto, do outro
a(s) figura(s)

- item 1
- item 2

Observe ue neste slide, as figuras
estão utilizando *overlays*

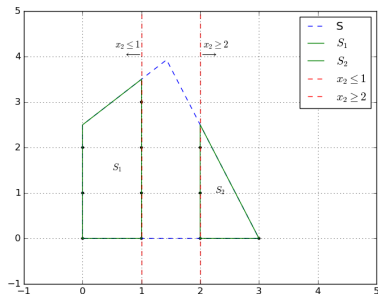


Duas colunas: Figura e texto

Deste lado fica o texto, do outro
a(s) figura(s)

- item 1
- item 2

Observe ue neste slide, as figuras
estão utilizando *overlays*



Seção 3

Tabelas



Tabelas

- ▶ Uma excelente ferramenta para a criação de tabelas é o <https://www.tablesgenerator.com/>.
- ▶ Um exemplo de tabela é dado abaixo.

Tabela: É possível inserir legendas também.

Instância	Algoritmo A	Algoritmo B	Algoritmo C
<i>Arquivo 1</i>	1.00±0.7	1.90±0.5	0.90±0.3
<i>Arquivo 2</i>	6.80±0.5	8.60±1.4	5.90±0.2
<i>Arquivo 3</i>	7.90±1.2	13.25±3.1	8.50±1.7
<i>Arquivo 4</i>	8.00±0.0	12.70±0.2	6.10±1.0

Seção 4

Algoritmos



Algoritmos

- ▶ Siga as convenções para pseudocódigos conforme <https://link.springer.com/content/pdf/bbm%3A978-1-4471-5173-9%2F1.pdf>
- ▶ Para trocar o idioma das palavra-chave altere o idioma de *portuguese* para *english* na linha:
`\usepackage[portuguese, ...]{algorithm2e}`



Algoritmos

- Pseudocódigo do algoritmo de ordenação Bubble-Sort:

Algoritmo 1: BUBBLE-SORT.

Entrada: (i) Um vetor V .

Saída: (i) O vetor V ordenado crescentemente.

```
1  para  $i = 0$  até  $|V| - 1$  faça
2      trocou  $\leftarrow$  false;
3      para  $j = 0$  até  $|V| - 1$  faça
4          se  $V[j + 1] > V[j]$  então
5              aux  $\leftarrow V[j]$ ;           ► Troca as posições  $i$  e  $i + 1$  de  $V$ .
6               $V[j] \leftarrow V[j + 1]$ ;
7               $V[j + 1] \leftarrow aux$ ;
8          trocou  $\leftarrow$  true;
9      se trocou então
10         break;                          ►  $V$  já está ordenado.
11 retorna  $V$ ;
```

Seção 5

Código-Fonte



Código-Fonte

- ▶ Segue um exemplo de implementação do algoritmo Bubble-Sort na linguagem Python

```
def bubblesort(V): #Complexidade O(N^2)
    for i in range(len(V) - 1):
        trocou = False
        for j in range(len(V) - 1):
            if V[j + 1] < V[j]:
                aux = V[j]
                V[j] = V[j + 1]
                V[j + 1] = aux
                trocou = True
        if not trocou:
            break
    return V
```

Seção 6

Modelos Matemáticos



Modelos Matemáticos

6.1 Modelo descrito no texto



Modelos Matemáticos

PROBLEMA DA MOCHILA

► Conjuntos e parâmetros

\mathcal{I} Itens

w_i Peso do item i

b_i Benefício do item i

C Capacidade da mochila

► Variáveis

$$x_i = \begin{cases} 1 & \text{se o item } i \in \mathcal{I} \text{ está na mochila.} \\ 0 & \text{caso contrário.} \end{cases}$$



Modelos Matemáticos

► Restrições

1. Capacidade da mochila:

$$\sum_{i \in \mathcal{I}} w_i \times x_i \leq C \quad (1)$$

► Função objetivo

$$\max \sum_{i \in \mathcal{I}} b_i \times x_i \quad (2)$$



Modelos Matemáticos

6.2 usando o Optprog



Localização de Instalações

Formulação:

$$\min \sum_{s \in S} \left\{ C_s^f Y_s + \sum_{c \in C} V_c (G_{sc} X_{sc}) \right\} + \sum_{c \in C} V_c G_c X_{ic} \quad (3)$$

$$\sum_{s \in S} X_{sc} + X_{ic} = 1, \quad \forall c \in C \quad (4)$$

$$\sum_{c \in C} X_{sc} \leq Y_s |C|, \quad \forall s \in S, \quad (5)$$

$$X_{sc} \in \{0, 1\}, \quad \forall s \in S, \forall c \in C \quad (6)$$

$$Y_s \in \{0, 1\}, \quad \forall s \in S \quad (7)$$

Seção 7

Usando o TiKz



Usando o TiKz

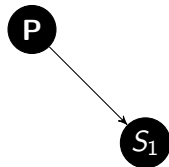
7.1 Com Overlay



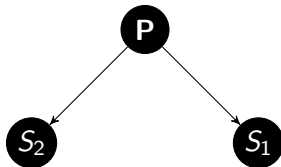
Árvore de BB - overlays



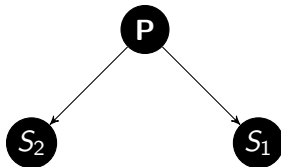
Árvore de BB - overlays



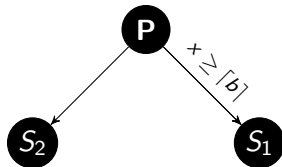
Árvore de BB - overlays



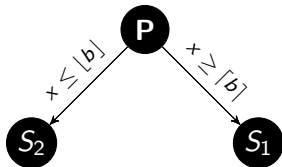
Árvore de BB - overlays



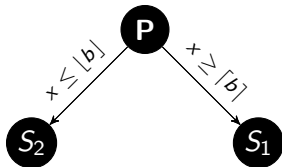
Árvore de BB - overlays



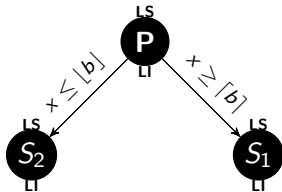
Árvore de BB - overlays



Árvore de BB - overlays



Árvore de BB - overlays



Usando o TiKz

7.2 Estático



Problema do fluxo de custo mínimo

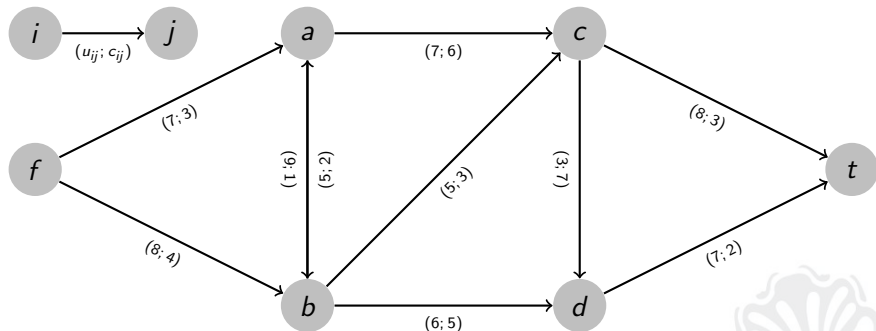


Figura: Exemplo de rede

Seção 8

Referências



Referências


- ▶ Sempre que se utilizar do trabalho de outrem é necessário incluir uma citação, que pode fazer parte do texto ou não.


Textual Fonseca et al. (2017) apresenta uma nova formulação para o problema.

Não-textual O problema de programação de horários é classificado como NP-Difícil (EVEN; ITAI; SHAMIR, 1975).



Referências I

 EVEN, S.; ITAI, A.; SHAMIR, A. On the complexity of time table and multi-commodity flow problems. In: *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*. [S.l.: s.n.], 1975. p. 184–193.

 FONSECA, G. H. et al. Integer programming techniques for educational timetabling. *European Journal of Operational Research*, v. 262, n. 1, p. 28 – 39, 2017.

