

# Caminho Mínimo: Uma Implementação Genérica OR-TOOLS (Python) Parte 03

Claudio Cesar de Sá<sup>1</sup>

Pesquisador Independente

## Roteiro

1. ~~Complexidade de Problemas~~
2. ~~O que é a Programação Linear Inteira (PLI)~~
3. ~~Um problema: Caminho Mínimo~~
4. ~~Modelagem com uma técnica de PLI~~ ⇒ **vídeo- 01**
5. ~~Implementação e código com OR-TOOLS (Python)~~ ⇒ **vídeo-02**
6. Generalizando o Caminho Mínimo (avançado) ⇒ **este vídeo**
7. Todo material:

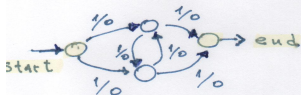
`https://github.com/claudiosa/CCS/.../presentations-seminars/cam\_min\_PL`  
e

`https://github.com/claudiosa/CCS/.../python/or-tools`

# Relembrando ....

Cálculo Mínimo - Livro do TAHA

Basicamente:



=>

$x_{ij}$	1	2	3	4
1	1/0	...		1/0
2		...		...
3		...	...	...
4	1/0	...		1/0

Objetivo:

$x_{ij}$				
1	0	0	1	0
2	1	0	0	0
3	0	1	0	0
4	0	0	0	0

→ se for  
nó FINAL

Estratégia:

Análise do Fluxo (F)  
por nó na rede / grafo

# Restrições – Intercalando com o código

Análise de Fluxo ( $\Phi$ ): se um nó <sup>2</sup>  
é selecionado pelo menor caminho

então:

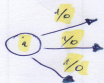
$$\Phi_{input}^i = \Phi_{output}^i = 1$$

caso contrário

$$\Phi_{inp} = \Phi_{out} = 0$$

## RESTRIÇÕES:

Fluxo de Saída:  $\Phi_{out}^i$

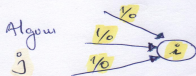


Algum  $j$   
é/o destino

$$\Rightarrow \Phi_{out}^i = 1$$

EXCETO SE  $i = \text{END}$

Fluxo de Entrada:  $\Phi_{inp}^i$



Algum  
 $j$   
PARA ...

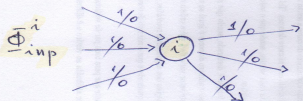
$$\Rightarrow \Phi_{inp}^i = 1$$

exceto se  $i = \text{START}$

# Análise de Fluxo

Fluxo nos Demais nós:

3



$\Phi_{out}^i$

$\Rightarrow$

$$\Phi_{inp}^i = \Phi_{out}^i$$

Seu

$i \neq \text{START}$   
 $i \neq \text{END}$

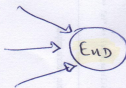
Fluxo no nó START/início:



$\Rightarrow$

$$\Phi_{inp}^{\text{start}} - \Phi_{out}^{\text{start}} = -1$$

Fluxo no nó FINAL/END:




$\Rightarrow$


$$\Phi_{inp}^{\text{end}} - \Phi_{out}^{\text{end}} = 1$$

# Função Custo

ALGUMAS OUTRAS RESTRIÇÕES:

4

a) Se   $\Rightarrow x_{ii} = 0 \quad \forall i$   
"Ciclos não levam a um bom caminho!"

b) Se   $\Rightarrow x_{ij} = 0$   
 $M = \text{HIGH}$   
M é HIGH e NÃO CONECTADO

Finalmente:

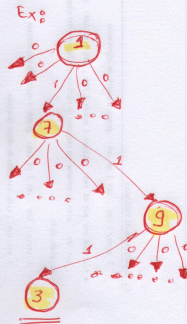
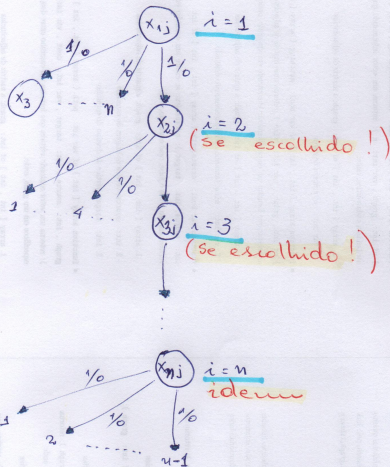
minimizar  $\left( \sum x_{ij} d_{ij} \right)$

VARIÁVEIS:  $x_{ij}$ ,  $\Phi_{in}^i$ ,  $\Phi_{out}^i$ , f-custo.

# Árvore de Busca $\approx$

## Árvore de Busca APROXIMADA:

5



R: 1  $\rightarrow$  7  $\rightarrow$  9  $\rightarrow$  3

mais real  $\uparrow$

# Quanto a OR-TOOLS:

- ▶ OR-TOOLS  
(<https://developers.google.com/optimization>)
- ▶ Ferramenta livre mantida pela Google: uso em **otimização combinatorial**
- ▶ Suporta vários *solvers* (*núcleo resolvidor*)
- ▶ Dois pacotes mais importantes: Programação Linear (Inteira e Mista) e Programação por Restrições
- ▶ Linguagens de *front-end*: C++, C#, Java e Python
- ▶ Vamos usar o CP-SAT com Python
- ▶ CP: *Constraint Programming*
- ▶ ... ao código



## Contato e Comentários:

- ▶ <https://claudiocesar.wordpress.com/>
- ▶ <https://github.com/claudiosa>
- ▶ Email: [claudio.sa@udesc.br](mailto:claudio.sa@udesc.br)
- ▶ Email: [ccs1664@gmail.com](mailto:ccs1664@gmail.com)
- ▶ *Thank you so much!*