

# Caminho Mínimo: Uma Implementação com a OR-TOOLS (Python)

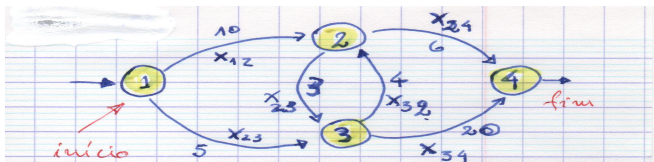
Claudio Cesar de Sá<sup>1</sup>

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 anterior**
5. Implementação e código com OR-TOOLS (Python) ⇒ **este vídeo**
6. Generalizando o Caminho Mínimo (avançado) ⇒ **próximo vídeo**
7. Todo material: [https://github.com/claudiosa/CCS/tree/master/presentations-seminars/cam\\_min\\_PL](https://github.com/claudiosa/CCS/tree/master/presentations-seminars/cam_min_PL)

## Retomando ....



Equações de Fluxo ( $\Phi_{\text{in}} = \Phi_{\text{out}}$ )

# 1 :  $1 = x_{12} + x_{13}$

# 2 :  $x_{12} + x_{32} = x_{24} + x_{23}$

# 3 :  $x_{23} + x_{13} = x_{32} + x_{34}$

# 4 :  $x_{34} + x_{24} = 1$

com  $0 \leq x_{ij} \leq 1$   $i, j = 1, 2, \dots$   
pto do menor caminho:

minimizar  $\left( 10 \cdot x_{12} + 5x_{13} + \right.$   
 $\left. 3 \cdot x_{23} + 6x_{24} + \right.$   
 $\left. 4 \cdot x_{32} + 20 \cdot x_{34} \right)$

Vamos ao código!

# Implementação:

- ▶ 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!*