Caminho Mínimo: Uma Modelagem com Programação Linear Parte – 01

Claudio Cesar de Sá[1]

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 ⇒ **este vídeo**
5. Implementação e código com OR-TOOLS (Python) ⇒ **próximo vídeo**
6. Generalizando o Caminho Mínimo (avançado) ⇒ **um outro vídeo**
7. Este material: `https://github.com/claudiosa/CCS/tree/master/presentations-seminars/cam_min_PL`

Classes de Problemas

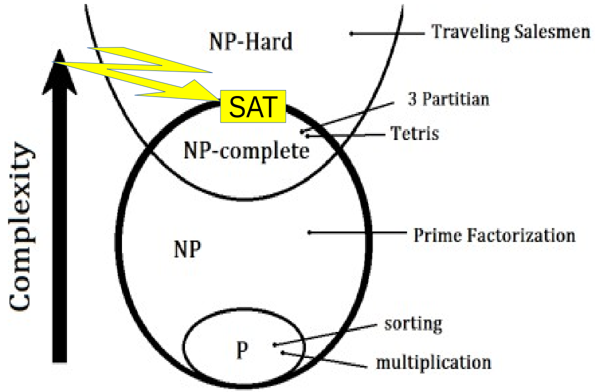


Figura: Alguns problemas e suas complexidades

Programação Linear

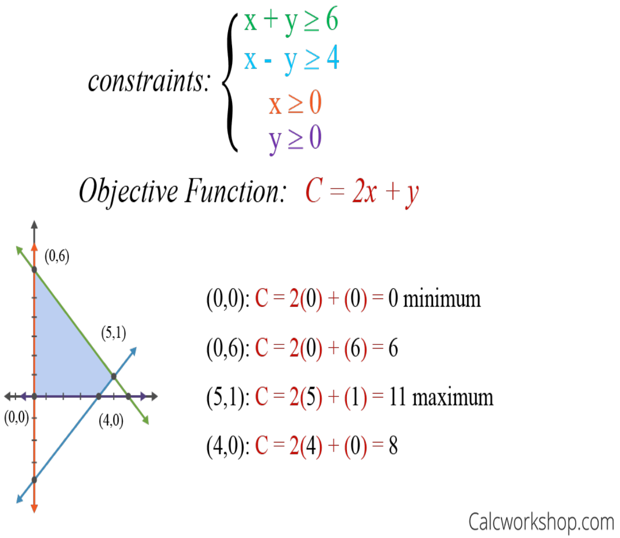
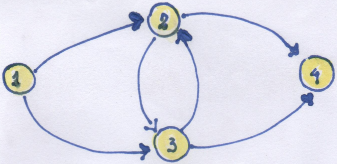


Figura: Uma técnica de resolver problemas: equações lineares

Elementos de um Grafo

Grafo:



Nós = {1, 2, 3, 4}

Arestas ou Arcos = {(1 2), (1 3) ... (2 4), (3 4)}

Logo: (a b) : (origem, destino)

$(a\,b) \equiv a \rightarrow b$ $\therefore$ $(a\,b) \neq (b\,a)$

Problema do Caminho Mínimo

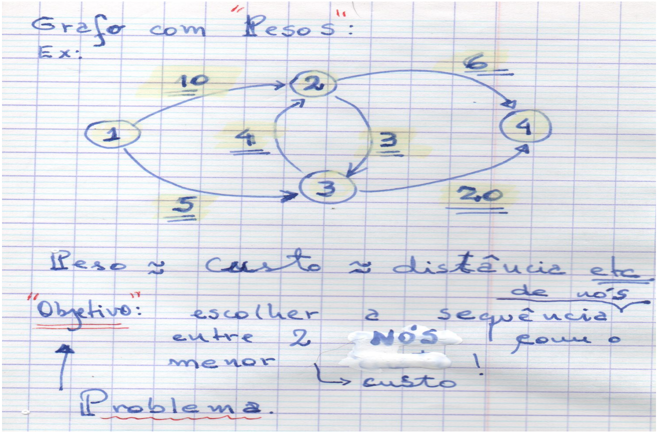


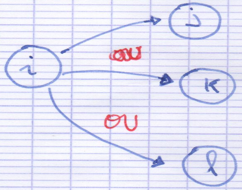
Figura: Um problema da classe P (mas muito útil) com uma solução via PLI

Referência:
https://en.wikipedia.org/wiki/Shortest_path_problem

Variável de Decisão e Uso

VARIÁVEL DE DECISÃO : ("decide algo")

Ex: $X = \{0, 1\}$; $\{F, V\}$, ...
(binária esta!)



Qual caminho a tomar?

$x_{ij} = 1/0$

$x_{ik} = 1/0$

$x_{il} = 1/0$

Só pode 1 arco ser escolhido!

Logo: $\boxed{x_{ij} + x_{ik} + x_{il} = 1}$

Notação rigorosa: $X_{\text{origem, destino}}$

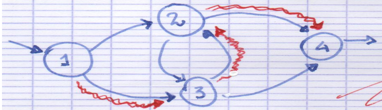
$i \rightarrow j \neq j \rightarrow i$

Matriz de Decisão e Uso

X_{ij}	1	2	3	4
1	0/1	0/1	0/1	0/1
2	⋮	⋮	⋮	⋮
3				
4	0/1	0/1	0/1	0/1

Ex1:

Matriz ou Variável de Decisão



$\equiv$

Destino

j:

x_{ij}	1	2	3	4
1	0	0	1	0
2	0	0	0	1
i: 3	0	1	0	0
4	0	0	0	0

Origem

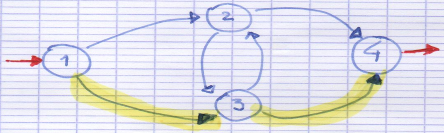
Escolhidos:
(Sequência)

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

Logo $x_{13} = x_{32} = x_{24} = 1$

Definindo um Caminho

Logo um CAMINHO é uma SEQUÊNCIA DE NÓS (escolhidos):



Ex:

$1:4 \Rightarrow 1 \rightarrow 2 \rightarrow 4$

$1 \rightarrow 3 \rightarrow 4$

$Ex_1 \Rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 4$ (exemplo anterior)

...

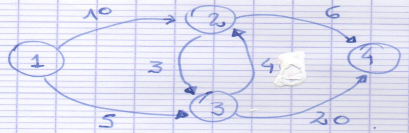
Se repetir nó $\Rightarrow$ Ineficiente!

Ex_2:

$$1 \xrightarrow{x_{13}} 3 \xrightarrow{x_{34}} 4$$

$\therefore x_{13} = x_{34} = 1$ e os demais iguais a zero

O Caminho e a Matriz de Peso



Matriz com Pesos:

juntando: caminho + matriz decisão x_{ij}

para um caminho 1 → 4

A solução seria: 1 → 3, 3 → 2, 2 → 4

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

Como selecionar x_{13}, x_{32} e x_{24}?

Mas como encontrar esta combinação ideal?

Menor caminho = menor custo

Formalizando os Elementos

d_{ij}	j: 1	2	3	4
i: 1	∅	10	5	M
2	M	∅	3	6
3	M	4	∅	20
4	M	M	M	∅

MATRIZ - peso dos arcos

(Peso ou Valor...)

Matriz ponderada

i: origem

j: destino

d : distância, custo, capacidade, ... etc

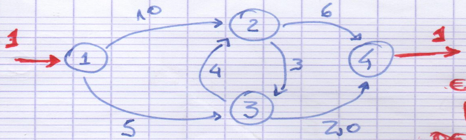
M: um valor negativo ou "grande" (sem conexão)

Ex: M = 9999

BASICAMENTE UM PRODUTO ESCALAR

$\approx$ $Pesos_{ij}$ × X_{ij} e minimizar estas escolhas!

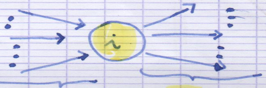
Estratégia deste Problema: **Fluxo** (Φ)



ESTRATÉGIA:
FLUXO
~~DE~~ INP = OUT
EM CADA NÓ!

Um caminho passa única vez em qualquer nó!

Logo:



$\Phi_{inp} = \Phi_{out}$ para $\forall$ nó

Assim;

Equações lineares de escolher ou não um caminho e seu peso!

Φ: "FLUXO"

Equações Finais

$$\Sigma^{i}_{in}\left(\text{fluxo}\right)_{\#in} = \Sigma^{i}_{out}\left(\text{fluxo}\right)_{\#out}$$

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

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

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

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

PARA ESCOLHA DO CAMINHO MÍNIMO

é minimizar $= \sum_{(ij)\in C} d_{ij}\, x_{ij}$

i.é:

Para o caso acima é minimizar estas arestas ponderadas:

$$f\_min = 5x_{13} + 10x_{12} + 3x_{23} + 6x_{24} + 4x_{32} + 20x_{34}$$

6 arestas

TODAS!

Próximos Passos:

- Implementar na OR-TOOLS com Python (próximo vídeo)
- Ferramenta livre mantida pela Google
- Suporta várias linguagens de *front-end*: C++, C#, Java e Python
- Vários *solvers*
- Vamos usar o *solver*: CP-SAT
- CP: *Constraint Programming*
- *Obrigado!*

Contato e Comentários:

- https://claudiocesar.wordpress.com/
- https://github.com/claudiosa
- Email: claudio.sa@udesc.br
- Email: ccs1664@gmail.comr
- *Thank you so much*!