Lógica Computacional: 21/22

Trabalho 1

Grupo 7

- David José de Sousa Machado (A91665)
- Ivo Miguel Gomes Lima (A90214)

Inicialização

Para a resolução destes exercícios usamos a biblioteca OR-Tools que criou uma interface para o SCIP. Esta biblioteca foi instalada com o commando `pip install ortools`.

```
!pip install ortools
```

```
Requirement already satisfied: ortools in /usr/local/lib/python3.7/dist-packages (9.1.9490)
Requirement already satisfied: absl-py>=0.13 in /usr/local/lib/python3.7/dist-packages (from ortools) (0.15.0)
Requirement already satisfied: protobuf>=3.18.0 in /usr/local/lib/python3.7/dist-packages (from ortools) (3.19.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from absl-py>=0.13->ortools) (1.15.0)
```

```
import networkx as nx
from ortools.linear_solver import pywraplp
from tabulate import tabulate
import random
```

Problema 1: Horário de uma *StartUp*

Foi pedida a criação de um horário semanal para uma *Startup*, seguindo as seguintes condições:

1. Cada reunião ocupa uma sala (enumeradas $1...S$) durante um *"slot"* $(\text{tempo}, \text{dia})$. Assume-se os dias enumerados $1..D$ e, em cada dia, os tempos enumerados $1..T$.

2. Cada reunião tem associado um projeto (enumerados $1..P$) e um conjunto de participantes. Os diferentes colaboradores são enumerados $1..C$.
3. Cada projeto tem associado um conjunto de colaboradores, dos quais um é o líder. Cada projeto realiza um dado número de reuniões semanais. São *"inputs"* do problema o conjunto de colaboradores de cada projeto, o seu líder e o número de reuniões semanais.
4. O líder do projeto participa em todas as reuniões do seu projeto; os restantes colaboradores podem ou não participar consoante a sua disponibilidade, num mínimo (*"quorum"*) de 50% do total de colaboradores do projeto. A disponibilidade de cada participante, incluindo o lider, é um conjunto de *"slots"* (*"inputs"* do problema).

Análise do problema

Para criarmos um horário coerente e compatível com as disponibilidades de cada um dos intervenientes, foi necessário estabelecer algumas restrições.

Condições inerentes

As condições inerentes são relativas à verificação da coerência do mesmo. Estas condições são:

- Deve existir um número de reuniões semanais por projeto R_p, dado no input do problema:

$$\forall_{p<P} \qquad \sum_{s<S,\ d<D,\ h<H,\ c_{Lider}} x_{p,s,d,h,c} = R_p$$

- Para haver reunião de um projeto numa certa sala, dia e hora, o líder tem de estar presente:

$$\forall_{p<P} \qquad \sum_{s<S,\ d<D,\ h<H,\ c_{Lider}} x_{p,s,d,h,c} = 1$$

> **Nota:** As condições acima enunciadas acabam por culminar numa única pois o Líder tem de ir a todas as R reuniões o que implica a existência das mesmas.

- Não pode haver reunião de um projeto numa certa sala, dia e hora, se os colaboradores estiverem indisponíveis:

$$\forall_{s<S} \cdot \forall_{d<D} \cdot \forall_{h<H} \cdot \forall_{p<P} \qquad \sum_{c \in Proj\ \wedge\ c \in\ Indisponível} x_{p,s,d,h,c} = 0$$

- Cada colaborador, numa dada sala, dia e hora, não pode participar num projeto que não é o seu:

$$\forall_{s<S} \cdot \forall_{d<D} \cdot \forall_{h<H} \cdot \forall_{p<P} \cdot \forall_{c \notin Proj} \quad \sum x_{p,s,d,h,c} = 0$$

Limitações (que impõem limites máximos à alocação)

- Cada sala, num dado dia e hora, apenas pode acolher um projeto:

$$\forall_{s<S} \cdot \forall_{d<D} \cdot \forall_{h<H} \quad \sum_{p<P,\, c \in Proj} x_{p,s,d,h,c} \leq 1$$

- Cada projeto só pode ter no máximo uma reunião por dia e hora:

$$\forall_{p<P} \cdot \forall_{d<D} \cdot \forall_{h<H} \quad \sum_{s<S,\, c \in Proj} x_{p,s,d,h,c} \leq 1$$

- Cada colaborador, num dado dia e hora, só pode participar na sala do seu projeto:

$$\forall_{s<S} \cdot \forall_{d<D} \cdot \forall_{h<H} \cdot \forall_{p<P} \cdot \forall_{c \in Proj} \quad \sum_{s<S} x_{p,s,d,h,c} \leq 1$$

> **Nota:** Novamente as duas condições acima chegam a uma única pois ao garantirmos que cada colaborador apenas pode participar na reunião do seu projeto, garantimos que ela existe.

Obrigações (que impõem limites mínimos à alocação)

- Para haver reunião de um projeto numa certa sala, dia e hora, o líder e pelo menos 50% dos colaboradores devem estar disponíveis assim como a sala:

$$\forall_{s<S} \cdot \forall_{d<D} \cdot \forall_{h<H} \cdot \forall_{p<P} \quad \sum_{c \in \, (Proj \wedge Líder)} x_{p,s,d,h,c} \geq \frac{c}{2}$$

Implementação:

Gerador de Testes Aleatório

```
Slots = [(d, h) for d in range(5) for h in range(8)]
Colabs = set(range(30))

Colaboradores = [random.sample(Slots, 20) for _ in range(30)]
Projectos = []
for _ in range(5):
  team = random.sample(Colabs, 6)
  Projectos.append((random.randint(1, 5), team))
  Colabs = Colabs - set(team)

for c in range(30):
  Colaboradores[c].sort()
  print(c, Colaboradores[c])

for num_r, workers in Projectos:
  print(num_r, workers)

     0 [(0, 1), (0, 2), (0, 4), (0, 5), (1, 0), (1, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 0), (2, 1), (2, 2), (2, 5), (2, 7), (3, 0), (3, 6)
     1 [(0, 0), (0, 5), (1, 0), (1, 1), (1, 3), (1, 6), (1, 7), (2, 1), (2, 3), (2, 4), (2, 7), (3, 0), (3, 1), (3, 2), (3, 3), (3, 7), (4, 0)
     2 [(0, 0), (0, 1), (0, 3), (0, 5), (1, 1), (1, 6), (1, 7), (2, 1), (2, 2), (2, 6), (2, 7), (3, 1), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7)
     3 [(0, 0), (0, 1), (0, 4), (0, 6), (0, 7), (1, 2), (1, 3), (1, 4), (1, 7), (2, 1), (2, 2), (2, 4), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5)
     4 [(0, 0), (0, 1), (0, 3), (0, 4), (0, 6), (0, 7), (1, 0), (1, 5), (1, 6), (2, 0), (2, 1), (2, 2), (2, 4), (2, 6), (2, 7), (3, 0), (3, 1)
     5 [(0, 0), (0, 1), (0, 2), (0, 6), (0, 7), (1, 1), (1, 2), (1, 6), (1, 7), (2, 0), (2, 1), (2, 2), (2, 7), (3, 0), (3, 1), (3, 2), (3, 7)
     6 [(0, 0), (0, 2), (0, 3), (0, 5), (0, 6), (0, 7), (1, 0), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 5), (3, 0), (3, 4), (3, 7)
     7 [(0, 0), (0, 1), (0, 5), (0, 6), (0, 7), (1, 2), (1, 4), (1, 5), (1, 6), (2, 2), (2, 5), (2, 6), (2, 7), (3, 0), (3, 1), (3, 4), (3, 5)
     8 [(0, 0), (0, 3), (0, 6), (0, 7), (1, 0), (1, 5), (1, 6), (1, 7), (2, 0), (2, 2), (2, 4), (3, 0), (3, 3), (3, 4), (3, 6), (3, 7), (4, 0)
     9 [(0, 0), (0, 1), (0, 2), (0, 4), (0, 6), (0, 7), (1, 2), (1, 3), (1, 4), (1, 5), (1, 7), (2, 2), (2, 5), (3, 2), (3, 4), (3, 7), (4, 1)
     10 [(0, 3), (0, 4), (0, 6), (0, 7), (1, 0), (1, 1), (1, 2), (1, 3), (1, 7), (2, 2), (2, 6), (2, 7), (3, 0), (3, 1), (3, 2), (3, 6), (3, 7
     11 [(0, 0), (0, 4), (0, 6), (0, 7), (1, 0), (1, 2), (1, 3), (1, 4), (1, 7), (2, 0), (2, 1), (2, 2), (2, 6), (2, 7), (3, 2), (3, 6), (4, 0
     12 [(0, 2), (0, 3), (0, 6), (1, 0), (1, 1), (1, 4), (1, 5), (1, 6), (2, 0), (2, 4), (2, 5), (2, 6), (2, 7), (3, 0), (3, 3), (3, 4), (3, 5
     13 [(0, 1), (0, 2), (0, 5), (0, 6), (1, 0), (1, 1), (1, 3), (1, 4), (1, 5), (1, 7), (2, 2), (2, 6), (3, 0), (3, 2), (3, 4), (3, 5), (3, 6
     14 [(0, 2), (0, 3), (0, 4), (1, 0), (1, 2), (1, 5), (1, 6), (1, 7), (2, 1), (2, 2), (3, 0), (3, 1), (3, 2), (4, 0), (4, 1), (4, 2), (4, 3
     15 [(0, 1), (0, 4), (0, 6), (1, 0), (1, 2), (1, 3), (1, 4), (1, 6), (2, 7), (3, 0), (3, 1), (3, 3), (3, 4), (3, 7), (4, 1), (4, 2), (4, 3
     16 [(0, 0), (0, 1), (0, 5), (1, 4), (1, 5), (2, 1), (2, 3), (2, 4), (2, 6), (2, 7), (3, 0), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 7
     17 [(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 6), (0, 7), (1, 1), (1, 2), (1, 3), (1, 4), (1, 6), (2, 1), (2, 5), (2, 6), (3, 2), (3, 4
     18 [(0, 1), (0, 3), (0, 5), (0, 7), (1, 0), (1, 4), (1, 5), (1, 6), (1, 7), (2, 0), (2, 1), (2, 4), (2, 5), (3, 0), (3, 2), (3, 4), (3, 6
     19 [(0, 0), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (1, 1), (1, 3), (1, 4), (1, 7), (2, 3), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (4, 0
     20 [(0, 0), (0, 1), (0, 4), (0, 5), (1, 0), (1, 1), (1, 2), (1, 4), (1, 7), (2, 3), (2, 5), (2, 7), (3, 2), (3, 4), (3, 6), (3, 7), (4, 0
     21 [(0, 0), (0, 2), (0, 5), (0, 6), (1, 3), (1, 6), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 7), (3, 0), (3, 1), (3, 6), (3, 7), (4, 1
     22 [(0, 2), (0, 3), (0, 4), (0, 5), (1, 2), (1, 3), (1, 6), (1, 7), (2, 1), (2, 3), (2, 6), (2, 7), (3, 0), (3, 1), (3, 2), (3, 6), (4, 1
     23 [(0, 2), (0, 3), (0, 4), (0, 5), (0, 7), (1, 1), (1, 2), (1, 5), (2, 1), (2, 3), (2, 5), (3, 0), (3, 2), (3, 5), (4, 0), (4, 1), (4, 2
     24 [(0, 1), (0, 2), (0, 5), (0, 6), (1, 0), (1, 1), (1, 3), (1, 4), (2, 3), (2, 5), (2, 6), (3, 0), (3, 1), (3, 2), (3, 5), (4, 0), (4, 1
     25 [(0, 1), (0, 7), (1, 0), (1, 1), (1, 3), (1, 5), (1, 7), (2, 1), (2, 7), (3, 1), (3, 2), (3, 5), (3, 7), (4, 0), (4, 1), (4, 2), (4, 3
     26 [(0, 1), (0, 4), (0, 5), (0, 7), (1, 4), (1, 5), (1, 6), (1, 7), (2, 0), (2, 1), (2, 2), (2, 3), (3, 0), (3, 1), (3, 3), (3, 6), (3, 7
     27 [(0, 4), (0, 5), (1, 1), (1, 2), (1, 3), (1, 6), (1, 7), (2, 2), (2, 5), (2, 6), (2, 7), (3, 1), (3, 2), (3, 3), (3, 5), (3, 6), (3, 7
     28 [(0, 1), (0, 2), (0, 5), (0, 6), (1, 1), (1, 2), (1, 4), (1, 6), (1, 7), (2, 1), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4), (3, 7), (4, 0
     29 [(0, 2), (0, 3), (0, 5), (0, 6), (1, 1), (1, 3), (1, 6), (2, 0), (2, 3), (3, 0), (3, 1), (3, 2), (3, 3), (3, 4), (3, 6), (3, 7), (4, 0
     1 [15, 0, 10, 2, 6, 26]
     4 [28, 4, 24, 25, 13, 14]
     4 [18, 22, 11, 12, 27, 3]
```

```
5 [9, 21, 16, 17, 23, 5]
1 [20, 19, 29, 7, 1, 8]
```

```
horario = pywraplp.Solver.CreateSolver('SCIP')

# Sala, Dias, Horas
S, D, H = 5, 5, 8
# Proj, Colab
P, C = 5, 30
# Num Reunioes Semanais
R = 5

# Inicialização
x = {}
for s in range(S):
  for d in range(D):
    for h in range(H):
      for p in range(P):
        for c in range(C):
          x[s, d, h, p, c] = horario.BoolVar('x[%i, %i, %i, %i, %i]' % (s, d, h, p, c))

# Condições inerentes

# O líder tem de estar em todas as R reuniões do seu projecto
for p in range(P):
  horario.Add(sum(x[s, d, h, p, Projectos[p][1][0]] for s in range(S) for d in range(D) for h in range(H)) == Projectos[p][0])

# Slot (d, h) fora da disponibilidade do colaborador, logo não pode ser usado
for s in range(S):
  for d in range(D):
    for h in range(H):
      for p in range(P):
        for c in range(C):
          if (d, h) not in Colaboradores[c]:
            horario.Add(x[s, d, h, p, c] == 0)

# Colaboradores que não são do projecto não podem estar nele
for s in range(S):
  for d in range(D):
    for h in range(H):
      for p in range(P):
        for c in range(C):
          if c not in Projectos[p][1]:
            horario.Add(x[s, d, h, p, c] == 0)

# Limitações

# Cada sala tem alocada, no máximo, um projeto
for s in range(S):
  for d in range(D):
    for h in range(H):
      horario.Add(sum(x[s, d, h, p, Projectos[p][1][0]] for p in range(P)) <= 1)

# Cada colaborador de um projeto só pode estar numa sala
for d in range(D):
  for h in range(H):
    for p in range(P):
      for c in Projectos[p][1]:
        horario.Add(sum(x[s, d, h, p, c]  for s in range(S)) <= 1)

# Obrigações

# Participação de 50% com o líder incluido
for s in range(S):
  for d in range(D):
    for h in range(H):
      for p in range(P):
        horario.Add(sum(x[s, d, h, p, c] for c in Projectos[p][1]) >= 3 * x[s, d, h, p, Projectos[p][1][0]])

# Fazer o solve
status = horario.Solve()
if status == pywraplp.Solver.OPTIMAL:
  for p in range(P):
    presencas = []
    for dia in range(D):
      print("DIA {:<14}".format(dia), end="")
    print()
    for hora in range(H):
      for dia in range(D):
        print("||", hora, end=" || ")
        for s in range(S):
          if round(x[s, dia, hora, p, Projectos[p][1][0]].solution_value()) == 1:
            presencas.append([c for c in range(C) if round(x[s, dia, hora, p, c].solution_value())])
            print(s, end=" ")
          else:
            print("x", end=" ")
      print()
    print(presencas)
else:
  print("impossível")

     DIA 0             DIA 1             DIA 2             DIA 3             DIA 4
     || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x
     || 1 || x x x x x || 1 || x x x x x || 1 || x x x x x || 1 || x x x x x || 1 || x x x x x
     || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x
     || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x || 3 || 0 x x x x || 3 || x x x x x
     || 4 || x x x x x || 4 || x x x x x || 4 || x x x x x || 4 || x x x x x || 4 || x x x x x
     || 5 || x x x x x || 5 || x x x x x || 5 || x x x x x || 5 || x x x x x || 5 || x x x x x
     || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x
     || 7 || x x x x x || 7 || x x x x x || 7 || x x x x x || 7 || x x x x x || 7 || x x x x x
     [[2, 15, 26]]
     DIA 0             DIA 1             DIA 2             DIA 3             DIA 4
     || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x
     || 1 || x x x x x || 1 || x x x x x || 1 || 0 x x x x || 1 || x x x x x || 1 || x x x x x
     || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x
     || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x || 3 || 0 x x x x
     || 4 || x x x x x || 4 || 0 x x x x || 4 || x x x x x || 4 || x x x x x || 4 || x x x x x
     || 5 || x x x x x || 5 || x x x x x || 5 || x x x x x || 5 || x x x x x || 5 || 0 x x x x
     || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x
     || 7 || x x x x x || 7 || x x x x x || 7 || x x x x x || 7 || x x x x x || 7 || x x x x x
     [[4, 14, 25, 28], [14, 24, 25, 28], [13, 24, 28], [4, 13, 14, 24, 28]]
     DIA 0             DIA 1             DIA 2             DIA 3             DIA 4
     || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x
     || 1 || x x x x x || 1 || x x x x x || 1 || x x x x x || 1 || x x x x x || 1 || x x x x x
     || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x || 2 || 0 x x x x || 2 || x x x x x
     || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x
     || 4 || x x x x x || 4 || x x x x x || 4 || x x x x x || 4 || 0 x x x x || 4 || x x x x x
     || 5 || 0 x x x x || 5 || x x x x x || 5 || x x x x x || 5 || x x x x x || 5 || x x x x x
     || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x
     || 7 || 0 x x x x || 7 || x x x x x || 7 || x x x x x || 7 || x x x x x || 7 || x x x x x
     [[3, 11, 18, 22, 27], [3, 12, 18], [18, 22, 27], [3, 11, 18]]
     DIA 0             DIA 1             DIA 2             DIA 3             DIA 4
     || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x
     || 1 || x x x x x || 1 || x x x x x || 1 || x x x x x || 1 || x x x x x || 1 || 0 x x x x
     || 2 || 0 x x x x || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x
     || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x
     || 4 || x x x x x || 4 || x x x x x || 4 || x x x x x || 4 || x x x x x || 4 || 0 x x x x
|| 5 || x x x x x || 5 || x x x x x || 5 || 0 x x x x || 5 || x x x x x || 5 || x x x x x
|| 6 || 0 x x x x || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x
|| 7 || x x x x x || 7 || x x x x x || 7 || x x x x x || 7 || x x x x x || 7 || x x x x x
[[9, 21, 23], [5, 9, 17, 21, 23], [9, 16, 17, 21], [9, 17, 21, 23], [5, 9, 17, 21]]
DIA 0             DIA 1             DIA 2             DIA 3             DIA 4
|| 0 || x x x x x || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x || 0 || x x x x x
|| 1 || x x x x x || 1 || x x x x x || 1 || x x x x x || 1 || x x x x x || 1 || x x x x x
|| 2 || x x x x x || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x || 2 || x x x x x
|| 3 || x x x x x || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x || 3 || x x x x x
|| 4 || x x x x x || 4 || x x x x x || 4 || x x x x x || 4 || x x x x x || 4 || x x x x x
|| 5 || x x x x x || 5 || x x x x x || 5 || x x x x x || 5 || x x x x x || 5 || x x x x x
|| 6 || x x x x x || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x || 6 || x x x x x
|| 7 || x x x x x || 7 || x x x x x || 7 || 0 x x x x || 7 || x x x x x || 7 || x x x x x
[[1, 7, 20]]
```

✓ 0 s concluído à(s) 21:32