Trabalho 1 - Horário

Outubro, 2021

Inês Pires Presa - A90355
Tiago dos Santos Silva Peixoto Carriço - A91695

Variáveis:

Inputs do Problema

- S - Sala
- D - Dia
- T - Tempo
- P - Projeto
- C - Colaborador
- $projetos_i$ - associa a cada projeto o seu lider, número de reuniões semanais e lista de colaboradores, sendo $(1 \leq i \leq P)$
- $colaboradores_i$ - associa a cada colaborador a lista de *slots* em que está disponível, sendo $(1 \leq i \leq C)$

Auxiliares

- $x_{s,d,t,p}$ - representa a atribuição de uma sala s a um projeto p, que decorre no dia d no *slot* t
- $y_{c,d,t,p}$ - representa a alocação de um dado colaborador c, num projeto p a decorrer no dia d no *slot* t
- $z_{d,p}$ - representa a existência de alguma reunião do projeto p no dia d

Onde $s \in [1..S], c \in [1..C], d \in [1..D], t \in [1..T], p \in [1..P]$

Condições:

1. Líder tem de participar em todas as reuniões do projeto
2. Mínimo de 50% de colaboradores têm de participar na reunião do projeto
3. Cada projeto tem um dado número de reuniões semanais
4. Cada colaborador só pode ser colocado num slot em que esteja disponível
5. Cada sala só é utilizada para uma reunião em cada slot
6. Cada colaborador só participa num projeto de cada vez
7. Cada colaborador só pode ser colocado nos projetos em que está incluido
8. A variável $z_{d,p}$ tem valor 1 caso haja alguma reunião do projeto p no dia d e tem valor 0 caso contrário
9. Maximizar o número de dias em que cada projeto tem reuniões

```
!pip install ortools
```

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

Implementação:

Definir os valores para os *inputs* do problema

```
from ortools.linear_solver import pywraplp
import random
```

```
'''
#Exemplo 1
S, D, T, P, C = 2, 5, 5, 3, 8

# Número de projeto: (Líder de projeto, Número de reuniões semanais,
                      Lista de colaboradores)
```

```
projetos = {
    1: (1, 10, [1,3,5,8]),
    2: (2, 12,[1,2,4,7]),
    3: (3, 10,[2,5,6,7,8])
}

# Número do colaborador: Lista de slots

colaboradores = {
    1: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4),(2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    2: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4),(2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    3: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 1), (4, 2), (4, 3),(4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    4: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    5: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4),(2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    6: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    7: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    8: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)]
}'''

'''
#Exemplo 2
S, D, T, P, C = 2, 5, 5, 3, 8

projetos = {
    1: (1, 10, [1,3,5,8]),
    2: (2, 12,[1,2,4,7]),
    3: (3, 9,[2,3,5,6,7,8])
}

# Número do colaborador: Lista de slots

colaboradores = {
    1: [(1, 1), (1, 2), (1, 3), (1, 4),  (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (4, 1),
     (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    2: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2),  (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5),
     (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    3: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 4), (3, 5),
     (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    4: [(1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 2), (3, 3), (3, 4), (3, 5), (4, 1),
     (4, 2), (4, 3), (4, 4), (4, 5), (5, 2), (5, 3), (5, 4), (5, 5)],
    5: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    6: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), ],
    7: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1),  (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5),
     (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)],
    8: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4),
     (3, 5), (4, 2), (4, 3), (4, 4), (4, 5), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)]
}'''

#Exemplo 3

S, D, T, P, C = 3, 5, 7, 4, 7

projetos = {
    1: (1, 20, [1,3,5]),
    2: (2, 11,[1,2,4,7]),
    3: (3, 10,[2,3,5,6,7]),
    4: (4, 9, [1,2,3,4,7])
}

colaboradores ={
    1:[(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 7), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (3, 1),
       (3, 2), (3, 3), (3, 4), (3, 5),(3, 6),  (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (5, 1),
       (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7)],
    2:[(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7),
       (3, 1), (3, 2), (3, 3),  (3, 5), (3, 6), (3, 7), (4, 1), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (5, 1),
       (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7)],
    3:[(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 1),  (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (3, 1),
       (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7),
       (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6)],
```

```
    4:[(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7),
       (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6),
       (4, 7), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7)],
    5:[(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7),
       (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6),
       (4, 7), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7)],
    6:[(1, 1), (1, 2), (1, 3), (1, 5), (1, 6), (1, 7), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7),  (3, 2),
       (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (5, 1), (5, 2),
       (5, 3), (5, 4), (5, 5), (5, 6), (5, 7)],
    7:[(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 7),  (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (3, 1), (3, 2),
       (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (5, 1),
       (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7)]
}
```

Inicialização do *solver* e declaração das matrizes de alocação x e y

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

x = {}
for s in range(1, S+1):
  x[s] = {}
  for d in range(1,D+1):
    x[s][d] = {}
    for t in range(1,T+1):
      x[s][d][t] = {}
      for p in range(1,P+1):
        x[s][d][t][p] = solver.BoolVar('x[%i][%i][%i][%i]' % (s,d,t,p))

y = {}
for c in range(1, C+1):
  y[c] = {}
  for d in range(1,D+1):
    y[c][d] = {}
    for t in range(1,T+1):
      y[c][d][t] = {}
      for p in range(1,P+1):
        y[c][d][t][p] = solver.BoolVar('y[%i][%i][%i][%i]' % (c,d,t,p))

z = {}
for d in range(1,D+1):
  z[d] = {}
  for p in range(1,P+1):
    z[d][p] = solver.BoolVar('z[%i][%i]' % (d,p))
```

Modelação das restrições e sua introdução no *solver*

1. Líder tem de participar em todas as reuniões do projeto

$$\forall_{1\leq d\leq D}\cdot\forall_{1\leq p\leq P}\cdot\forall_{1\leq t\leq T}\quad(\sum_{1\leq s\leq S}x_{s,d,t,p})=y_{lider,d,t,p}$$

```
for d in range(1,D+1):
  for p in range(1,P+1):
    for t in range(1,T+1):
      lider = projetos[p][0]
      solver.Add(sum([x[s][d][t][p] for s in range(1,S+1)]) == y[lider][d][t][p])
```

2. Mínimo de 50% dos colaboradores (Min = `0.5*len(projetos[p][2])`) têm de participar na reunião do projeto

$$\forall_{1\leq d\leq D}\cdot\forall_{1\leq p\leq P}\cdot\forall_{1\leq t\leq T}\quad(\sum_{1\leq c\leq C}y_{c,d,t,p})\leq Min*y_{lider,d,t,p}$$

```
for d in range(1,D+1):
  for p in range(1,P+1):
    for t in range(1,T+1):
      lider = projetos[p][0]
      colabs = projetos[p][2]
      solver.Add(sum([y[c][d][t][p] for c in colabs]) >= 0.5*len(colabs)*y[lider][d][t][p])
```

3. Cada projeto tem um dado número de reuniões semanais (R_p = `projetos[p][1]`)

$$\forall_{1 \leq p \leq P} \sum_{1 \leq s \leq S,\, 1 \leq t \leq T,\, 1 \leq d \leq D} x_{s,d,t,p} == R_p$$

```
for projeto, tuplo in projetos.items():
  lider, reunioes, lista = tuplo

  solver.Add(sum([x[s][d][t][projeto] for s in range(1, S+1) for d in range(1, D+1) for t in range(1, T+1)]) == reunioes)
```

4. Cada colaborador só pode ser colocado num slot em que esteja disponível

$$\forall_{1 \leq d \leq D} \cdot \forall_{1 \leq p \leq P} \cdot \forall_{1 \leq t \leq T} \cdot \forall_{1 \leq c \leq C} \quad (d,t) \notin colaboradores_c \rightarrow y_{c,d,t,p} = 0$$

```
for d in range(1,D+1):
  for t in range(1,T+1):
    for p in range(1,P+1):
      for c in range(1,C+1):
        if (d,t) not in colaboradores[c]:
          solver.Add(y[c][d][t][p] == 0)
```

5. Cada sala só é utilizada para uma reunião em cada slot

$$\forall_{1 \leq d \leq D} \cdot \forall_{1 \leq t \leq T} \cdot \forall_{1 \leq s \leq S} \quad \sum_{1 \leq p \leq P} x_{s,d,t,p} \leq 1$$

```
for s in range(1,S+1):
  for d in range(1,D+1):
    for t in range(1,T+1):
      solver.Add(sum([x[s][d][t][p] for p in range(1,P+1)]) <= 1)
```

6. Cada colaborador só participa num projeto de cada vez

$$\forall_{1 \leq d \leq D} \cdot \forall_{1 \leq t \leq T} \cdot \forall_{1 \leq c \leq C} \quad \sum_{1 \leq p \leq P} y_{c,d,t,p} \leq 1$$

```
for c in range(1, C+1):
  for d in range(1, D+1):
    for t in range(1,T+1):
      solver.Add(sum([y[c][d][t][p] for p in range(1,P+1)]) <= 1)
```

7. Cada colaborador só pode ser colocado nos projetos em que está incluído

$$\forall_{1 \leq d \leq D} \cdot \forall_{1 \leq p \leq P} \cdot \forall_{1 \leq t \leq T} \cdot \forall_{1 \leq c \leq C} \quad c \notin projetos_{p,2} \rightarrow y_{c,d,t,p} = 0$$

```
for c in range(1, C+1):
  for d in range(1, D+1):
    for t in range(1, T+1):
      for p in range(1,P+1):
        if c not in projetos[p][2]:
          solver.Add(y[c][d][t][p] == 0)
```

8. A variável $z_{d,p}$ tem valor 1 caso haja alguma reunião do projeto p no dia d e tem valor 0 caso contrário

$$(\forall_{1 \leq d \leq D} \cdot \forall_{1 \leq p \leq P} \quad z_{d,p} \leq (\sum_{1 \leq s \leq S,\, 1 \leq t \leq T} x_{s,d,t,p})) \wedge (\forall_{1 \leq d \leq D} \cdot \forall_{1 \leq p \leq P} \cdot \forall_{1 \leq s \leq S} \cdot \forall_{1 \leq t \leq T} \quad z_{d,p} \geq x_{s,d,t,p})$$

```
for d in range(1, D+1):
  for p in range(1, P+1):
    solver.Add(z[d][p] <= sum([x[s][d][t][p] for s in range(1, S+1) for t in range(1, T+1)]))
    for s in range(1, S+1):
      for t in range(1, T+1):
        solver.Add(z[d][p] >= x[s][d][t][p])
```

9. Maximizar o número de dias em que cada projeto tem reuniões

```
solver.Maximize(sum([z[d][p] for d in range(1,D+1) for p in range(1,P+1)]))
```

Procura da solução do problema

```
r = solver.Solve()
if r == pywraplp.Solver.OPTIMAL:
  print("Solução encontrada")
else:
  print("Não foi encontrada solução")
```

```
Solução encontrada
```

Impressão do calendário semanal obtido:

```
pip install texttable
```

```
Requirement already satisfied: texttable in /usr/local/lib/python3.7/dist-packages (1.6.4)
```

```
from tabulate import tabulate

if r == pywraplp.Solver.OPTIMAL:

  head = ["Dia %i" % d for d in range(1,D+1)]
  head.insert(0, "Slots")

  h = [[] for x in range(0, T+1)]
  for t in range(1,T+1):
    h[t].insert(0, "Slot %i" % t)
  for d in range(1,D+1):
    for t in range(1,T+1):
      h[t].insert(d, "")
      for p in range(1,P+1):
        for s in range(1, S+1):
          if round(x[s][d][t][p].solution_value()) == 1:
            h[t][d] += ("*Projeto %i - sala %i\n Colab: " % (p,s))
            for c in range(1,C+1):
              if round(y[c][d][t][p].solution_value()) == 1:
                h[t][d] += ("%i, " % c)
            h[t][d] = h[t][d][:-2]
            h[t][d] += ("\n\n")


  print(tabulate(h, headers=head))
```

Slots	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5
Slot 1	*Projeto 3 - sala 3 Colab: 2, 3, 6	*Projeto 1 - sala 2 Colab: 1, 5	*Projeto 3 - sala 1 Colab: 3, 5, 7	*Projeto 3 - sala 2 Colab: 2, 3, 5	*Projeto 4 - sala 2 Colab: 1, 2, 4
		*Projeto 4 - sala 1 Colab: 2, 3, 4	*Projeto 4 - sala 2 Colab: 1, 2, 4	*Projeto 4 - sala 3 Colab: 1, 4, 7	
Slot 2	*Projeto 3 - sala 3 Colab: 2, 3, 6	*Projeto 2 - sala 3 Colab: 2, 4, 7	*Projeto 4 - sala 3 Colab: 1, 3, 4	*Projeto 1 - sala 2 Colab: 1, 5	*Projeto 3 - sala 2 Colab: 3, 5, 6
Slot 3	*Projeto 3 - sala 3 Colab: 3, 6, 7	*Projeto 1 - sala 1 Colab: 1, 5	*Projeto 1 - sala 2 Colab: 1, 5	*Projeto 2 - sala 2 Colab: 1, 2	*Projeto 1 - sala 3 Colab: 1, 5
	*Projeto 4 - sala 1 Colab: 1, 2, 4	*Projeto 2 - sala 3 Colab: 2, 4, 7	*Projeto 2 - sala 1 Colab: 2, 4	*Projeto 4 - sala 3 Colab: 3, 4, 7	*Projeto 2 - sala 2 Colab: 2, 4
Slot 4	*Projeto 1 - sala 2 Colab: 1, 5	*Projeto 1 - sala 1 Colab: 1, 3	*Projeto 1 - sala 1 Colab: 1, 5	*Projeto 1 - sala 1 Colab: 1, 5	*Projeto 1 - sala 2 Colab: 1, 5
	*Projeto 4 - sala 3 Colab: 2, 3, 4	*Projeto 2 - sala 2 Colab: 2, 4			*Projeto 3 - sala 1 Colab: 3, 6, 7
Slot 5	*Projeto 1 - sala 1 Colab: 1, 3	*Projeto 1 - sala 1 Colab: 1, 5	*Projeto 1 - sala 3 Colab: 1, 5	*Projeto 1 - sala 2 Colab: 1, 5	*Projeto 2 - sala 1 Colab: 2, 4
	*Projeto 2 - sala 3 Colab: 2, 4		*Projeto 2 - sala 2 Colab: 2, 4	*Projeto 3 - sala 1 Colab: 2, 3, 6	*Projeto 3 - sala 2 Colab: 3, 5, 6
Slot 6	*Projeto 2 - sala 3 Colab: 2, 4	*Projeto 1 - sala 1 Colab: 1, 5	*Projeto 1 - sala 2 Colab: 1, 5	*Projeto 1 - sala 1 Colab: 1, 5	*Projeto 1 - sala 1 Colab: 1, 5
		*Projeto 3 - sala 3 Colab: 2, 3, 6			
Slot 7	*Projeto 1 - sala 1 Colab: 1, 3	*Projeto 1 - sala 1 Colab: 1, 5			*Projeto 4 - sala 1 Colab: 1, 2, 4
	*Projeto 2 - sala 3 Colab: 2, 4				