

Exercicio1

November 1, 2020

Trabalho Realizado Por:

Carlos Ferreira - a87953 Daniel Ribeiro - a87994

Exercício 1

1. Pretende-se construir um horário semanal para o plano de reuniões de projeto de uma “StartUp” de acordo com as seguintes condições:
 1. Cada reunião ocupa uma sala (enumeradas $1..S$) durante um “slot” (tempo, 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 líder, é um conjunto de “slots” (“inputs” do problema).

A variável “o colaborador vai à reunião na sala s , no tempo t , no dia d , e trabalhará no projeto p ” vai ser representada da forma

$$0 \leq x_{s,t,d,p,c} \leq 1$$

Limitações (que impõem limites máximos à alocação)

1. Cada colaborador só estar em uma reunião de cada vez:

$$\forall_{d < D} \cdot \forall_{t < T} \cdot \forall_{c < C} \sum_{s < S, p < P} x_{s,t,d,p,c} = 1$$

2. Não pode haver a mesma reunião ao mesmo tempo de projetos iguais em salas diferentes:

$$\forall_{p < P} \cdot \forall_{d < D} \cdot \forall_{t < T} \sum_{s < S, l < L} x_{s,t,d,p,l} = 1$$

Obrigações (que impõem limites mínimos à alocação)

1. Cada projeto realiza n de reuniões semanais (como o líder está presente em todas, o líder tem que fazer n reuniões semanais:

$$\forall_{p < P} \cdot \sum_{s < S, t < T, d < D, l < L} x_{s,t,d,p,l} = N, \text{ onde } l \text{ o lder do projeto } p$$

2. No mínimo 50% dos colaboradores têm de aparecer no projeto:

$$\forall_{d < D} \cdot \forall_{t < T} \cdot \forall_{s < S} \cdot \forall_{p < P} \cdot \sum_{c < C} x_{s,t,d,p,c} \geq \frac{C}{2} + C \bmod 2 \quad \bigvee \sum_{c < C} x_{s,t,d,p,c} = 0$$

3. O líder têm de aparecer em todas as reuniões do seu projeto (se um colaborador for a uma reunião o líder tem que estar lá):

$$\forall_{s < S} \cdot \forall_{t < T} \cdot \forall_{d < D} \cdot \forall_{p < P} \cdot \forall_{c < C} \quad (x_{s,t,d,p,c} == 1) \rightarrow (x_{s,t,d,p,l} == 1)$$

, com l o líder do projeto p

4. Cada sala tem alocado no máximo um projeto de cada vez:

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

, com l o líder do projeto p

```
[46]: from z3 import *

#Exemplo1
#S,T,D,P,C = 2,2,2,2,4

#Exemplo2
S,T,D,P,C = 0,0,0,0,0

def constroiHorario(reunioesSemanais, colaboradoresProjeto, lider ,□
↪disponibilidade):

    horario = Solver()
    X = {}

    for s in range(1,S+1):
        for t in range(1,T+1):
            for d in range(1,D+1):
                for p in range(1,P+1):
                    for c in range(1,C+1):
                        #Se pertencer ao projeto tem de estar presente ou não
```

```

        if c in colaboradoresProjeto[p]:
            X[s,t,d,p,c] = Int(str(s) + '_' + str(t) + '_' +
↪str(d) + '_' + str(p) + '_' + str(c))
            horario.add (X[s,t,d,p,c] >= 0 , X[s,t,d,p,c] <=1)
            #Se não pertencer ao projeto não está presente
        else:
            X[s,t,d,p,c] = Int(str(s) + '_' + str(t) + '_' +
↪str(d) + '_' + str(p) + '_' + str(c))
            horario.add (X[s,t,d,p,c] == 0)

    #print (X)

    # o colaborador não pode estar em mais que uma reuniao ao mesmo tempo
↪(Limitação nº1)

    for t in range(1,T+1):
        for d in range(1,D+1):
            for c in range(1,C+1):
                horario.add(Sum([X[s,t,d,p,c] for s in range(1,S+1) for p in
↪range (1,P+1)]) <= 1 )

    # Não pode haver a mesma reunião ao mesmo tempo de projetos iguais em salas
↪diferentes (Limitacao nº2)
    # Não necessário pois como para todo tempo e dia , todo colaborador apenas
↪pode estar numa sala e projeto ( limitação nº1),
    # e como o lider participará em todas as reuniões de um projeto não será
↪possível a ocorrência de reuniões do mesmo projeto
    # em salas diferentes.

    '''for t in range(1,T+1):
        for d in range(1,D+1):
            for p in range(1,P+1):
                horario.add(Sum([X[s,t,d,p,lider[p]] for s in range(1,S+1)]) <=
↪1)
    '''

    # se o projeto necessita de n reunioes <=> o lider tem de ir a n reunioes
↪desse projeto (Obrigação nº 1)

    for p in range(1,P+1):
        horario.add(Sum([X[s,t,d,p,lider[p]] for s in range (1,S+1) for d in
↪range (1,D+1) for t in range (1,T+1)]) == reunioesSemanais[p] )

```

```

# cada reuniao necessita no minimo de 50% dos colaboradores do projeto
→ (Obrigação nº2)

# Para toda sala , todo tempo , todo dia , todo projeto , se a soma dos
→ colaboradores for 0 significa que não existirá reunião
# , caso contrário , esta soma terá de ser maior que 50% dos colaboradores
→ , caso o número de colaboradores seja ímpar , juntamos
# o resto da divisão inteira de C por 2.

for s in range(1,S+1):
    for t in range(1,T+1):
        for d in range(1,D+1):
            for p in range(1,P+1):
                horario.add(Or(Sum([X[s,t,d,p,c] for c in
→ colaboradoresProjeto[p] ]) == 0 , Sum([X[s,t,d,p,c] for c in
→ colaboradoresProjeto[p] ]) >= ((len(colaboradoresProjeto[p])/2) +
→ (len(colaboradoresProjeto[p])%2))))

# Se um colaborador vai, o lider tambem tem de ir (Obrigação nº 3)

for s in range(1,S+1):
    for t in range(1,T+1):
        for d in range(1,D+1):
            for p in range(1,P+1):
                for c in range(1,C+1):
                    # horario.add(X[s,t,d,p,c] - X[s,t,d,p,lider[p]] <= 0)
                    horario.add(Implies(X[s,t,d,p,c]==1
→ ,X[s,t,d,p,lider[p]] ==1))

# Em cada sala para um determinado slot (tempo , dia) só decorre a reunião
→ para no máximo 1 projeto (Obrigação nº4)

for t in range(1,T+1):
    for d in range(1,D+1):
        for s in range(1,S+1):
            horario.add(Sum([X[s,t,d,p,lider[p]] for p in
→ range(1,P+1)]) <= 1)

# nao disponivel <=> nao pode ir a reuniao (disponibilidade)

for c in range(1,C+1):
    for t in range(1,T+1):
        for d in range(1,D+1):

```

```

        if (t,d) not in disponibilidade[c]:
            horario.add(And([(X[s,t,d,p,c] == 0) for s in range (1,S+1)]
→for p in range(1,P+1))))

#print(horario)
#print("S,D,T,P,C")

r = horario.check()
print(r)

if r == sat:
    m = horario.model()
    #print(m.eval(Sum(list(X.values()))))
    #print(m)
    for d in range (1,D+1):
        print ("Dia" , d,":")
        for p in range (1,P+1):
            for t in range (1,T+1):
                for s in range (1,S+1):
                    for c in range (1,C+1):
                        if m[X[s,t,d,p,c]] == 1:
                            print
→("Projeto",p,"Tempo",t,"Sala",s,"Colaborador",c)

```

Vamos fazer Testes a cada Restrição:

Limitação nº1: Cada colaborador só pode trabalhar em um projeto de cada vez:

[47]: *# 1 tempo , 1 dia , 2 projetos para o mesmo lider*

```

S,T,D,P,C = 1,1,1,2,1
reunioesSemanais = {1:1,2:1}
colaboradoresProjeto = {1:[1],2:[1]}
lider = {1:1, 2:1}
disponibilidade = {1:[(1,1)]}
constroiHorario (reunioesSemanais, colaboradoresProjeto, lider ,
→disponibilidade)

```

unsat

Limitação nº2: A reunião do projeto não pode ocorrer em salas diferentes ao mesmo tempo.

[48]: *# 2 salas , 1 tempo , 1 dia , 1 projeto que necessita de 2 reunioes mas so*
↪ existe um tempo logo deve dar unsat

```
S,T,D,P,C = 2,1,1,1,1
reunioesSemanais = {1:2}
colaboradoresProjeto = {1:[1]}
lider = {1:1}
disponibilidade = {1:[(1,1)]}
constroiHorario (reunioesSemanais, colaboradoresProjeto, lider ,
↪disponibilidade)
```

unsat

Obrigação nº1: Cada projeto realiza de reuniões semanais (como o lider está presente em todas, o lider tem que fazer reuniões semanais):

[49]: *# 3 reuniões semanais mas lider so esta disponivel para 2 , logo nao e possivel*
↪ ocorrer as 3 reunões

```
S,T,D,P,C = 1,2,1,1,1
reunioesSemanais = {1:3}
colaboradoresProjeto = {1:[1]}
lider = {1:1}
disponibilidade = {1:[(1,1),(2,1)]}
constroiHorario (reunioesSemanais, colaboradoresProjeto, lider ,
↪disponibilidade)
```

unsat

Obrigação nº2: No mínimo 50% dos colaboradores têm de aparecer no projeto:

[50]: *# 3 colaboradores mas apenas o lider pode ir , no minimo tinham de ir 2 a*
↪ reunião

```
S,T,D,P,C = 1,1,1,1,3
reunioesSemanais = {1:1}
colaboradoresProjeto = {1:[1,2,3]}
lider = {1:1}
disponibilidade = {1:[(1,1)],2:[],3:[]}
constroiHorario (reunioesSemanais, colaboradoresProjeto, lider ,
↪disponibilidade)
```

unsat

[51]:

```
S,T,D,P,C = 1,2,1,1,5
reunioesSemanais = {1:1}
colaboradoresProjeto = {1:[1,2,3,4,5]}
lider = {1:1}
disponibilidade = {1:[(1,1),(2,1)],2:[(1,1)],3:[],4:[(2,1)],5:[(2,1)]}
```

```
constroiHorario (reunioesSemanais, colaboradoresProjeto, lider ,  
↪disponibilidade)
```

sat

Dia 1 :

Projeto 1 Tempo 2 Sala 1 Colaborador 1

Projeto 1 Tempo 2 Sala 1 Colaborador 4

Projeto 1 Tempo 2 Sala 1 Colaborador 5

Obrigação nº3: O líder têm de aparecer em todas as reuniões do seu projeto:

```
[52]: S,T,D,P,C = 1,2,1,1,2  
reunioesSemanais = {1:2}  
colaboradoresProjeto = {1:[1,2]}  
lider = {1:1}  
disponibilidade = {1:[(1,1)],2:[(1,1),(2,1)]}  
constroiHorario (reunioesSemanais, colaboradoresProjeto, lider ,  
↪disponibilidade)
```

unsat

Obrigação nº4: Cada sala tem alocado no máximo um projeto de cada vez:

```
[53]: # 2 projetos , 1 sala , 1 tempo , 1 dia  
  
S,T,D,P,C = 1,1,1,2,2  
reunioesSemanais = {1:1,2:1}  
colaboradoresProjeto = {1:[1],2:[2]}  
lider = {1:1,2:2}  
disponibilidade = {1:[(1,1)],2:[(1,1)]}  
constroiHorario (reunioesSemanais, colaboradoresProjeto, lider ,  
↪disponibilidade)
```

unsat

Seguidamente alguns exemplos do bom funcionamento do nosso programa:

Exemplo 1:

```
[58]: S,T,D,P,C = 2,2,2,2,5  
reunioesSemanais = {1:1,2:3}  
colaboradoresProjeto = {1:[1,2,4,5] , 2:[3,5]}  
lider = {1:4,2:5}  
disponibilidade = {1: [(1,1)] , 2: [] , 3: [(1,2),(2,2)] , 4: [(1,1)] , 5:  
↪[(1,1),(2,1) ,(1,2),(2,2)]}  
constroiHorario (reunioesSemanais, colaboradoresProjeto, lider ,  
↪disponibilidade)
```

sat

Dia 1 :

Projeto 1 Tempo 1 Sala 1 Colaborador 1

```

Projeto 1 Tempo 1 Sala 1 Colaborador 4
Projeto 2 Tempo 1 Sala 2 Colaborador 5
Projeto 2 Tempo 2 Sala 2 Colaborador 5
Dia 2 :
Projeto 2 Tempo 1 Sala 2 Colaborador 3
Projeto 2 Tempo 1 Sala 2 Colaborador 5

```

Exemplo 2:

```

[54]: S,T,D,P,C = 2,4,5,6,8
      #40 possiveis reunioes 2x4x5
      reunioesSemanais = {1:4,2:6,3:10,4:5,5:5,6:10}
      colaboradoresProjeto = {1:[1,3,4,8],2:[2,8],3:[3,7],4:[4],5:[5,7],6:[6,8,5]}
      lider = {1:1, 2:2,3:3,4:4,5:5,6:6}

      disponibilidade = {}

      for i in range(1,9):
          disponibilidade[i] =[(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)]

      constroiHorario (reunioesSemanais, colaboradoresProjeto, lider ,
          ↳disponibilidade)

```

```

sat
Dia 1 :
Projeto 1 Tempo 1 Sala 2 Colaborador 1
Projeto 1 Tempo 1 Sala 2 Colaborador 8
Projeto 2 Tempo 1 Sala 1 Colaborador 2
Projeto 2 Tempo 3 Sala 1 Colaborador 2
Projeto 2 Tempo 4 Sala 2 Colaborador 2
Projeto 3 Tempo 3 Sala 2 Colaborador 3
Projeto 4 Tempo 2 Sala 2 Colaborador 4
Projeto 6 Tempo 2 Sala 1 Colaborador 5
Projeto 6 Tempo 2 Sala 1 Colaborador 6
Projeto 6 Tempo 2 Sala 1 Colaborador 8
Projeto 6 Tempo 4 Sala 1 Colaborador 6
Projeto 6 Tempo 4 Sala 1 Colaborador 8
Dia 2 :
Projeto 1 Tempo 2 Sala 2 Colaborador 1
Projeto 1 Tempo 2 Sala 2 Colaborador 8
Projeto 3 Tempo 3 Sala 2 Colaborador 3
Projeto 3 Tempo 4 Sala 2 Colaborador 3
Projeto 4 Tempo 1 Sala 1 Colaborador 4
Projeto 5 Tempo 1 Sala 2 Colaborador 5
Projeto 6 Tempo 2 Sala 1 Colaborador 5
Projeto 6 Tempo 2 Sala 1 Colaborador 6

```


Projeto 6 Tempo 3 Sala 1 Colaborador 5
Projeto 6 Tempo 3 Sala 1 Colaborador 6
Projeto 6 Tempo 3 Sala 1 Colaborador 8
Projeto 6 Tempo 4 Sala 1 Colaborador 5
Projeto 6 Tempo 4 Sala 1 Colaborador 6

Dia 3 :

Projeto 1 Tempo 2 Sala 2 Colaborador 1
Projeto 1 Tempo 2 Sala 2 Colaborador 4
Projeto 2 Tempo 4 Sala 2 Colaborador 2
Projeto 3 Tempo 2 Sala 1 Colaborador 3
Projeto 3 Tempo 3 Sala 2 Colaborador 3
Projeto 4 Tempo 1 Sala 1 Colaborador 4
Projeto 5 Tempo 1 Sala 2 Colaborador 5
Projeto 6 Tempo 3 Sala 1 Colaborador 5
Projeto 6 Tempo 3 Sala 1 Colaborador 6
Projeto 6 Tempo 3 Sala 1 Colaborador 8
Projeto 6 Tempo 4 Sala 1 Colaborador 5
Projeto 6 Tempo 4 Sala 1 Colaborador 6

Dia 4 :

Projeto 1 Tempo 4 Sala 2 Colaborador 1
Projeto 1 Tempo 4 Sala 2 Colaborador 8
Projeto 2 Tempo 2 Sala 1 Colaborador 2
Projeto 3 Tempo 3 Sala 2 Colaborador 3
Projeto 3 Tempo 4 Sala 1 Colaborador 3
Projeto 4 Tempo 1 Sala 1 Colaborador 4
Projeto 4 Tempo 3 Sala 1 Colaborador 4
Projeto 5 Tempo 1 Sala 2 Colaborador 5
Projeto 5 Tempo 2 Sala 2 Colaborador 5

Dia 5 :

Projeto 2 Tempo 1 Sala 2 Colaborador 2
Projeto 3 Tempo 2 Sala 2 Colaborador 3
Projeto 3 Tempo 3 Sala 1 Colaborador 3
Projeto 3 Tempo 4 Sala 2 Colaborador 3
Projeto 5 Tempo 3 Sala 2 Colaborador 5
Projeto 6 Tempo 1 Sala 1 Colaborador 6
Projeto 6 Tempo 1 Sala 1 Colaborador 8
Projeto 6 Tempo 2 Sala 1 Colaborador 5
Projeto 6 Tempo 2 Sala 1 Colaborador 6
Projeto 6 Tempo 2 Sala 1 Colaborador 8
Projeto 6 Tempo 4 Sala 1 Colaborador 5
Projeto 6 Tempo 4 Sala 1 Colaborador 6

[]: