

Lógica Computacional TP1

Realizado por: Miguel Gonçalves a90416 João
Nogueira a87973



Universidade do Minho
Escola de Ciências

Exercicio 1

1. Pretende-se construir um horário semanal para o plano de reuniões de projeto de uma “StartUp” de acordo com as seguintes condições:
 - A. 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.
 - B. Cada reunião tem associado um projeto (enumerados 1..P) e um conjunto de participantes. Os diferentes colaboradores são enumerados 1..C.
 - C. 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.
 - D. 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).

Limitações

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

$$\begin{aligned} & \forall_{d < D} \cdot \forall_{t < T} \\ & \cdot \forall_{c < C} \\ & \sum_{s < S, p < P} x_{s,t,d,p,c} \\ & \leq 1 \end{aligned}$$

Obrigações

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

$$\begin{aligned} & \forall_{p < P} \cdot \\ & \sum_{s < S, t < T, d < D, l < L} x_{s,t,d,p,l} = N, \text{ onde } l \text{ é o líder do projeto } p \end{aligned}$$

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

$$\begin{aligned} & \forall_{d < D} \cdot \forall_{t < T} \\ & \cdot \forall_{s < S} \cdot \forall_{p < P} \cdot \\ & \sum_{c < C} x_{s,t,d,p,c} \\ & > \frac{C}{2} \end{aligned}$$

$$- 2 \\ + C \bmod 2 \bigvee \\ \sum_{c < C} x_{s,t,d,p,c} = 0$$

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

$$\begin{aligned} & \forall s < S \cdot \forall t < T \\ & \cdot \forall d < D \cdot \forall p < P \\ & \cdot \forall c < C \\ & (x_{s,t,d,p,c} \\ & == 1) \\ & \rightarrow (x_{s,t,d,p,l} \\ & == 1) \end{aligned}$$

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

$$\begin{aligned} & \forall d < D \cdot \forall t < T \\ & \cdot \forall s < S \\ & \sum_{p < P} x_{s,t,d,p,l} \leq 1 \end{aligned}$$

Versão com o z3

In []:

```
!pip install z3-solver
```

In [40]:

```
from z3 import *

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

def horario(reunioes, colaborador, lider , disponibilidade):

    horario = Solver()
    X = {}

    for s in range(1,S+1): #Para cada sala enumerada 1...S
        for t in range(1,T+1): #Para cada tempo enumerado 1..T
            for d in range(1,D+1): #Para cada dia enumerado 1..D
                for p in range(1,P+1): #Para cada projeto enumerado 1..P
                    for c in range(1,C+1): #Para cada colaborador enumerado 1..C

                        #Se pertencer ao projeto tem de estar presente ou não (ou é 1 ou
0)
                        if c in colaborador[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 é 0
                        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)

    # Limitação 1 - Cada colaborador só estar em uma reunião de cada vez

    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 )

# Obrigação 1 - Cada projeto realiza n de reuniões semanais

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)]) == reunioes[p] )

# Obrigação 2 - No mínimo 50% dos colaboradores têm de aparecer no projeto
# Se a soma dos colaboradores for 0 significa que não existirá reunião, caso contrári
o , 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 colaborador[p] ]) == 0 ,
sum([X[s,t,d,p,c] for c in colaborador[p] ]) >= ((len(colaborador[p])/2) + (len(colabora
dor[p])%2))))

# Obrigação 3 - O líder têm de aparecer em todas as reuniões do seu projeto

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(Implies(X[s,t,d,p,c]==1 ,X[s,t,d,p,lider[p]] ==1))

# Obrigação 4 - Cada sala tem alocado no máximo um projeto de cada vez

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 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 i
n range(1,P+1)]))

r = horario.check()
print(r)

if r == sat:
    m = horario.model()
    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)

```

In [38]:

```
S,T,D,P,C = 2,3,2,2,5
reunioes = {1:1,2:3}
colaborador = {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)]}
horario (reunioes, colaborador, lider , disponibilidade)
```

```
sat
Dia 1 :
Projeto 1 Tempo 1 Sala 2 Colaborador 1
Projeto 1 Tempo 1 Sala 2 Colaborador 4
Projeto 2 Tempo 1 Sala 1 Colaborador 5
Projeto 2 Tempo 2 Sala 2 Colaborador 5
Dia 2 :
Projeto 2 Tempo 2 Sala 1 Colaborador 5
```

Exemplo 2

In [39]:

```
S,T,D,P,C = 2,4,5,6,8

reunioes = {1:4,2:6,3:10,4:5,5:5,6:10}
colaborador = {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)]

horario (reunioes, colaborador, lider , disponibilidade)
```

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

```

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

```

Versão SCIP

In []:

```
!pip install ortools
```

In [49]:

```

from ortools.linear_solver import pywraplp

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

def horario(reunioes, colaborador, lideres, disponibilidade):

    solver = pywraplp.Solver.CreateSolver('SCIP')
    x = {}

    for s in range(1,S+1): #Para cada sala enumerada 1...S
        for t in range(1,T+1): #Para cada tempo enumerado 1..T
            for d in range(1,D+1): #Para cada dia enumerado 1..D
                for p in range(1,P+1): #Para cada projeto enumerado 1..P
                    for c in range(1,C+1): #Para cada colaborador enumerado 1..C
                        #Se pertencer ao projeto ou é 0 ou 1
                        if c in colaborador[p]:
                            x[s,t,d,p,c] = solver.BoolVar("x[%d,%d,%d,%d,%d]" % (s,t,d,p,c))
                            solver.Add(x[s,t,d,p,c] >= 0, x[s,t,d,p,c] <= 1)

                        #Se não pertencer é 0
                        else:
                            x[s,t,d,p,c] = solver.BoolVar("x[%d,%d,%d,%d,%d]" % (s,t,d,p,c))
                            solver.Add(x[s,t,d,p,c] == 0)

    # Limitação 1 - O colaborador não pode estar em mais que uma reuniao ao mesmo tempo (Li
    mitação n°1)
    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 )

# Obrigação 1 - Cada projeto realiza n de reuniões semanais
for p in range(P+1):
    solver.Add(sum([x[s,t,d,p,lider[p]] for s in range (1,S+1) for d in range (1,D+1) fo
r t in range (1,T+1)]) == reunioes[p] )

# Obrigação 2 - No mínimo 50% dos colaboradores têm de aparecer no projeto
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):
                solver.Add(Or(sum([x[s,t,d,p,c] for c in colaborador[p] ]) == 0 , sum([x[s,t,
d,p,c] for c in colaborador[p] ]) >= ((len(colaborador[p])/2) + (len(colaborador[p])%2)
))

# Obrigação 3 - O líder têm de aparecer em todas as reuniões do seu projeto
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):
                    solver.Add(Implies(x[s,t,d,p,c]==1 ,x[s,t,d,p,lider[p]] ==1))

# Obrigação 4 - Cada sala tem alocado no máximo um projeto de cada vez
for t in range(1,T+1):
    for d in range(1,D+1):
        for s in range(1,S+1):
            solver.Add(sum([x[s,t,d,p,lider[p]] for p in range(1,P+1)]) <= 1)

status = solver.Solve()
if status == pywraplp.Solver.OPTIMAL:
    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 round(x[s,t,d,p,c].solution_value()) == 1:
                            print ("Projeto",p,"Tempo",t,"Sala",s,"Colaborador",c)

```

Exemplo 1

In []:

```

S,T,D,P,C = 2,2,2,2,5
reunioes = {1:1,2:3}
colaborador = {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)]}
horario(reunioes, colaborador, lider , disponibilidade)

```