LCTrabalho1Exercicio1

November 1, 2020

1 Trabalho 1

1.1 Lógica Computacional 2020-2021

O propósito deste trabalho é a análise de problemas de alocação usando técnicas de SAT, em lógica proposicional, e IP em lógica linear inteira.

Trabalho realizado por:

> 1. Paulo Costa - A87986
> 2. André Araújo - A87987

1.1.1 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" $(\text{tempo}, \text{dia})$. 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).

1.1.2 Ideias:

Primeiramente, colocamos as Limitações e as Obrigações do problema que temos apresentado.
Pensamos que era muito útil delinear as funções que vão tornar possível a realização deste exercício, funções essas que delimitam os limites máximos de cada parâmetro, e ainda limites mínimos para estes.
Antes de mais nada, achamos que era importantíssimo definir que, uma sala só poderia ter um projeto de cada vez e que cada colaborador só pode ir a uma reunião de cada vez.
Estes foram os pontos que decidimos delinear como nossos limites máximos à alocação.

Em seguida, decidimos que como limites mínimos à alocação tinhamos que, os colaboradores têm de comparecer a no mínimo 50% das reuniões dos seus projetos, já o líder de cada grupo tem de comparecer a todas as reuniões do seu projeto. E colocamos ainda o facto de que cada projeto tem de fazer n reuniões semanais.
Já estes, foram os pontos que optamos para os nossos limites mínimos a ter na alocação.

Limitações (que impõem limites máximos à alocação):

1. Cada sala so pode ter um projeto de cada vez

$$\forall_d \cdot \forall_t \cdot \forall_s \cdot \sum_p x_{d,t,s,p,l} \leq 1$$

2. Cada colaborador só pode ir a uma reunião de cada vez:

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

Obrigações (que impõem limites mínimos à alocação):

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

$$\forall_{d<D} \cdot \forall_{t<T} \cdot \forall_{s<S} \cdot \forall_{p<P} \cdot \quad (\sum_{c<C} x_{d,t,s,p,c} \geq (\sum_{c<C} 1 \div 2)) \vee (\sum_{c<C} x_{d,t,s,p,c} = 0)$$

4. O líder têm de aparecer em todas as reuniões do seu projeto:

$$\forall_d \cdot \forall_t \cdot \forall_s \cdot \forall_p \cdot \forall_c \cdot x_{d,t,s,p,c} <= x_{d,t,s,p,l}$$

5. Cada projeto tem de fazer n reuniões semanais:

$$\forall_{p<P} \cdot \quad \sum_{s,t,d} x_{d,t,s,p,l} = N, tal\ que\ l\ \ lder\ projeto\ p$$

[1]:
```
from z3 import *


S,T,D = 2,3,2

def construir(reunioes, colaboradores, lideres, disponibilidade):

    horario = Solver()
    P=list(reunioes) # lista que contem todos os projetos
    C=list(disponibilidade) # lista que contem todos os colaboradores
    X= {} # dicionario que relaciona um tuplo com a variavel das salas,tempos,␣
 ↪dias, projeto,colaboradores com a sua variavel logica

    for s in range (1,S+1):                 # range (1,S+1), porque assim␣
 ↪ficaremos com o output das salas correto (numeração)
```

```
        for t in range (1,T+1):
            for d in range (1,D+1):
                for p in P:
                    for c in C:
                        X[s,t,d,p,c]= Int("salas" + str(s) + "tempos" + str(t) +␣
↪"dias" + str(d) + "projetos" + str(p) + "colaboradores" + str(c))
                        if c not in colaboradores[p]:  # #se o colaborador nao␣
↪estiver no projeto coloca a 0
                            horario.add(X[s,t,d,p,c] == 0)
                        elif (t,d) not in disponibilidade[c]: #se o colaborador␣
↪nao estiver disponivel coloca a 0
                            horario.add(X[s,t,d,p,c] == 0)
                        else:
                            horario.add(X[s,t,d,p,c] >= 0,X[s,t,d,p,c] <=1)  ␣
↪#ou é 0 ou é 1

    # Cada colaborador só pode ir a uma reunião de cada vez
    for d in range(1,D+1):
        for t in range(1,T+1):
            for c in C:
                horario.add(sum([X[s,t,d,p,c] for p in P for s in␣
↪range(1,S+1)])<=1)

    # Cada projeto tem de fazer N reuniões semanais:

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

    # tem de estar presentes 50% dos colaboradores
    for s in range (1,S+1):
        for t in range (1,T+1):
            for d in range (1,D+1):
                for p in P:
                    horario.add(Or(sum([X[s,t,d,p,c] for c in␣
↪colaboradores[p]])>=(len(colaboradores[p])/2 + len(colaboradores[p])%2),
                                   (sum([X[s,t,d,p,c] for c in␣
↪colaboradores[p]])==0)))

    # Cada sala so pode ter um projeto de cada vez
    for d in range (1,D+1):
        for t in range (1,T+1):
            for s in range (1,S+1):
                horario.add(sum([X[s,t,d,p,lideres[p]] for p in P])<=1)

    #líder têm de aparecer em todas as reuniões do seu projeto:
```

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

    r=horario.check()
    if r==sat:
        m=horario.model()
        for d in range (1,D+1):
            print("Dia: "+str(d))
            for s in range (1,S+1):
                print("     Sala: "+str(s))
                for t in range (1,T+1):
                    for p in P:
                        for c in C :
                            if m[X[s,t,d,p,c]]==1:
                                print("          Tempo: "+str(t)+", Projeto:␣
 ↪"+str(p)+", Colaborador: "+str(c))

        return
    else:
        return "impossivel"
```

Exemplo 1

[2]:
```
reunioes= {1:1}    #ou seja, pretende-se que o projeto 1, tenha 1 reunião␣
 ↪semanal,...

colaboradores = {1:{1,2,3,4}}    #ou seja, no projeto1, temos como colaboradores␣
 ↪as pessoas com "id" 1,2,3,4,5,...

lideres= {1:2,2:4,3:5}     #ou seja, no projeto1, temos como líder o colaborador␣
 ↪2,...

disponibilidade= {1:[(t,d) for t in range(1,5) for d in range(1,3)]}          ␣
 ↪#ou seja, o colaborador 1, está disponível no tempo 1 dia 1

for c in range(2,):
    disponibilidade[c]=[(t,d) for t in range(1,5) for d in range(1,3)]
disponibilidade= {1:[(1,2),(1,1)],2:[(1,1),(1,2)],3:[(1,1),(1,2)],4:
 ↪[(1,1),(1,2)]}

construir(reunioes, colaboradores, lideres, disponibilidade)
```

```
sat
```

```
Dia: 1
     Sala: 1
     Sala: 2
          Tempo: 1, Projeto: 1, Colaborador: 2
          Tempo: 1, Projeto: 1, Colaborador: 4
Dia: 2
     Sala: 1
     Sala: 2
```

Exemplo 2

[3]:
```
reunioes= {1:3,2:5}    #ou seja, pretende-se que o projeto 1, tenha 3 reunião␣
 ↪semanal,...

colaboradores = {1:{1,2,3,4},2:{5,6,3,7}}    #ou seja, no projeto1, temos como␣
 ↪colaboradores as pessoas com "id" 1,2,3,4,...

lideres= {1:2,2:6}     #ou seja, no projeto1, temos como líder o colaborador 2,..
 ↪.

disponibilidade= {1:[(t,d) for t in range(1,3) for d in range(1,4)],
                  2:[(t,d) for t in range(1,4) for d in range(1,4)],
                  3:[(t,d) for t in range(1,6) for d in range(1,3)],
                  4:[(t,d) for t in range(1,6) for d in range(1,3)],
                  5:[(t,d) for t in range(1,4) for d in range(1,2)],
                  6:[(t,d) for t in range(1,5) for d in range(1,4)],
                  7:[(t,d) for t in range(1,3) for d in range(1,3)]}

construir(reunioes, colaboradores, lideres, disponibilidade)
```

```
sat
Dia: 1
     Sala: 1
          Tempo: 1, Projeto: 2, Colaborador: 6
          Tempo: 1, Projeto: 2, Colaborador: 7
          Tempo: 2, Projeto: 2, Colaborador: 3
          Tempo: 2, Projeto: 2, Colaborador: 6
     Sala: 2
          Tempo: 1, Projeto: 1, Colaborador: 1
          Tempo: 1, Projeto: 1, Colaborador: 2
          Tempo: 1, Projeto: 1, Colaborador: 3
          Tempo: 2, Projeto: 1, Colaborador: 1
          Tempo: 2, Projeto: 1, Colaborador: 2
          Tempo: 3, Projeto: 2, Colaborador: 5
          Tempo: 3, Projeto: 2, Colaborador: 6
Dia: 2
     Sala: 1
          Tempo: 1, Projeto: 2, Colaborador: 6
```

```
          Tempo: 1, Projeto: 2, Colaborador: 7
          Tempo: 2, Projeto: 2, Colaborador: 6
          Tempo: 2, Projeto: 2, Colaborador: 7
     Sala: 2
          Tempo: 2, Projeto: 1, Colaborador: 2
          Tempo: 2, Projeto: 1, Colaborador: 4
```

Exemplo 3

[4]:
```
reunioes= {1:3,2:5,3:2}    #ou seja, pretende-se que o projeto 1, tenha 3␣
 ↪reunião semanal,...

colaboradores = {1:{1,2,3,4},2:{5,2,6,7},3:{1,6,3,5}}    #ou seja, no projeto1,␣
 ↪temos como colaboradores as pessoas com "id" 1,2,3,4,...

lideres= {1:2,2:6,3:3}     #ou seja, no projeto1, temos como líder o colaborador␣
 ↪2,...

disponibilidade= {1:[(t,d) for t in range(1,3) for d in range(1,4)],
                  2:[(t,d) for t in range(1,4) for d in range(1,4)],
                  3:[(t,d) for t in range(1,6) for d in range(1,3)],
                  4:[(t,d) for t in range(1,6) for d in range(1,3)],
                  5:[(t,d) for t in range(1,4) for d in range(1,2)],
                  6:[(t,d) for t in range(1,5) for d in range(1,4)],
                  7:[(t,d) for t in range(1,3) for d in range(1,3)]}


construir(reunioes, colaboradores, lideres, disponibilidade)
```

```
sat
Dia: 1
     Sala: 1
          Tempo: 1, Projeto: 2, Colaborador: 6
          Tempo: 1, Projeto: 2, Colaborador: 7
          Tempo: 2, Projeto: 2, Colaborador: 5
          Tempo: 2, Projeto: 2, Colaborador: 6
          Tempo: 2, Projeto: 2, Colaborador: 7
          Tempo: 3, Projeto: 2, Colaborador: 2
          Tempo: 3, Projeto: 2, Colaborador: 6
     Sala: 2
          Tempo: 3, Projeto: 3, Colaborador: 3
          Tempo: 3, Projeto: 3, Colaborador: 5
Dia: 2
     Sala: 1
          Tempo: 1, Projeto: 1, Colaborador: 1
          Tempo: 1, Projeto: 1, Colaborador: 2
          Tempo: 2, Projeto: 2, Colaborador: 6
          Tempo: 2, Projeto: 2, Colaborador: 7
```

```
          Tempo: 3, Projeto: 1, Colaborador: 2
          Tempo: 3, Projeto: 1, Colaborador: 4
     Sala: 2
          Tempo: 1, Projeto: 2, Colaborador: 6
          Tempo: 1, Projeto: 2, Colaborador: 7
          Tempo: 2, Projeto: 1, Colaborador: 1
          Tempo: 2, Projeto: 1, Colaborador: 2
          Tempo: 3, Projeto: 3, Colaborador: 3
          Tempo: 3, Projeto: 3, Colaborador: 6
```

[14]:
```
reunioes= {1:2,2:3,3:3,4:2,5:2}    #ou seja, pretende-se que o projeto 1, tenha␣
 ↪3 reunião semanal,...

colaboradores = {1:{1,2,3,4},2:{2,5,6,7,8},3:{3,2,9},4:{4,6,3,7},5:{5,1,6,10}}  ␣
 ↪ #ou seja, no projeto1, temos como colaboradores as pessoas com "id" 1,2,3,4,..
 ↪.

lideres= {1:1,2:8,3:9,4:4,5:5}     #ou seja, no projeto1, temos como líder o␣
 ↪colaborador 2,...

disponibilidade= {1:[(t,d) for t in range(1,3) for d in range(1,4)],
                  2:[(t,d) for t in range(1,4) for d in range(1,4)],
                  3:[(t,d) for t in range(1,6) for d in range(1,3)],
                  4:[(t,d) for t in range(1,6) for d in range(1,3)],
                  5:[(t,d) for t in range(1,4) for d in range(1,2)],
                  6:[(t,d) for t in range(1,5) for d in range(1,4)],
                  7:[(t,d) for t in range(1,3) for d in range(1,3)],
                  8:[(t,d) for t in range(1,4) for d in range(1,2)],
                  9:[(t,d) for t in range(1,5) for d in range(1,4)],
                  10:[(t,d) for t in range(1,3) for d in range(1,3)]}


construir(reunioes, colaboradores, lideres, disponibilidade)
```

```
sat
Dia: 1
     Sala: 1
          Tempo: 1, Projeto: 5, Colaborador: 1
          Tempo: 1, Projeto: 5, Colaborador: 5
          Tempo: 1, Projeto: 5, Colaborador: 6
          Tempo: 1, Projeto: 5, Colaborador: 10
          Tempo: 2, Projeto: 5, Colaborador: 1
          Tempo: 2, Projeto: 5, Colaborador: 5
          Tempo: 3, Projeto: 3, Colaborador: 3
          Tempo: 3, Projeto: 3, Colaborador: 9
     Sala: 2
          Tempo: 1, Projeto: 2, Colaborador: 2
          Tempo: 1, Projeto: 2, Colaborador: 7
```

```
          Tempo: 1, Projeto: 2, Colaborador: 8
          Tempo: 2, Projeto: 2, Colaborador: 6
          Tempo: 2, Projeto: 2, Colaborador: 7
          Tempo: 2, Projeto: 2, Colaborador: 8
          Tempo: 3, Projeto: 2, Colaborador: 2
          Tempo: 3, Projeto: 2, Colaborador: 6
          Tempo: 3, Projeto: 2, Colaborador: 8
Dia: 2
     Sala: 1
          Tempo: 1, Projeto: 1, Colaborador: 1
          Tempo: 1, Projeto: 1, Colaborador: 2
          Tempo: 2, Projeto: 1, Colaborador: 1
          Tempo: 2, Projeto: 1, Colaborador: 3
          Tempo: 3, Projeto: 4, Colaborador: 4
          Tempo: 3, Projeto: 4, Colaborador: 6
     Sala: 2
          Tempo: 1, Projeto: 4, Colaborador: 3
          Tempo: 1, Projeto: 4, Colaborador: 4
          Tempo: 1, Projeto: 4, Colaborador: 7
          Tempo: 2, Projeto: 3, Colaborador: 2
          Tempo: 2, Projeto: 3, Colaborador: 9
          Tempo: 3, Projeto: 3, Colaborador: 3
          Tempo: 3, Projeto: 3, Colaborador: 9
```

1.1.3 Conclusão:

Achamos que acabamos por conseguir concretizar uma boa solução, ao exercício, que consegue cumprir todos os requisitos pretendidos.
Tentamos colocar como exemplos, valores de inputs diferentes e com complexidade cada vez mais complicada e, como podemos verificar conseguimos obter com êxito os resultados para esses respetivos horários.
Sem dúvida, que não foi um trabalho fácil, sentimos dificuldade em certas funções e sobretudo para conseguir compreender quais as fórmulas que teriamos de usar e ter em conta, para o cumprimento de cada uma das Limitações e das Obrigações.
O exercício realizado na aula prática, também relativamente à criação de horários. Obviamente, este exercício tinha um grau de dificuldade muito mais elevado e com condições muito mais trabalhosas.
Apesar de tudo isso, conseguimos, a nosso ver, uma solução a este exercício muito positiva e o qual achamos super interessante e muito útil em diversas áreas e em diferentes ocasiões!