Trabalho Prático 1

Bruno Jardim (a91680) José Ferreira (a91636)

1- Horário Semanal de Reuniões de uma "StartUp"

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

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).

Análise do Problema

Existem C colaboradores, estes colaboradores estão numerados de $[0..C-1]$, a este valores atribuiremos a variável c, estes colaboradores estão associados a um determinado projeto $p \in [0..P-1]$.

Mais ainda, podemos identificar uma sala disponível, num determinado dia, num certo intervalo como sendo um triplo $(s,d,i) \in [0..S-1] \times [0..D-1] \times [0..I-1]$, sendo o s,d,i, a sala, o dia e o intervalo, respetivamente.

Para solucionar o problema iremos utilizar o quíntuplo $Q_{c,p,s,d,i}$, as variáveis tomarão valores binários $(0,1)$

$$Q_{c,p,s,d,i} = 1 \quad \text{se e só se} \\ \text{o colaborador } c \text{ está na reunião do projeto } p \text{ na sala } s \text{ no dia } d \text{ e no intervalo } i$$

Para efeitos contabilísticos, no que toca ao número de reuniões, consideraremos o quádruplo $R_{p,s,d,i}$, as variáveis tomarão valores binários $(0,1)$

$$R_{p,s,d,i} = 1 \quad \text{se e só se} \quad \text{existe uma reunião do projeto } p \text{ na sala } s \text{ no dia } d \text{ e no intervalo } i$$

Para conseguirmos representar os colaboradores e os projetos em que eles estão, consideraremos o tuplo $CP_{c,p}$, as variáveis tomarão valores binários $(0,1)$

$$CP_{c,p} = 1 \quad \text{se e só se} \quad \text{o colaborador } c \text{ pertence ao projeto } p$$

Dados a passar ao programa:\ **C**-> Matriz em que cada linha simboliza os projetos que um colaborador está envolvido\ **N**-> Lista dos números de reuniões a efetuar (i.e.: n[1]=7 indica que o projeto 7 terá 7 reuniões) \ **L**-> Matriz em que cada linha simboliza os projetos e o líder do mesmo\ **S**-> O número de salas\ **D**-> O número de dias em que se efetuaram reuniões\ **T** -> O número de tempos em que é possível efetuar uma reunião em cada dia\ **disp**-> A disponibilidade dos colaboradores em forma de dicionário, as keys serão o número do colaborador, os values serão um dicionário com as keys sendo a sala e o tempo e os values uma lista de valores binários com comprimento igual ao número de dias

In [1]:
```
import sys
sys.path.insert(1,r'C:\Users\Bruno\OneDrive\Ambiente de Trabalho\LC')
import numpy as np
from pyscipopt import Model,quicksum
from ipynb.fs.full.PrintSourceCode import printCode,printSC,printC
```

In [2]:
```
from pyscipopt import Model,quicksum
def criahorario(CD, N, LD, S, T, D, disp):
    P = len(LD[0])
    C = len(LD)

    horario = Model()

    # Criação de variaveis
    Q = {}
    for c in range(C):
        Q[c] = {}
        for p in range(P):
            Q[c][p] = {}
            for s in range(S):
                Q[c][p][s] = {}
                for d in range(D):
                    Q[c][p][s][d] = {}
                    for t in range(T):
                        Q[c][p][s][d][t] = horario.addVar(str(c) + '_' + \
                                                          str(p) + '_' + str(s) +
                                                          vtype='B')

    R = {}
    for p in range(P):
        R[p] = {}
        for s in range(S):
            R[p][s] = {}
            for d in range(D):
                R[p][s][d] = {}
                for t in range(T):
                    R[p][s][d][t] = horario.addVar(str(p) + '_' + str(s) + '_' + str(

    ### Restrições para as reuniões

    # Para cada slot temos no máximo uma reunião LIM
    for s in range(S):
        for d in range(D):
            for t in range(T):
                horario.addCons(quicksum([R[p][s][d][t] for p in range(P)]) <= 1)

    # Cada projeto realiza um N número de reuniões OBR
    for p in range(P):
        horario.addCons(N[p] == quicksum([R[p][s][d][t] for s in range(S) \
                                               for d in range(D) for t in range(T)]))

    ### Restrição para os colaboradores

    # Um colaborador n pode estar em uma sala que n tenha reuniao marcada
    for d in range(D):
        for s in range(S):
            for t in range(T):
                for p in range(P):
                    for c in range(C):
                        horario.addCons(Q[c][p][s][d][t] <= R[p][s][d][t])

    # Cada reunião tem no minimo 50% dos colaboradores
    for p in range(P):
        for s in range(S):
            for d in range(D):
                for t in range(T):
                    horario.addCons(quicksum([Q[c][p][s][d][t] for c in range(C)]) >=
                                    (((quicksum([CD[c_][p] for c_ in range(C)])) * R[

    # Um colaborador não pode frequentar duas reuniões em simultâneo LIM
    for c in range(C):
        for d in range(D):
            for t in range(T):
                horario.addCons(quicksum([Q[c][p][s][d][t] for s in range(S) for p in

    # O líder tem de esstar presente em todas as reuniões do seu projeto
    for c in range(C):
        for p in range(P):
            for s in range(S):
                for d in range(D):
                    for t in range(T):
                        horario.addCons(Q[c][p][s][d][t] >= (R[p][s][d][t] * LD[c][p]

    # Obriga a um colaborador a pertencer ao projeto consoante a matriz de input
    for c in range(C):
        for p in range(P):
            for s in range(S):
                for d in range(D):
                    for t in range(T):
                        horario.addCons(Q[c][p][s][d][t] <= CD[c][p])

    # verificar que o colaborador tem disponibilidade
    for c in range(C):
        for p in range(P):
            for s in range(S):
                for d in range(D):
                    for t in range(T):
                        key = 'Sala ' + str(s) + ', Tempo ' + str(t)
                        horario.addCons(Q[c][p][s][d][t] <= disp[c][key][d])

    horario.optimize()
    if horario.getStatus() == 'optimal':
        printC(disp, D)
        printSC(horario, C, P, S, D, T, Q, R)
        return True
    else:
        return False
```

In [8]:
```
from random import choices

S,T,D = 2,2,6

N = [8,7]

L = [[1,0],
     [0,1],
     [0,0],
     [0,0]]

C = [[1,0],
     [1,1],
     [1,1],
     [0,0]]

disponibilidade = {}

d = [0, 1]
prob = [0.3, 0.5]

for c in range( len(C) ):
    disponibilidade[c] = {}
    for sala in range(S):
        for tempo in range(T):
            key = 'Sala '+str(sala)+', Tempo '+str(tempo)
            disponibilidade[c][key] = choices(d, weights=prob, k=D)


while not criahorario(C,N,L,S,T,D,disponibilidade):
    a =1
```

```
Disponibilidade colaborador 0
                         Dia 0          Dia 1          Dia 2          Dia 3          D
ia 4          Dia 5
Sala 0, Tempo 0              0              0              1              1
   1              1
Sala 0, Tempo 1              1              0              1              1
   0              0
Sala 1, Tempo 0              1              1              0              1
   1              1
Sala 1, Tempo 1              0              0              0              1
   1              1

Disponibilidade colaborador 1
                         Dia 0          Dia 1          Dia 2          Dia 3          D
ia 4          Dia 5
Sala 0, Tempo 0              0              1              1              1
   1              1
Sala 0, Tempo 1              1              0              0              0
   1              1
Sala 1, Tempo 0              1              1              0              0
   1              1
Sala 1, Tempo 1              1              1              0              1
   1              1

Disponibilidade colaborador 2
                         Dia 0          Dia 1          Dia 2          Dia 3          D
ia 4          Dia 5
Sala 0, Tempo 0              1              1              1              1
   1              0
Sala 0, Tempo 1              1              1              1              0
   0              0
Sala 1, Tempo 0              1              1              1              1
   1              1
Sala 1, Tempo 1              1              0              1              1
   1              1

Disponibilidade colaborador 3
                         Dia 0          Dia 1          Dia 2          Dia 3          D
ia 4          Dia 5
Sala 0, Tempo 0              0              0              1              1
   1              1
Sala 0, Tempo 1              1              1              0              1
   0              0
Sala 1, Tempo 0              0              0              1              1
   1              0
Sala 1, Tempo 1              0              1              1              1
   0              1

Reuniões
                         Dia 0          Dia 1          Dia 2          Dia 3          D
ia 4          Dia 5
Sala 0, Tempo 0              -             P1             P1             P0
  P0             P0
Sala 0, Tempo 1             P0              -             P0              -
  P1             P1
Sala 1, Tempo 0             P1             P0              -              -
   -              -
Sala 1, Tempo 1             P1              -              -             P1
  P0             P0

Horário
                         Dia 0          Dia 1          Dia 2          Dia 3          D
ia 4          Dia 5
Sala 0, Tempo 0              -           P1:1           P1:1       P0:0,1,2       P0:
0,1,2         P0:0,1
Sala 0, Tempo 1         P0:0,2              -         P0:0,2              -
P1:1           P1:1
Sala 1, Tempo 0           P1:1         P0:0,2              -              -
   -              -
Sala 1, Tempo 1           P1:1              -              -           P1:1          P
0:0,2         P0:0,2
```

Disponibilidade colaborador 0 Dia 0 Dia 1 Dia 2 Dia 3 Dia 4 Dia 5 Sala 0, Tempo 0 0 1 0 1 0 0 Sala 0, Tempo 1 1 0 0 1 1 0 Sala 1, Tempo 0 1 1 0 1 1 0 Sala 1, Tempo 1 1 1 1 1 1 1

Disponibilidade colaborador 1 Dia 0 Dia 1 Dia 2 Dia 3 Dia 4 Dia 5 Sala 0, Tempo 0 1 0 1 1 0 1 Sala 0, Tempo 1 1 1 1 1 1 1 Sala 1, Tempo 0 0 0 1 1 1 1 Sala 1, Tempo 1 0 0 0 1 1 1

Disponibilidade colaborador 2 Dia 0 Dia 1 Dia 2 Dia 3 Dia 4 Dia 5 Sala 0, Tempo 0 1 1 1 1 1 0 Sala 0, Tempo 1 1 0 1 1 1 1 Sala 1, Tempo 0 1 0 0 0 1 0 Sala 1, Tempo 1 1 1 0 1 1 1

Disponibilidade colaborador 3 Dia 0 Dia 1 Dia 2 Dia 3 Dia 4 Dia 5 Sala 0, Tempo 0 1 0 1 1 1 1 Sala 0, Tempo 1 1 1 1 1 0 0 Sala 1, Tempo 0 1 1 1 0 0 1 Sala 1, Tempo 1 1 0 1 1 1 1

Reuniões Dia 0 Dia 1 Dia 2 Dia 3 Dia 4 Dia 5 Sala 0, Tempo 0 P1 P0 P1 P1 - P1 Sala 0, Tempo 1 P0 P1 P1 P0 P0 P1 Sala 1, Tempo 0 P0 - - - P0 - Sala 1, Tempo 1 - P0 - - - P0

Horário Dia 0 Dia 1 Dia 2 Dia 3 Dia 4 Dia 5 Sala 0, Tempo 0 P1:1 P0:0,2 P1:1,2 P1:1 - P1:1 Sala 0, Tempo 1 P0:0,1,2 P1:1 P1:1,2 P0:0,1,2 P0:0,1,2 P1:1 Sala 1, Tempo 0 P0:0,2 - - - P0:0,2 - Sala 1, Tempo 1 - P0:0,2 - - - P0:0,2

In [7]:
```
import numpy as np


def initRQ(S,T,D):
    # [Sala X,Tempo Y]:['-','-','-','-']
    #  [0-S]    [0-T]           D
    R = {}

    for s in range(S):
        for t in range(T):
            key = 'Sala '+str(s)+', Tempo '+str(t)
            R[key]=[]
            for d in range(D):
                R[key].append('-')
    return R
def printRQ(R,D):

    dias = [ 'Dia '+str(n) for n in range(D) ]

    row_format ="{:>15}" * (D + 1)
    print(row_format.format("", *dias))

    for key in R:
        print(row_format.format(key, *R[key]))

def printC(R,D):
    dias = [ 'Dia '+str(n) for n in range(D) ]

    for c in R:
        print('\nDisponibilidade colaborador '+str(c))


        row_format ="{:>15}" * (D + 1)
        print(row_format.format("", *dias))

        for key in R[c]:
            print(row_format.format(key, *R[c][key]))





def printSC(m,C,P,S,D,T,Q,R):



    Q_ = initRQ(S,T,D) #['Sala X, Tempo Y'] : ['-','-',...,'-']
    R_ = initRQ(S,T,D) #['Sala X, Tempo Y'] : ['-','-',...,'-']


    s_t = S*T
    matX = [ ['-' for col in range(D)] for lin in range(s_t) ]

    # colher os dados das reunioes
    for p in range(P):
        for s in range(S):
            for d in range(D):
                for t in range(T):
                    if m.getVal(R[p][s][d][t]):
                        R_['Sala '+str(s)+', Tempo '+str(t)][d] = 'P'+str(p)


    for c in range(C):
        for p in range(P):
            for s in range(S):
                for d in range(D):
                    for t in range(T):
                        if m.getVal(Q[c][p][s][d][t]):
                            keyQ = 'Sala '+str(s)+', Tempo '+str(t)

                            if Q_[keyQ][d][0] == 'P':
                                if Q_[keyQ][d][1] != str(p):
                                    print('Erro estão a decorrer duas reunioes ao
                                           str(s)+' tempo '+str(t)+'(P:'+Q_
                                Q_[keyQ][d] = Q_[keyQ][d]+','+str(c)
                            else:
                                Q_[keyQ][d] = 'P'+str(p)+':'+str(c)

    print('\nReuniões')
    printRQ(R_,D)
    print('\nHorário')
    printRQ(Q_,D)
```

In [ ]: