

Lógica Computacional

João Carlos Viana Pereira Marques A84684 Saimon Alves Costa Sousa A76575

Trabalho Prático 1

Exercício 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 $1 \dots S$,) durante um “slot” (tempo, dia). Assume-se os dias enumerados $1 \dots D$ e, em cada dia, os tempos enumerados $1 \dots T$.
2. Cada reunião tem associado um projeto (enumerados $1 \dots P$) e um conjunto de participantes. Os diferentes colaboradores são enumerados $1 \dots C$.
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).

Análise do Problema

Este é um problema de alocação. Teremos a variável S, D, T, P e C , que representam respetivamente, o número das salas disponíveis na StartUp, o número de dias de trabalho, o número de tempos por dia, o número do projeto e o número de colaboradores.

Utilizando as variáveis definidas em cima podemos criar uma matriz de alocação com o seguinte significado:

$X_{p,c,s,d,t} = 1$ se e só se o projeto p é alocado ao colaborador c na sala s no dia d no tempo t

Agora vamos enumerar as limitações que achamos pertinentes para a resolução do problema:

1. Cada sala tem no máximo 1 projeto de cada vez

$$\sum_{p < P} X_{p,c,s,d,t} \leq 1$$

2. Cada projeto tem no máximo uma sala de cada vez

$$\sum_{s < S} X_{p,c,s,d,t} \leq 1$$

3. Cada colaborador só pode estar no máximo num projeto em cada tempo dia

$$\sum_{p < P} X_{p,c,s,d,t} \leq 1$$

Implementões obrigatórias:

1. Cada projeto tem de realizar um N número de reuniões semanais

$$\sum_{s < S, d < D, t < T} X_{p,c,s,d,t} = N$$

2. O líder tem de participar em todas as reuniões do seu projeto

$$\sum_{s < S, t < T, d < D} X_{p,l,s,d,t} = N$$

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

```
def horario (S,D,T,P,C):
```

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

```
    x = {}
```

```
    for s in range (S):
```

```
        for d in range (D):
```

```
            for t in range (T):
```

```
                for p in range (P):
```

```
                    x[s,d,t,p] = horario.BoolVar("x[%i,%i,%i,%i]"
```

```
% (s,d,t,p)
```

```
#cada sala tem 1 projeto de cada vez
```

```
    for s in range(S):
```

```
        for d in range(D):
```

```
            for t in range(T):
```

```
                horario.Add(sum([x[p,c,s,d,t] for p in range(P)]) <=
```

```
1)
```

```
#cada colaborador tem uma reunião em cada (T,D)
```

```
    for c in range(C):
```

```
        for s in range(S):
```

```
            for d in range(D):
```

```
                for t in range(T):
```

```
                    horario.Add(sum([x[p,c,s,d,t] for p in range(P)])
```

```
<= 1)
```

Exercício 2

Queremos construir um programa que inicialize um tabuleiro sudoku $N^2 \times N^2$, para $N \in \{3,4,5,6\}$, e que preencha uma fração α , para $\alpha \in \{0.0,0.2,0.4,0.6\}$ das N^4 casas respeitando a regra:

- Cada inteiro no intervalo de 1 a N^2 ocorre uma e uma só vez em cada linha, coluna ou secção

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

```

#declarar o solver
solver = pywraplp.Solver.CreateSolver('SCIP')

def sudoku(n,a):
    n_=n**2
    numcasas=round(n**4 * a)

    #definir casas a serem preenchidas
    A=[]
    for i in range(numcasas):
        A.append((random.randint(0,n_-1), random.randint(0,n_-1)))

    #criar variáveis
    x={}
    rows=range(0,n_)
    cols=range(0,n_)
    val=range(1,n_+1)
    for r in rows:
        for c in cols:
            for v in val:
                x[r,c,v] = solver.BoolVar("x[%i,%i,%i]" % (r,c,v))

    #restrição para que em cada célula exista apenas um número de 1 a
N^2
    for r in rows:
        for c in cols:
            solver.Add(sum(x[r,c,v] for v in val) == 1)

    #restrição para cada linha conter todos os números de 1 a N^2
    for r in rows:
        for v in val:
            solver.Add(sum(x[r,c,v] for c in cols) == 1)

    #restrição para cada coluna conter todos os números de 1 a N^2
    for c in rows:
        for v in val:
            solver.Add(sum(x[r,c,v] for r in rows) == 1)

    #restrição para aplicar as regras anteriores a cada sub-tabuleiro
NxN
    a = []
    count = n
    while(count<n_):
        a.append(count)
        count+=n
    for i in a:
        for k in a:
            for v in val:

```

```

        solver.Add(sum(x[r+i,c+k,v] for r in range(0,n) for c
in range(0,n)) == 1)

```

#imprimir grelha com uma fração das casas preenchidas

```
status=solver.Solve()
```

```
if status == pywraplp.Solver.OPTIMAL:
```

```
print(n_, ' x ', n_)
```

```
for r in rows :
```

```
print(end='\n')
```

```
for c in cols :
```

```
for v in val:
```

```
if round(x[r,c,v].solution_value()) == 1 and (r,c)
```

in A:

```
print('[', v, ']', end=' ')
```

```
elif round(x[r,c,v].solution.value()==1) and
```

(r, c) not in A :

```
print(' [ ', ' ]', end=' ')
```

```
print( '\n' )
```

```
#imprimir grelha preenchida na totalidade
```

```
status=solver.Solve()
```

```
if status == pywraplp.Solver.OPTIMAL:
```

```
for r in rows :
```

```
print(end='\n')
```

```
for c in cols :
```

```
for v in val:
```

```
if round(x[r,c,v].solution.value()) == 1:
```

```
print('[', v, ']', end=' ')
```

```
print( '\n\n' )
```

```
sudoku(3,0.0)
```

sudoku(3, 0.2)

sudoku(3, 0.4)

sudoku (3, 0.6)

9 x 9

[illegible]

[2]	[5]	[8]	[3]	[6]	[1]	[7]	[9]	[4]
[7]	[3]	[6]	[5]	[9]	[4]	[2]	[8]	[1]
[4]	[1]	[9]	[2]	[8]	[7]	[5]	[6]	[3]
[6]	[4]	[3]	[9]	[2]	[8]	[1]	[5]	[7]
[9]	[7]	[2]	[1]	[4]	[5]	[6]	[3]	[8]
[5]	[8]	[1]	[7]	[3]	[6]	[4]	[2]	[9]
[1]	[2]	[7]	[8]	[5]	[3]	[9]	[4]	[6]
[8]	[9]	[4]	[6]	[1]	[2]	[3]	[7]	[5]
[3]	[6]	[5]	[4]	[7]	[9]	[8]	[1]	[2]

9 x 9

[5]	[8]	[]	[]	[4]	[]	[]	[]	[]
[]	[]	[]	[8]	[]	[3]	[]	[]	[]
[]	[]	[]	[]	[7]	[]	[]	[]	[8]
[2]	[]	[]	[]	[]	[]	[]	[]	[]
[]	[]	[]	[]	[]	[]	[2]	[]	[4]
[]	[4]	[9]	[]	[]	[]	[]	[]	[]
[]	[]	[]	[]	[]	[]	[]	[]	[]
[]	[]	[]	[]	[]	[]	[]	[]	[1]

[5]	[8]	[7]	[6]	[4]	[9]	[1]	[2]	[3]
[9]	[2]	[1]	[8]	[5]	[3]	[6]	[4]	[7]
[4]	[3]	[6]	[2]	[7]	[1]	[9]	[5]	[8]
[3]	[9]	[4]	[7]	[1]	[2]	[8]	[6]	[5]
[2]	[7]	[8]	[4]	[6]	[5]	[3]	[1]	[9]
[6]	[1]	[5]	[3]	[9]	[8]	[2]	[7]	[4]
[7]	[4]	[9]	[1]	[8]	[6]	[5]	[3]	[2]
[1]	[5]	[3]	[9]	[2]	[7]	[4]	[8]	[6]
[8]	[6]	[2]	[5]	[3]	[4]	[7]	[9]	[1]

9 x 9

[]	[3]	[]	[]	[]	[]	[]	[]	[]
[]	[]	[6]	[]	[]	[2]	[]	[]	[]
[]	[]	[]	[1]	[]	[]	[]	[]	[]
[9]	[6]	[]	[]	[]	[]	[]	[3]	[7]
[1]	[8]	[3]	[]	[5]	[]	[9]	[4]	[2]
[]	[]	[]	[]	[]	[]	[5]	[]	[]
[]	[4]	[]	[]	[]	[]	[2]	[]	[]
[2]	[9]	[8]	[4]	[]	[]	[6]	[]	[]
[]	[5]	[7]	[]	[]	[]	[3]	[]	[4]

[4]	[3]	[2]	[9]	[7]	[6]	[8]	[5]	[1]
-------	-------	-------	-------	-------	-------	-------	-------	-------

[8]	[1]	[6]	[5]	[4]	[2]	[7]	[9]	[3]
[5]	[7]	[9]	[1]	[3]	[8]	[4]	[2]	[6]
[9]	[6]	[5]	[8]	[2]	[4]	[1]	[3]	[7]
[1]	[8]	[3]	[6]	[5]	[7]	[9]	[4]	[2]
[7]	[2]	[4]	[3]	[9]	[1]	[5]	[6]	[8]
[3]	[4]	[1]	[7]	[6]	[5]	[2]	[8]	[9]
[2]	[9]	[8]	[4]	[1]	[3]	[6]	[7]	[5]
[6]	[5]	[7]	[2]	[8]	[9]	[3]	[1]	[4]

9 x 9

[[] [7] [1] [] [9] [3] [] [] [6] [
[5] [] [] [8] [7] [6] [2] [9] [] [1] [
[3] [] [] [6] [] [] [] [2] [] [7] [
[7] [] [2] [] [] [] [4] [1] [6] [] [
[] [] [8] [] [] [] [6] [] [2] [] [
[] [] [6] [9] [2] [] [1] [8] [7] [5] [
[] [] [] [4] [6] [1] [] [3] [] [] [
[9] [] [1] [] [] [2] [] [] [5] [] [
[] [] [] [2] [5] [] [] [] [] [] [

[2]	[7]	[1]	[4]	[9]	[3]	[5]	[8]	[6]
[5]	[4]	[8]	[7]	[6]	[2]	[9]	[3]	[1]
[3]	[9]	[6]	[1]	[8]	[5]	[2]	[4]	[7]
[7]	[2]	[3]	[8]	[5]	[4]	[1]	[6]	[9]
[1]	[8]	[5]	[9]	[7]	[6]	[4]	[2]	[3]
[4]	[6]	[9]	[2]	[3]	[1]	[8]	[7]	[5]
[8]	[5]	[4]	[6]	[1]	[7]	[3]	[9]	[2]
[9]	[1]	[7]	[3]	[2]	[8]	[6]	[5]	[4]
[6]	[3]	[2]	[5]	[4]	[9]	[7]	[1]	[8]

```
sudoku(4,0.0)
```

16 x 16

[illegible]

```
[ 1 ] [ 10 ] [ 9 ] [ 6 ] [ 4 ] [ 7 ] [ 11 ] [ 5 ] [ 14 ] [ 13 ] [ 16 ]
[ 12 ] [ 3 ] [ 8 ] [ 2 ] [ 15 ]
[ 5 ] [ 12 ] [ 11 ] [ 2 ] [ 3 ] [ 6 ] [ 16 ] [ 9 ] [ 15 ] [ 8 ] [ 7 ]
[ 10 ] [ 1 ] [ 14 ] [ 4 ] [ 13 ]
[ 14 ] [ 13 ] [ 15 ] [ 7 ] [ 10 ] [ 12 ] [ 1 ] [ 8 ] [ 2 ] [ 5 ] [ 4 ]
[ 3 ] [ 11 ] [ 16 ] [ 6 ] [ 9 ]
[ 8 ] [ 4 ] [ 3 ] [ 16 ] [ 15 ] [ 2 ] [ 14 ] [ 13 ] [ 1 ] [ 9 ] [ 6 ]
[ 11 ] [ 7 ] [ 12 ] [ 5 ] [ 10 ]
[ 11 ] [ 9 ] [ 5 ] [ 14 ] [ 12 ] [ 15 ] [ 7 ] [ 16 ] [ 13 ] [ 4 ]
[ 3 ] [ 6 ] [ 8 ] [ 10 ] [ 1 ] [ 2 ]
[ 10 ] [ 3 ] [ 2 ] [ 12 ] [ 13 ] [ 4 ] [ 6 ] [ 11 ] [ 8 ] [ 1 ] [ 5 ]
[ 16 ] [ 14 ] [ 9 ] [ 15 ] [ 7 ]
[ 16 ] [ 7 ] [ 4 ] [ 13 ] [ 1 ] [ 9 ] [ 8 ] [ 10 ] [ 11 ] [ 14 ]
[ 15 ] [ 2 ] [ 5 ] [ 6 ] [ 12 ] [ 3 ]
[ 6 ] [ 1 ] [ 8 ] [ 15 ] [ 14 ] [ 5 ] [ 2 ] [ 3 ] [ 10 ] [ 7 ] [ 12 ]
[ 9 ] [ 4 ] [ 11 ] [ 13 ] [ 16 ]
[ 15 ] [ 11 ] [ 14 ] [ 9 ] [ 16 ] [ 1 ] [ 12 ] [ 4 ] [ 3 ] [ 2 ] [ 8 ]
[ 5 ] [ 13 ] [ 7 ] [ 10 ] [ 6 ]
[ 7 ] [ 16 ] [ 12 ] [ 1 ] [ 6 ] [ 8 ] [ 5 ] [ 2 ] [ 9 ] [ 11 ] [ 10 ]
[ 13 ] [ 15 ] [ 4 ] [ 3 ] [ 14 ]
[ 3 ] [ 6 ] [ 10 ] [ 8 ] [ 11 ] [ 13 ] [ 15 ] [ 14 ] [ 7 ] [ 12 ]
[ 1 ] [ 4 ] [ 16 ] [ 2 ] [ 9 ] [ 5 ]
[ 4 ] [ 2 ] [ 13 ] [ 5 ] [ 9 ] [ 3 ] [ 10 ] [ 7 ] [ 6 ] [ 16 ] [ 14 ]
[ 15 ] [ 12 ] [ 1 ] [ 8 ] [ 11 ]
[ 9 ] [ 8 ] [ 16 ] [ 11 ] [ 7 ] [ 10 ] [ 4 ] [ 6 ] [ 5 ] [ 3 ] [ 13 ]
[ 1 ] [ 2 ] [ 15 ] [ 14 ] [ 12 ]
[ 13 ] [ 15 ] [ 1 ] [ 10 ] [ 2 ] [ 14 ] [ 3 ] [ 12 ] [ 4 ] [ 6 ]
```

```

[ 11 ] [ 7 ] [ 9 ] [ 5 ] [ 16 ] [ 8 ]
[ 2 ] [ 14 ] [ 7 ] [ 4 ] [ 5 ] [ 16 ] [ 13 ] [ 15 ] [ 12 ] [ 10 ]
[ 9 ] [ 8 ] [ 6 ] [ 3 ] [ 11 ] [ 1 ]
[ 12 ] [ 5 ] [ 6 ] [ 3 ] [ 8 ] [ 11 ] [ 9 ] [ 1 ] [ 16 ] [ 15 ] [ 2 ]
[ 14 ] [ 10 ] [ 13 ] [ 7 ] [ 4 ]

```

sudoku(4,0.2)

16 x 16

```

[   ] [   ] [   ] [ 11 ] [ 16 ] [   ] [ 14 ] [   ] [   ] [   ] [   ] [
] [   ] [   ] [   ] [   ]
[   ] [   ] [   ] [   ] [   ] [   ] [   ] [ 2 ] [   ] [ 13 ] [ 12 ] [
] [   ] [ 16 ] [   ] [   ]
[ 7 ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [ 10 ] [
] [   ] [   ] [   ] [ 9 ]
[   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [ 6 ]
[   ] [   ] [   ] [ 1 ] [   ]
[   ] [   ] [   ] [   ] [ 2 ] [   ] [   ] [   ] [   ] [   ] [   ]
[   ] [   ] [   ] [   ] [   ]
[   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [ 14 ] [   ]
[ 10 ] [ 13 ] [   ] [   ] [ 1 ]
[   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ]
[   ] [ 15 ] [   ] [   ] [ 2 ]
[   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ]
[   ] [   ] [   ] [ 10 ] [   ]
[   ] [   ] [   ] [   ] [   ] [   ] [   ] [ 16 ] [   ] [   ] [   ]
[   ] [   ] [   ] [   ] [   ]
[   ] [   ] [ 12 ] [ 7 ] [   ] [   ] [   ] [ 5 ] [   ] [ 9 ] [   ] [
] [   ] [   ] [   ] [   ]
[   ] [   ] [ 13 ] [   ] [   ] [   ] [   ] [ 7 ] [   ] [   ] [   ] [ 3
] [   ] [   ] [   ] [ 16 ]
[   ] [   ] [   ] [   ] [ 13 ] [ 9 ] [   ] [   ] [   ] [   ] [ 16 ] [
] [ 10 ] [   ] [   ] [   ]
[   ] [   ] [   ] [ 6 ] [   ] [   ] [   ] [   ] [   ] [   ] [   ]
[   ] [ 16 ] [ 7 ] [   ] [   ]
[   ] [ 15 ] [ 11 ] [ 10 ] [   ] [   ] [   ] [ 4 ] [   ] [   ] [   ] [
] [   ] [   ] [ 2 ] [   ]
[   ] [ 7 ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [ 4 ] [   ]
[   ] [   ] [   ] [   ] [   ]
[ 9 ] [   ] [   ] [   ] [   ] [   ] [   ] [ 8 ] [   ] [   ] [   ]
[   ] [ 1 ] [ 5 ] [   ] [   ]

```

```

[ 13 ] [ 5 ] [ 15 ] [ 11 ] [ 16 ] [ 4 ] [ 14 ] [ 10 ] [ 9 ] [ 8 ]
[ 2 ] [ 1 ] [ 7 ] [ 6 ] [ 12 ] [ 3 ]
[ 6 ] [ 3 ] [ 9 ] [ 4 ] [ 1 ] [ 7 ] [ 15 ] [ 2 ] [ 14 ] [ 13 ] [ 12 ]
[ 11 ] [ 8 ] [ 16 ] [ 5 ] [ 10 ]

```



```

[ 7 ] [ 1 ] [ 14 ] [ 2 ] [ 12 ] [ 3 ] [ 8 ] [ 6 ] [ 4 ] [ 16 ] [ 10 ]
[ 5 ] [ 11 ] [ 15 ] [ 13 ] [ 9 ]
[ 16 ] [ 12 ] [ 10 ] [ 8 ] [ 9 ] [ 13 ] [ 5 ] [ 11 ] [ 7 ] [ 3 ] [ 6 ]
[ 15 ] [ 2 ] [ 4 ] [ 1 ] [ 14 ]
[ 5 ] [ 11 ] [ 16 ] [ 12 ] [ 2 ] [ 14 ] [ 10 ] [ 1 ] [ 3 ] [ 15 ]
[ 8 ] [ 13 ] [ 4 ] [ 9 ] [ 6 ] [ 7 ]
[ 2 ] [ 6 ] [ 3 ] [ 15 ] [ 11 ] [ 5 ] [ 4 ] [ 9 ] [ 12 ] [ 14 ] [ 7 ]
[ 10 ] [ 13 ] [ 8 ] [ 16 ] [ 1 ]
[ 10 ] [ 9 ] [ 7 ] [ 13 ] [ 6 ] [ 8 ] [ 12 ] [ 3 ] [ 16 ] [ 1 ] [ 5 ]
[ 4 ] [ 15 ] [ 11 ] [ 14 ] [ 2 ]
[ 4 ] [ 14 ] [ 8 ] [ 1 ] [ 7 ] [ 15 ] [ 16 ] [ 13 ] [ 11 ] [ 6 ] [ 9 ]
[ 2 ] [ 12 ] [ 3 ] [ 10 ] [ 5 ]
[ 1 ] [ 10 ] [ 5 ] [ 9 ] [ 4 ] [ 12 ] [ 2 ] [ 16 ] [ 8 ] [ 11 ] [ 14 ]
[ 7 ] [ 6 ] [ 13 ] [ 3 ] [ 15 ]
[ 8 ] [ 16 ] [ 12 ] [ 7 ] [ 15 ] [ 10 ] [ 3 ] [ 5 ] [ 1 ] [ 9 ] [ 13 ]
[ 6 ] [ 14 ] [ 2 ] [ 11 ] [ 4 ]
[ 11 ] [ 2 ] [ 13 ] [ 14 ] [ 8 ] [ 1 ] [ 6 ] [ 7 ] [ 15 ] [ 10 ] [ 4 ]
[ 3 ] [ 5 ] [ 12 ] [ 9 ] [ 16 ]
[ 15 ] [ 4 ] [ 6 ] [ 3 ] [ 13 ] [ 9 ] [ 11 ] [ 14 ] [ 2 ] [ 5 ] [ 16 ]
[ 12 ] [ 10 ] [ 1 ] [ 7 ] [ 8 ]
[ 14 ] [ 8 ] [ 2 ] [ 6 ] [ 10 ] [ 11 ] [ 1 ] [ 15 ] [ 5 ] [ 12 ] [ 3 ]
[ 9 ] [ 16 ] [ 7 ] [ 4 ] [ 13 ]
[ 12 ] [ 15 ] [ 11 ] [ 10 ] [ 5 ] [ 16 ] [ 9 ] [ 4 ] [ 13 ] [ 7 ]
[ 1 ] [ 8 ] [ 3 ] [ 14 ] [ 2 ] [ 6 ]
[ 3 ] [ 7 ] [ 1 ] [ 5 ] [ 14 ] [ 2 ] [ 13 ] [ 12 ] [ 6 ] [ 4 ] [ 15 ]
[ 16 ] [ 9 ] [ 10 ] [ 8 ] [ 11 ]
[ 9 ] [ 13 ] [ 4 ] [ 16 ] [ 3 ] [ 6 ] [ 7 ] [ 8 ] [ 10 ] [ 2 ] [ 11 ]
[ 14 ] [ 1 ] [ 5 ] [ 15 ] [ 12 ]

```

sudoku(4,0.4)

16 x 16

```

[   ] [   ] [ 10 ] [   ] [ 8 ] [   ] [ 6 ] [   ] [   ] [   ] [   ]
[ 12 ] [   ] [   ] [   ] [   ]
[   ] [ 9 ] [ 11 ] [ 8 ] [   ] [ 7 ] [   ] [   ] [   ] [   ] [   ]
[   ] [   ] [   ] [   ]
[ 14 ] [   ] [ 6 ] [   ] [ 16 ] [   ] [ 10 ] [   ] [   ] [   ] [   ]
[   ] [   ] [   ] [   ] [ 1 ]
[   ] [ 2 ] [   ] [   ] [   ] [   ] [   ] [ 3 ] [   ] [ 13 ] [   ]
[   ] [   ] [ 9 ] [ 8 ] [   ]
[ 16 ] [   ] [ 13 ] [ 5 ] [ 2 ] [   ] [   ] [   ] [   ] [   ] [ 1 ]
[   ] [   ] [   ] [   ]
[   ] [   ] [ 12 ] [   ] [   ] [ 1 ] [   ] [ 13 ] [   ] [ 16 ] [   ]
9 ] [ 15 ] [   ] [ 2 ] [   ]
[   ] [   ] [ 8 ] [   ] [   ] [   ] [   ] [   ] [ 11 ] [   ] [   ] [ 2 ]
[ 9 ] [ 16 ] [   ] [ 3 ]
[   ] [   ] [   ] [   ] [   ] [   ] [   ] [   ] [ 12 ] [   ] [   ] [ 7 ]

```

] [] [] [13] [11]
[6] [1] [16] [4] [] [] [] [] [] [12] [] []
[] [] [2] [] []
[] [5] [] [] [] [15] [] [] [7] [] [] []
[] [] [] [16] []
[] [] [] [] [] [] [] [] [] [9] [8]
[3] [11] [] [] [12]
[] [] [] [9] [10] [] [] [13] [6] [] [] [16] []
5] [1] [] [] []
[] [] [] [] [13] [] [] [1] [2] [10] []
[11] [] [8] [12] []
[] [] [] [] [11] [12] [] [16] [] [] [] []
[] [] [1] [] []
[] [] [] [] [] [] [9] [2] [16] [] [] [1]
[] [5] [] [] []
[] [] [] [] [5] [] [] [] [] [6] [14] []
[] [] [11] [] []

[7] [16] [10] [3] [8] [13] [6] [5] [1] [4] [9] []
12] [14] [15] [11] [2]
[1] [9] [11] [8] [15] [7] [4] [12] [5] [3] [2] []
14] [16] [6] [10] [13]
[14] [13] [6] [12] [16] [2] [10] [9] [15] [11] [7]
[] [8] [4] [3] [5] [1]
[5] [2] [4] [15] [14] [11] [1] [3] [6] [13] [10]
[16] [12] [9] [8] [7]
[16] [3] [13] [5] [2] [9] [11] [15] [4] [8] [1]
[10] [7] [12] [14] [6]
[11] [14] [12] [7] [6] [1] [8] [13] [3] [16] [5]
[9] [15] [4] [2] [10]
[15] [4] [8] [6] [7] [5] [12] [10] [11] [14]
[13] [2] [9] [16] [1] [3]
[9] [10] [1] [2] [3] [4] [16] [14] [12] [15] [6]
[7] [8] [5] [13] [11]
[6] [1] [16] [4] [10] [8] [14] [7] [13] [12]
[11] [15] [3] [2] [9] [5]
[12] [5] [3] [13] [9] [15] [2] [11] [7] [1] [4]
[6] [10] [14] [16] [8]
[2] [15] [7] [14] [1] [16] [5] [4] [10] [9] [8]
[3] [11] [13] [6] [12]
[8] [11] [9] [10] [12] [3] [13] [6] [14] [2]
[16] [5] [1] [7] [4] [15]
[4] [7] [5] [16] [13] [14] [3] [1] [2] [10] [15]
[11] [6] [8] [12] [9]
[10] [6] [14] [9] [11] [12] [15] [16] [8] [5]
[3] [13] [2] [1] [7] [4]
[13] [8] [15] [11] [4] [6] [9] [2] [16] [7] [12]
[1] [5] [10] [3] [14]
[3] [12] [2] [1] [5] [10] [7] [8] [9] [6] [14] []

4] [13] [11] [15] [16]

sudoku(4,0.6)

16 x 16

```
[  ] [  ] [ 15 ] [ 11 ] [ 13 ] [  ] [  ] [ 10 ] [  ] [  ] [ 6 ]
[ 9 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [ 8 ] [ 14 ] [ 15 ] [ 10 ]
[ 13 ] [  ] [  ] [ 9 ] [  ] [  ] [  ] [ 8 ] [ 14 ] [ 15 ] [ 10 ]
[  ] [ 7 ] [  ] [ 16 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 12 ] [  ] [ 8 ] [ 5 ] [  ] [  ] [ 6 ] [  ] [ 7 ] [  ] [  ]
[ 13 ] [  ] [  ] [  ] [ 15 ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 16 ] [  ] [ 10 ] [  ] [  ] [ 5 ] [  ] [ 4 ] [ 3 ] [  ] [ 8 ]
[ 11 ] [  ] [  ] [ 14 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 10 ] [  ] [ 5 ] [  ] [  ] [ 15 ] [  ] [  ] [  ] [  ] [ 4 ] [
] [  ] [  ] [  ] [ 11 ] [  ] [  ] [  ] [  ] [  ] [  ] [ 15 ]
[  ] [  ] [ 11 ] [ 3 ] [  ] [  ] [ 7 ] [  ] [  ] [  ] [  ] [ 15 ]
[ 10 ] [ 13 ] [  ] [ 1 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [ 3
] [ 16 ] [  ] [ 2 ] [  ] [  ] [  ] [  ] [  ] [  ] [ 7 ] [ 3
] [ 16 ] [  ] [ 2 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 7 ] [  ] [ 16 ] [  ] [  ] [ 10 ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 8 ] [  ] [ 15 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[  ] [  ] [  ] [ 16 ] [  ] [  ] [  ] [ 13 ] [  ] [  ] [  ] [  ]
[  ] [ 3 ] [ 14 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[  ] [  ] [  ] [  ] [ 1 ] [  ] [  ] [  ] [ 15 ] [ 3 ] [  ] [ 6
] [ 9 ] [  ] [ 4 ] [ 7 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[  ] [ 15 ] [  ] [ 6 ] [  ] [  ] [ 3 ] [  ] [ 8 ] [ 13 ] [  ] [  ]
[ 16 ] [  ] [  ] [ 5 ] [ 1 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[  ] [  ] [ 3 ] [ 12 ] [ 7 ] [  ] [  ] [ 15 ] [ 1 ] [ 10 ] [ 5 ] [
] [  ] [  ] [ 13 ] [ 16 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[  ] [  ] [ 2 ] [  ] [ 9 ] [  ] [ 13 ] [ 12 ] [ 4 ] [  ] [  ] [  ]
[ 15 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 3 ] [ 11 ] [  ] [ 15 ] [ 10 ] [ 14 ] [  ] [ 5 ] [  ] [  ] [  ] [  ]
[ 7 ] [  ] [ 4 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 1 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [  ] [ 14 ] [  ] [  ]
[  ] [  ] [ 16 ] [ 11 ] [ 9 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 4 ] [ 16 ] [ 14 ] [  ] [ 3 ] [  ] [ 15 ] [ 11 ] [  ] [  ] [ 9 ]
[  ] [  ] [  ] [  ] [  ] [ 5 ] [  ] [  ] [  ] [  ] [  ] [  ]
```

```
[ 14 ] [ 3 ] [ 15 ] [ 11 ] [ 13 ] [ 7 ] [ 16 ] [ 10 ] [ 2 ] [ 12 ] [ 6
] [ 9 ] [ 5 ] [ 1 ] [ 8 ] [ 4 ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 13 ] [ 6 ] [ 4 ] [ 9 ] [ 11 ] [ 12 ] [ 1 ] [ 8 ] [ 14 ] [ 15 ] [  ]
[ 10 ] [ 5 ] [ 7 ] [ 2 ] [ 16 ] [ 3 ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 12 ] [ 1 ] [ 8 ] [ 5 ] [ 2 ] [ 3 ] [ 6 ] [ 14 ] [ 7 ] [ 4 ] [ 16 ] [
13 ] [ 11 ] [ 9 ] [ 10 ] [ 15 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
[ 16 ] [ 7 ] [ 10 ] [ 2 ] [ 15 ] [ 5 ] [ 9 ] [ 4 ] [ 3 ] [ 1 ] [ 8 ] [
11 ] [ 12 ] [ 6 ] [ 14 ] [ 13 ] [  ] [  ] [  ] [  ] [  ] [  ] [  ]
```

```

[ 10 ] [ 12 ] [ 5 ] [ 13 ] [ 16 ] [ 15 ] [ 14 ] [ 3 ] [ 6 ] [ 2 ]
[ 4 ] [ 1 ] [ 8 ] [ 7 ] [ 9 ] [ 11 ]
[ 2 ] [ 8 ] [ 11 ] [ 3 ] [ 4 ] [ 9 ] [ 7 ] [ 6 ] [ 12 ] [ 16 ] [ 15 ]
[ 10 ] [ 13 ] [ 5 ] [ 1 ] [ 14 ]
[ 15 ] [ 14 ] [ 6 ] [ 4 ] [ 8 ] [ 13 ] [ 5 ] [ 1 ] [ 11 ] [ 9 ] [ 7 ]
[ 3 ] [ 16 ] [ 10 ] [ 2 ] [ 12 ]
[ 7 ] [ 9 ] [ 16 ] [ 1 ] [ 12 ] [ 10 ] [ 11 ] [ 2 ] [ 13 ] [ 5 ]
[ 14 ] [ 8 ] [ 4 ] [ 15 ] [ 3 ] [ 6 ]
[ 8 ] [ 10 ] [ 1 ] [ 16 ] [ 5 ] [ 6 ] [ 12 ] [ 13 ] [ 9 ] [ 7 ] [ 11 ]
[ 4 ] [ 3 ] [ 14 ] [ 15 ] [ 2 ]
[ 5 ] [ 2 ] [ 13 ] [ 14 ] [ 1 ] [ 11 ] [ 10 ] [ 16 ] [ 15 ] [ 3 ] [ 12 ]
[ 6 ] [ 9 ] [ 8 ] [ 4 ] [ 7 ]
[ 11 ] [ 15 ] [ 7 ] [ 6 ] [ 14 ] [ 4 ] [ 3 ] [ 9 ] [ 8 ] [ 13 ] [ 2 ]
[ 16 ] [ 10 ] [ 12 ] [ 5 ] [ 1 ]
[ 9 ] [ 4 ] [ 3 ] [ 12 ] [ 7 ] [ 2 ] [ 8 ] [ 15 ] [ 1 ] [ 10 ] [ 5 ] [
14 ] [ 6 ] [ 11 ] [ 13 ] [ 16 ]
[ 6 ] [ 5 ] [ 2 ] [ 8 ] [ 9 ] [ 16 ] [ 13 ] [ 12 ] [ 4 ] [ 11 ] [ 1 ]
[ 15 ] [ 14 ] [ 3 ] [ 7 ] [ 10 ]
[ 3 ] [ 11 ] [ 9 ] [ 15 ] [ 10 ] [ 14 ] [ 2 ] [ 5 ] [ 16 ] [ 6 ]
[ 13 ] [ 7 ] [ 1 ] [ 4 ] [ 12 ] [ 8 ]
[ 1 ] [ 13 ] [ 12 ] [ 10 ] [ 6 ] [ 8 ] [ 4 ] [ 7 ] [ 5 ] [ 14 ] [ 3 ]
[ 2 ] [ 15 ] [ 16 ] [ 11 ] [ 9 ]
[ 4 ] [ 16 ] [ 14 ] [ 7 ] [ 3 ] [ 1 ] [ 15 ] [ 11 ] [ 10 ] [ 8 ] [ 9 ]
[ 12 ] [ 2 ] [ 13 ] [ 6 ] [ 5 ]

```

```

sudoku(5,0.0)
sudoku(5,0.2)
sudoku(5,0.4)
sudoku(5,0.6)

```

```

sudoku(6,0.0)
sudoku(6,0.2)
sudoku(6,0.4)
sudoku(6,0.6)

```

Como podemos observar os tempos de execução tornam-se substancialmente mais elevados a medida que aumentamos o tamanho do tabuleiro e a fração de casas a serem preenchidas, isto deve-se ao aumento do número variáveis que resultará num aumento no numero de operações