

## Annex A Python Code

### Environment.py

```
1
2     """
3     Modified on Tue Jan 03 2023
4
5     @author: Bjorn Hogmo
6     """
7     import numpy as np
8     from numpy import random as rd
9     #
=====
10    #                                     Create LP Model Function
11    #
=====
12
13    def CreateModel():
14        #      Size of Blocks
15        w=[50,200,100,100,90,150]  # Width
16        h=[70,90,80,90,84,85]    # Height
17
18        delta=[20,20,20,20,40,20]      #Gap
19
20        rin  = [0.7,0.7,0.7,0.7,0.04,0.7]  #Location of Input Gate
21        rout = [0.7,0.7,0.7,0.7,0.04,0.7]  #Location of Output Gate
22
23
24        n= np.shape(w)[0]
25        a = [[1,1,0.9,0.4,0,0],
26              [1,1,0.5,0,0.5,0],
27              [1,1,0,0,0.26,0.5],
28              [1,1,0,0.93,0.15,0],
29              [1,1,0,0.24,0,0.5],
30              [1,0,0.05,0,0,0]]
31
32
33        W=1000
34        H=800
35
36        phi=50000
37        xin=np.zeros((n,))
38        yin=np.zeros((n,))
39
40        xout=np.zeros((n,))
41        yout=np.zeros((n,))
42
43        for i in range(0,n):
44            if ( rin[i]>=0 ) and ( rin[i]<= 0.25):
45                xin[i] = (4*rin[i]-0.5)*w[i]
46                yin[i] = -h[i]/2
47
48            if ( rin[i]>0.25 ) and ( rin[i] <=0.5 ):
49                xin[i]= w[i]/2
```













## Annex B Python Code

### MainEnvironment.py

```
1  from tkinter import *
2  import Environment as EnvPSO
3  import matplotlib.pyplot as plt
4  from drawnow import drawnow
5  import numpy as np
6  import json
7  import pandas as pd
8
9
10 trace = 0
11
12
13 class MyEnv(Frame):
14     def __init__(self, master):
15         Frame.__init__(self, master=None)
16
17         self.UNIT = UNIT    # pixels
18         self.H = 9# grid height
19         self.W = 11 # grid width
20         self.x = self.y = 0
21         canvas = Canvas(self, height=self.H * UNIT, width=self.W * UNIT, background = 'black', cursor="cross")
22         #
23         s = Scrollbar(self, command=canvas.yview)
24         canvas.pack(side=LEFT)
25         #
26         s.pack(side=RIGHT)
27         canvas.configure(yscrollcommand=s.set)
28         self.canvas = canvas
29         self.drawn = None
30         self.color = None
31         self.bttm_clicks = 0
32         self.kinds =
33         [canvas.create_rectangle, canvas.create_oval, canvas.create_line]
34
35         #
36         =====#
37         #
38         self.menubar = Menu(master)
39         self.filemenu = Menu(self.menubar, tearoff=0)
40         self.filemenu.add_command(label="Exit", command= master.destroy)
41
42         self.menubar.add_cascade(label="File", menu=self.filemenu)
43
44         self.b1 = Button(self, text="Generate!", command=self.Generate)
45         self.b1.pack()
46         self.b1.config(bg='dark green', fg='white')
47         self.b1.config(font=('helvetica', 20, 'underline italic'))
48
49         with open('Data.json') as data_file:
50             self.color = json.load(data_file)
```











