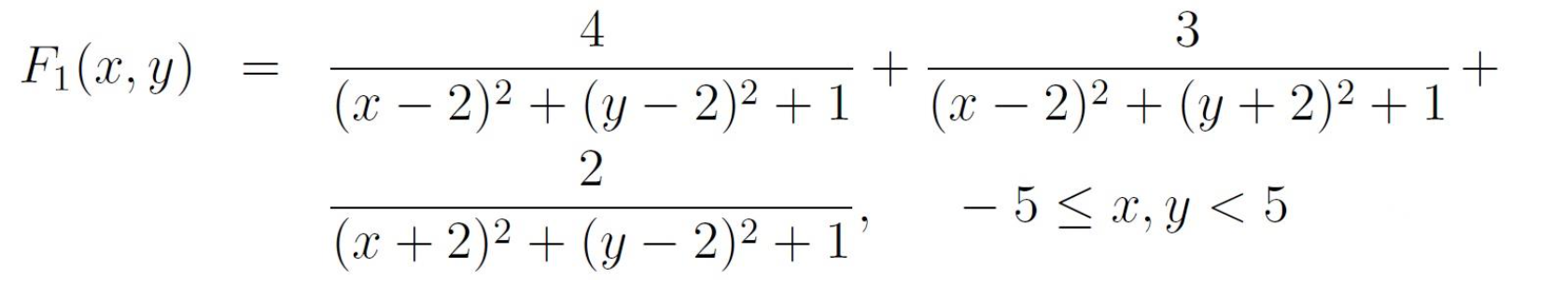


**國立高雄科技大學電子工程系**

**112 學年度第一學期人工智慧導論**

|  |  |
| --- | --- |
| **Professor:** Chin-Shiuh Shieh  **Subject:**  Heuristic Optimization Methods(啟發式最佳化方法) | **日期**: 03/07/2023  **Email your answer to:**  csshieh@nkust.edu.tw |
| **學生**: NGUYEN THAC HUNG 阮碩雄 | **學號**: F111169102 |

**Question**: Find Maximum F1(x, y) :

****

**Answer**:

**Idea to solve problem:**

Applying Gradient -Ascent Method to find values.

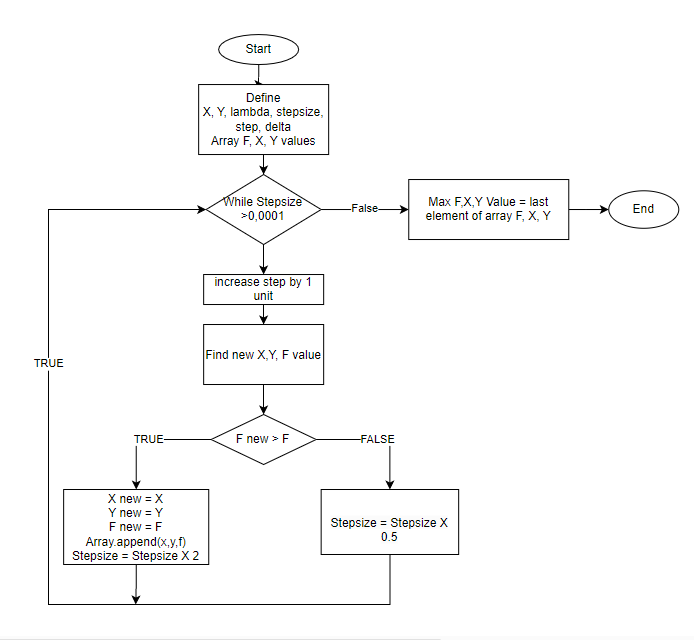
Using loop while with condition of step\_size > 0.0001.

- Increase the step by 1unit after looping to find the number of steps it takes to reach the optima.

- Get 3 values including Fvalue, Xvalue and Yvalue after looping. When the program completed, get 3 arrays including Fvalue Array, Xvalue Array and Y value Array. Using data to plot graph by 3 values. Then combinate it to plot function graph 3D. So I can use this combination graph to obviously report a case of being trapped at local optimum due to improper initialization.

**Support tool:** PyCharm Professional 2022, Numpy and Matplotlib libary.

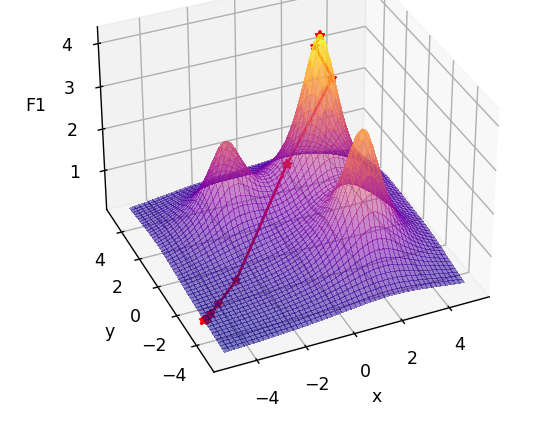
**Algorithm flow chart:**



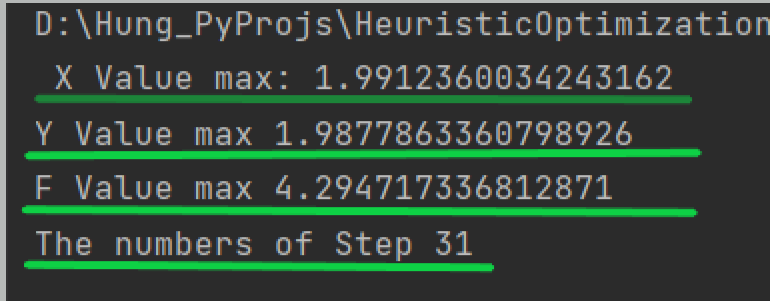
**Python program:**

# Function void  
def F1(x, y):  
 return 4 / ((x - 2) \*\* 2 + (y - 2) \*\* 2 + 1) + \  
 3 / ((x - 2) \*\* 2 + (y + 2) \*\* 2 + 1) + \  
 2 / ((x + 2) \*\* 2 + (y - 2) \*\* 2 + 1)  
  
# define Valuea  
x = random.randrange(-5, 5, 1)  
y = random.randrange(-5, 5, 1)  
  
delta = 0.01  
Step\_size = 0.01  
step = 0  
  
aF = arr.array('f', [])  
aX = arr.array('f',[])  
aY = arr.array('f',[])  
  
# While Loop  
while(Step\_size > 0.0001):  
 step = step + 1  
 nx = x + Step\_size \* (F1(x+delta, y) - F1(x, y))/delta  
 ny = y + Step\_size \* (F1(x, y + delta) - F1(x, y)) / delta  
 f = F1(x, y)  
 af.append(f)  
 nf = F1(nx, ny)  
 aX.append(x)  
 aY.append(y)  
 if nf > f:  
 x = nx  
 y = ny  
 Step\_size = Step\_size \* 2  
 else:  
 Step\_size = Step\_size \* 0.5  
  
  
print(' X Value max:', x)  
print('Y Value max', y)  
print('F Value max', f)  
print('The numbers of Step', step)  
  
# plot function graph  
xAxis = np.arange(-5, 5, 0.01)  
yAxis = np.arange(-5, 5, 0.01)  
X, Y = np.meshgrid(xAxis, yAxis)  
arrF1 = F1(x=X, y=Y)  
# Matching (x, y) into an array  
fig = plt.figure()  
ax = plt.axes(projection='3d')  
ax.plot\_surface(X=X, Y=Y, Z=arrF1, cmap="plasma", linewidth=0, antialiased=False,alpha=0.5)  
ax.plot(xs = aX, ys = aY, zs = af, color='red',marker='\*', markersize=5 )  
ax.set\_xlabel('x')  
ax.set\_ylabel('y')  
ax.set\_zlabel('F1')  
plt.show()

**Plot 3D of Function**



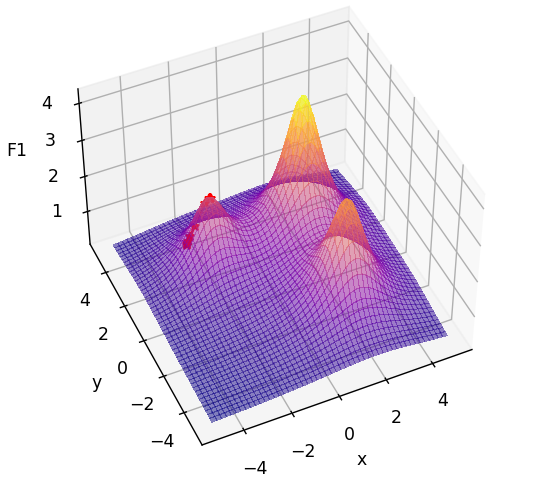
After run program, we have result:

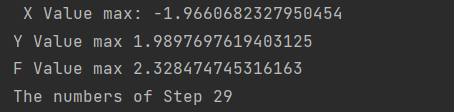


With x = 1.9912, y = 1.9977, Step = 31 => Maximize of F1 = 4.2947

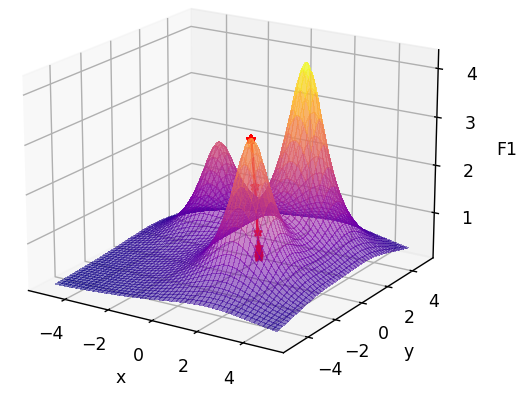
**Giving some case of being trapped at local optimum due to improper initialization:**

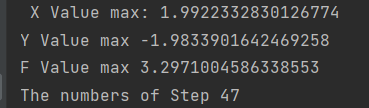
**Case 1:**





**Case 2:**





**Thanks for watching**