Convex hull

Graficación Computacional

Universidad Autonoma del Estado de México

Flores Hidalgo Yair Uriel

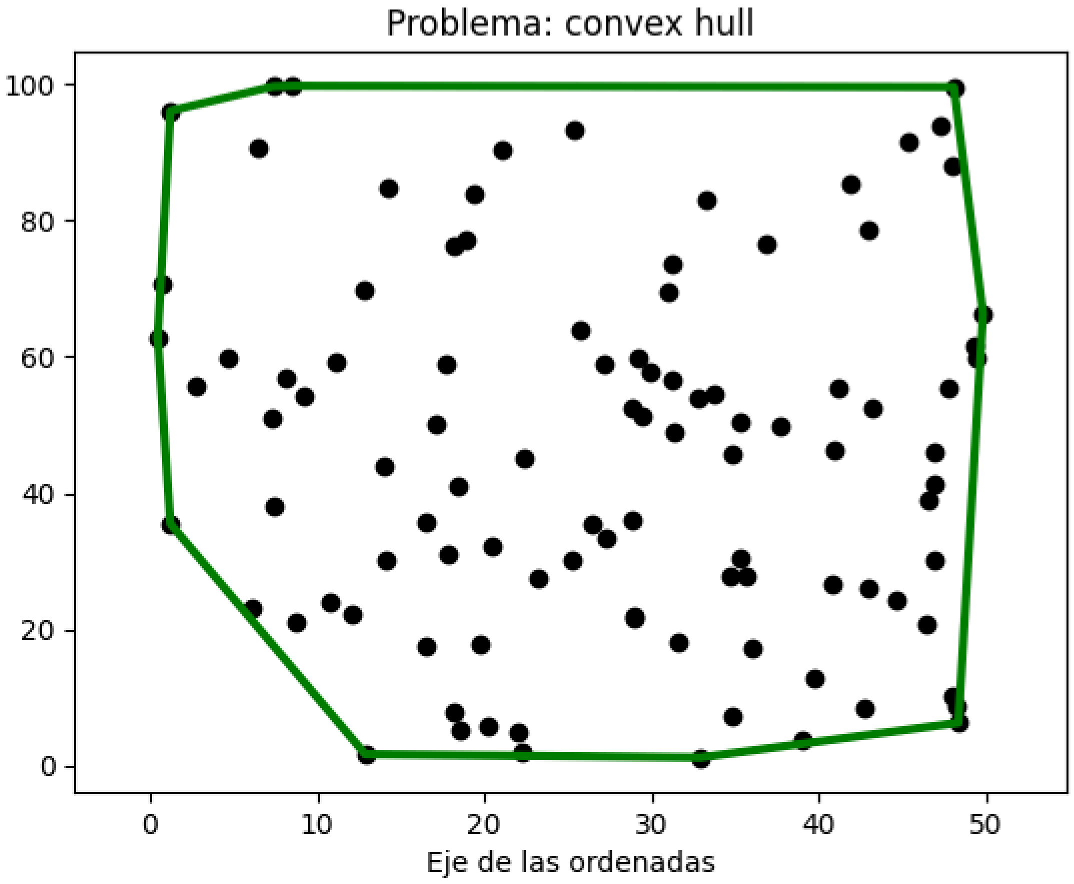
Alvarez Rodriguez Hazem

07/10/2024

Ejemplo 1 (realizado en clase)

In [18]:

```
import random as rand
import numpy as np
import matplotlib.pyplot as plt


#Funciones

def turn_right():
    array = [coord_points[0], coord_points[1]]
    for i in range (2, len(coord_points)):
        array.append(coord_points[i])
        while len(array) > 2 and np.linalg.det([array[-3], array[-2], array[-1]]) >
            array.pop(-2)
    return array
def convex_hull():
    coord_points.sort()
    l_upper = turn_right()
    coord_points.reverse()
    l_lower = turn_right()
    l = l_upper + l_lower
    return l

def graph(convex_pol, coord_points):
    #Acomodando listas adecuadas

    x_points = [i[0] for i in coord_points]
    y_points = [i[1] for i in coord_points]

    x_polygon = [i[0] for i in convex_pol]
    y_polygon = [i[1] for i in convex_pol]

    #definiendo limites de la grafica
    x_lim_der = max(x_points)+5
    y_lim_sup = max(y_points)+5
    x_lim_izq = min(x_points)-5
    y_lim_inf = min(y_points)-5
    # Asignacion de los liites extremos
    plt.xlim(x_lim_izq,x_lim_der)
    plt.ylim(y_lim_inf, y_lim_sup)

    #Graficacion
    plt.title('Problema: convex hull')
    plt.xlabel('Eje de las abscisas')
    plt.xlabel('Eje de las ordenadas')
    plt.plot(x_points, y_points, 'ko')
    plt.plot(x_polygon, y_polygon, 'g-', linewidth = 3.0)

#Generacion de coordenadas de forma aleatorea
num_points = 100
coord_points = []
for i in range(num_points): coord_points.append([rand.uniform(0,50), rand.uniform(0

# creacion y graficacion
convex_pol = convex_hull()
graph(convex_pol,coord_points)
```

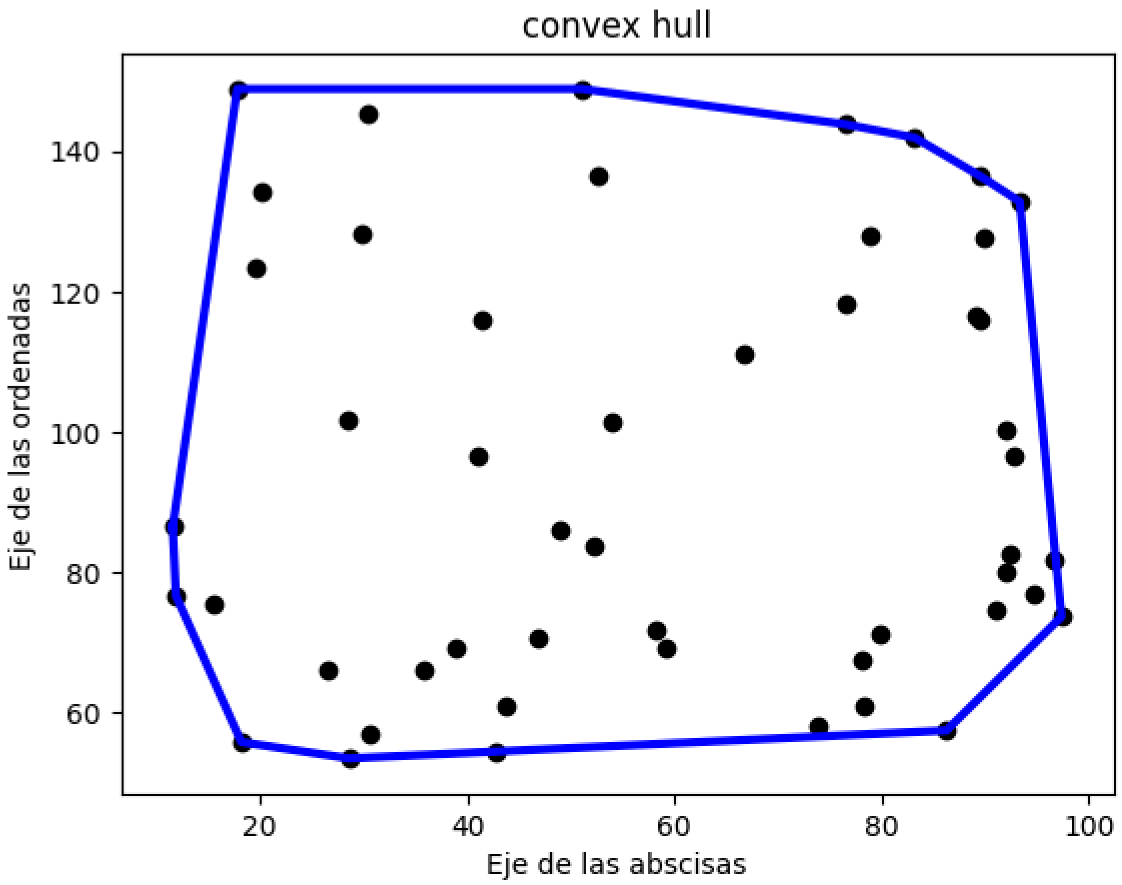


Ejemplo 2

In [29]:

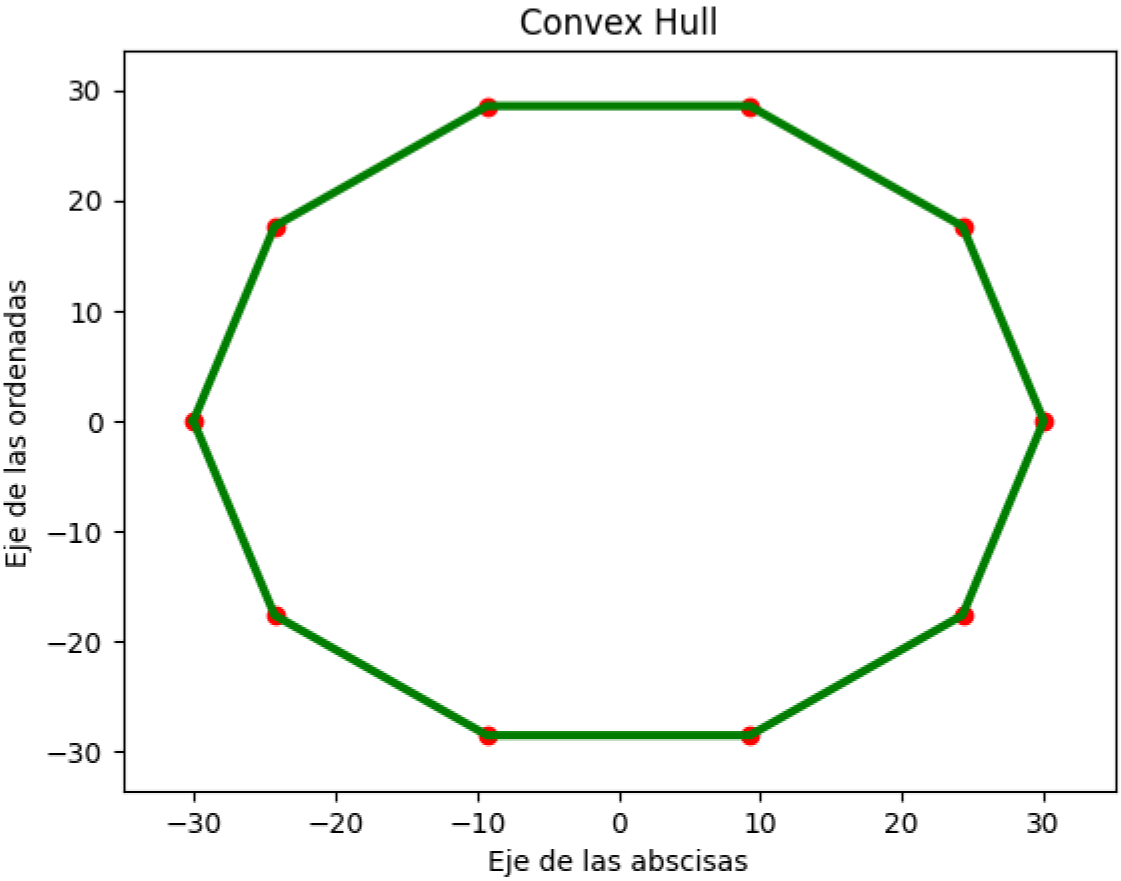
```
import random as rand
import numpy as np
import matplotlib.pyplot as plt

# Funciones
```

```
def turn_right():
    array = [coord_points[0], coord_points[1]]
    for i in range(2, len(coord_points)):
        array.append(coord_points[i])
        while len(array) > 2 and np.linalg.det([array[-3], array[-2], array[-1]]) >
            array.pop(-2)
    return array

def convex_hull():
    coord_points.sort()
    l_upper = turn_right()
    coord_points.reverse()
    l_lower = turn_right()
    l = l_upper + l_lower
    return l

def graph(convex_pol, coord_points):
    x_points = [i[0] for i in coord_points]
    y_points = [i[1] for i in coord_points]

    x_polygon = [i[0] for i in convex_pol]
    y_polygon = [i[1] for i in convex_pol]

    x_lim_der = max(x_points) + 5
    y_lim_sup = max(y_points) + 5
    x_lim_izq = min(x_points) - 5
    y_lim_inf = min(y_points) - 5

    plt.xlim(x_lim_izq, x_lim_der)
    plt.ylim(y_lim_inf, y_lim_sup)
    plt.title('convex hull')
    plt.xlabel('Eje de las abscisas')
    plt.ylabel('Eje de las ordenadas')
    plt.plot(x_points, y_points, 'ko')
    plt.plot(x_polygon, y_polygon, 'b-', linewidth=3.0)
    plt.show()


num_points = 50
coord_points = [[rand.uniform(10, 100), rand.uniform(50, 150), 1.0] for i in range(

# Creacion y graficacion
convex_pol = convex_hull()
graph(convex_pol, coord_points)
```



Ejemplo 3

In [28]:
```
import random as rand
import numpy as np
import matplotlib.pyplot as plt

# Funciones
def turn_right():
    array = [coord_points[0], coord_points[1]]
    for i in range(2, len(coord_points)):
        array.append(coord_points[i])
        while len(array) > 2 and np.linalg.det([array[-3], array[-2], array[-1]]) >
            array.pop(-2)
    return array

def convex_hull():
    coord_points.sort()
    l_upper = turn_right()
    coord_points.reverse()
    l_lower = turn_right()
    l = l_upper + l_lower
    return l

def graph(convex_pol, coord_points):
    x_points = [i[0] for i in coord_points]
    y_points = [i[1] for i in coord_points]
    x_polygon = [i[0] for i in convex_pol]
    y_polygon = [i[1] for i in convex_pol]

    x_lim_der = max(x_points) + 5
    y_lim_sup = max(y_points) + 5
    x_lim_izq = min(x_points) - 5
    y_lim_inf = min(y_points) - 5

    plt.xlim(x_lim_izq, x_lim_der)
    plt.ylim(y_lim_inf, y_lim_sup)
    plt.title('Convex Hull')
    plt.xlabel('Eje de las abscisas')
    plt.ylabel('Eje de las ordenadas')
    plt.plot(x_points, y_points, 'ro')
    plt.plot(x_polygon, y_polygon, 'g-', linewidth=3.0)
    plt.show()


num_points = 10
radius = 30
coord_points = [[radius * np.cos(2 * np.pi * i / num_points), radius * np.sin(2 * n

# Creacion y graficacion
convex_pol = convex_hull()
graph(convex_pol, coord_points)
```



```
In [ ]:
```