POLITECNICO
MILANO 1863

SCUOLA DELL'INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

COMPUTER SCIENCE AND ENGINEERING

Internet Of Things

1st Challenge Exercise 4

Simone Pio Bottaro 10774229

Gabriele Lorenzetti 10730455

A.A 2024/2025

Here some useful data and functions for completing exercise 4.

The energy is calculated with the formula E = b*(Ec + Etx(d)) where Etx(d) = k * d^2 nJ/bit, d is the distance from the sensor to the sink, k = 1 nJ/bit/m^2.

```python
import numpy as np
import matplotlib.pyplot as plt

sensors = np.array([[1, 2], [10, 3], [4, 8], [15, 7], [6, 1], [9, 12], [14, 4], [3, 10], [7, 7], [12, 14]])

b = 2000 #bit
Eb = 5000000 #nJ
Ec = 50 #nJ/bit
time = 10 #min

#calculate the distance between two points
def distance(point_1, point_2):
  x = point_2[0] - point_1[0]
  y = point_2[1] - point_1[1]
  distance = np.sqrt(x**2 + y**2)
  return distance

#calculate the energy between sensor and sink
def energy(dist):
  energy = b*(Ec + dist**2)
  return energy
```

4a) Lifetime of the system

First, we calculated the distance of all the points from the sink to see which one is the furthest and therefore the one that would consume the most energy.

The furthest one is the [1,2] point with a distance of 26,17 m from the sink.

```python
#calculation of all distances
for sensor in sensors:
  dist = distance(sensor, [20,20])
  print("Distanza tra sink e", sensor, ":", dist, "meters")
```

```
Distanza tra sink e [1 2] : 26.1725046566048 meters
Distanza tra sink e [10  3] : 19.72308292331602 meters
Distanza tra sink e [4 8] : 20.0 meters
Distanza tra sink e [15  7] : 13.92838827718412 meters
Distanza tra sink e [6 1] : 23.600847442411894 meters
Distanza tra sink e [ 9 12] : 13.601470508735444 meters
Distanza tra sink e [14  4] : 17.08800749063506 meters
Distanza tra sink e [ 3 10] : 19.72308292331602 meters
Distanza tra sink e [7 7] : 18.384776310850235 meters
Distanza tra sink e [12 14] : 10.0 meters
```

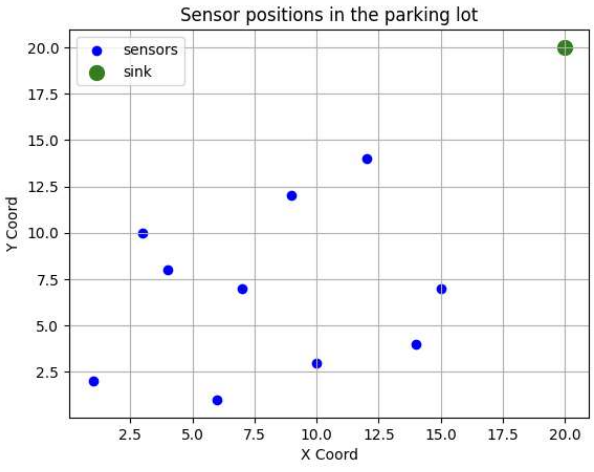
```python
#Sensor and sink position plot
x_coords = [sensor[0] for sensor in sensors]
y_coords = [sensor[1] for sensor in sensors]
plt.scatter(x_coords, y_coords, color='b', label='sensors')

plt.scatter(20, 20, color='g', marker='o', s=100, label='sink')

plt.xlabel('X Coord')
plt.ylabel('Y Coord')
plt.title('Sensor positions in the parking lot')

plt.legend()
plt.grid(True)

plt.show()
```



Once we found the point we calculated the energy with the previous function, 1470000 nJ.

The battery is of 5000000 nJ and the sensor can perform 3,4 cycles in 34 minutes because each sensor transmits a status update every 10 minutes.

So, it will be able to perform 3 completed cycles in 30 minutes.

```
#the point [1,2] it's the furthest and therefore the one that will consume the most energy
dist_A = distance([1,2], [20,20])
energy_A = energy(dist_A)
print(energy_A, "nJ")

cycles = 5000000/energy_A
print(cycles, "cycles")

life = cycles*time
print("the battery life is", life, "minutes")
```

```
1470000.0 nJ
3.401360544217687 cycles
the battery life is 34.01360544217687 minutes
```

4b) Optimal position of the sink

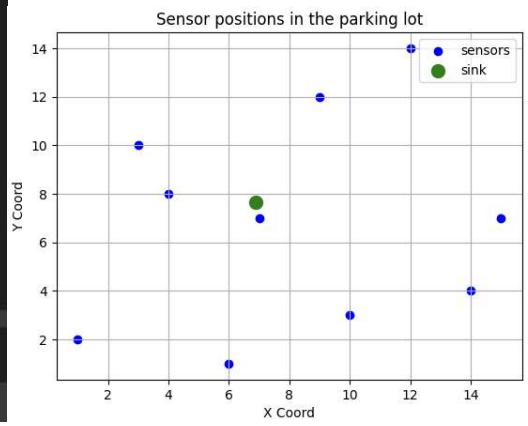
To find the optimal position of the sink we tried all the possible points where to put the sink, calculating each time the energy consumed by the furthest sensor. The optimal point for the sink is [6,87;7,66] and the sensor that consume the most energy is [1;2] whit 233065 nJ.

Which is able to perform 21,45 cycles in 214,5 minutes.

```
#iterates between all possible locations of the sink and the sensors
dist_max = 0
ener_min = 1000000000
x = 0
y = 0
n = 0
step_size = 0.01
for i in np.arange(0, 16, step_size):
  for j in np.arange(0, 16, step_size):
    for z in range(len(sensors)):
      dist = distance([i, j], sensors[z])
      if dist > dist_max:
        dist_max = dist
        n = z
    ener = energy(dist_max)
    if ener < ener_min:
      ener_min = ener
      x = i
      y = j
    dist_max = 0

print(n, x, y, ener_min)
```

```
0 6.87 7.66 233064.99999999994
```




```
#Calculate battery life
life = (5000000 / ener_min) * 10
print("the battery life is", life, "minutes")
```

```
the battery life is 214.53242657627703 minutes
```

4C)

The choice between a fixed and a dynamic sink involves several trade-offs in terms of system efficiency and lifetime.

• **Fixed sink:** simpler and cheaper, requiring no tracking algorithms or position updates. It also avoids energy consumption for movement. However, it leads to uneven power consumption among sensors, with those farther from the sink depleting faster, reducing network lifetime and limiting adaptability.

• **Dynamic sink:** distributes energy consumption more evenly, extending network lifetime and adapting to changes or failures. However, it requires additional hardware and software, increasing complexity and consuming energy for movement.

Conclusion: a mobile sink can significantly extend network lifetime, but its movement must be optimized to minimize energy consumption and synchronize with sensor activity cycles.

In relation to exercise 4, implementing a dynamic sink does not in our opinion make sense, as there are only 10 sensors all relatively close to the sink in the optimal position and displacing the sink would require too much energy.