```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.spatial import distance_matrix
from sys import maxsize
from itertools import permutations

speed = float(input("Enter speed (m/s): "))
xarray = np.array([])
yarray = np.array([])

s_diviation = 1.5
num_tennisballs_per_area = 2 ####Enter number of tennisballs per sub_area. The total amount of
tennisballs will be five times larger than this number. The upper limit, depending on your
computer powe, is around 1-2.
num_subareas = 1
med_x_coordinates = [3,3,27,27,15]
med_y_coordinates = [3,17,17,3,10]

for i in range(len(med_x_coordinates)):
  x = np.random.normal(med_x_coordinates[i], s_diviation, num_tennisballs_per_area)
  y = np.random.normal(med_y_coordinates[i], s_diviation, num_tennisballs_per_area)
  xarray = np.append(xarray, [x]).reshape(-1,1)
  yarray = np.append(yarray, [y]).reshape(-1,1)
  xycoor = np.concatenate((xarray,yarray),axis = 1)
  plt.scatter(x,y, c="yellow")

list_of_coordinates = xycoor.tolist()
rounded_list_of_coordinates = [[round(val, 1) for val in sublst] for sublst in
list_of_coordinates]
data = rounded_list_of_coordinates
df = pd.DataFrame(data)
matrix = pd.DataFrame(distance_matrix(df.values, df.values))
rounded_matrix = round(matrix, 1)
#Automatically multiplies by 5 (5 subareas)
num_coordinates = num_tennisballs_per_area * 5
#Shortest distance generator
def minimum_distance(graph, s):
 vertex = []
 for i in range(num_coordinates):
  if i != s:
   vertex.append(i)
 min_path = maxsize
 next_permutation=permutations(vertex)
 for i in next_permutation:
  distance = 0
  k = s
  for j in i:
   distance += graph[k][j]
   k = j
  distance += graph[k][s]
  min_path = min(min_path, distance)
 return min_path

if __name__ == "__main__":
 graph = rounded_matrix
 s = 0
 print("The minimum distance is:", round(minimum_distance(graph, s),1), "meters")

minimal_distance = minimum_distance(graph, s)
print("This would take", round(minimal_distance / speed, 1), "seconds to collect")

img = plt.imread("tenniscourt blue.jpg")
plt.imshow(img, extent=[0, 30, 0, 20])
plt.show()
```