In [1]:
```
pip install pygad
```

```
Requirement already satisfied: pygad in c:\users\91903\appdata\local\programs
\python\python310\lib\site-packages (3.0.1)
Requirement already satisfied: cloudpickle in c:\users\91903\appdata\local\pr
ograms\python\python310\lib\site-packages (from pygad) (2.2.1)
Requirement already satisfied: matplotlib in c:\users\91903\appdata\local\pro
grams\python\python310\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\91903\appdata\local\programs
\python\python310\lib\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\91903\appdata\loc
al\programs\python\python310\lib\site-packages (from matplotlib->pygad) (1.0.
7)
Requirement already satisfied: cycler>=0.10 in c:\users\91903\appdata\local\p
rograms\python\python310\lib\site-packages (from matplotlib->pygad) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\91903\appdata\lo
cal\programs\python\python310\lib\site-packages (from matplotlib->pygad) (4.3
9.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\91903\appdata\lo
cal\programs\python\python310\lib\site-packages (from matplotlib->pygad) (1.
4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\91903\appdata\loca
l\programs\python\python310\lib\site-packages (from matplotlib->pygad) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\91903\appdata\local
\programs\python\python310\lib\site-packages (from matplotlib->pygad) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\91903\appdata\loc
al\programs\python\python310\lib\site-packages (from matplotlib->pygad) (3.0.
9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\91903\appdata
\local\programs\python\python310\lib\site-packages (from matplotlib->pygad)
(2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\91903\appdata\local\progr
ams\python\python310\lib\site-packages (from python-dateutil>=2.7->matplotlib
->pygad) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

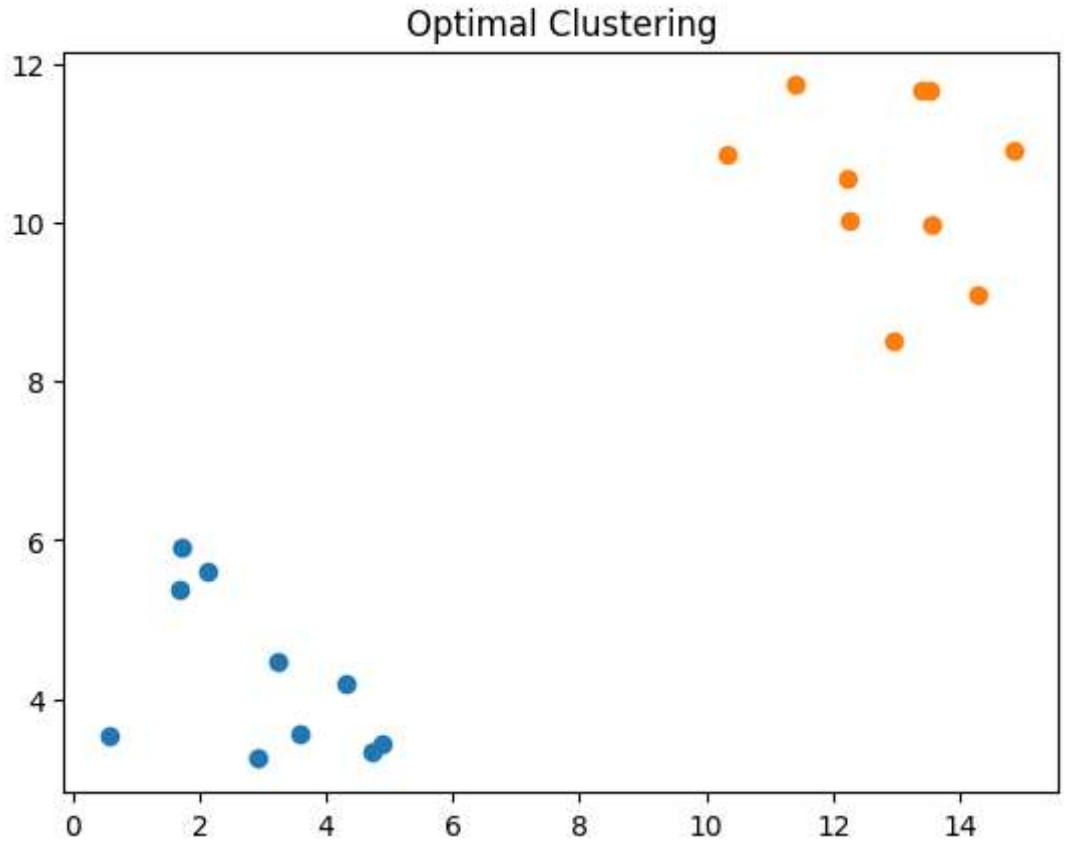
In [2]:
```
import numpy
import matplotlib.pyplot
import pygad
```

In [3]:
```
cluster1_num_samples = 10
cluster1_x1_start = 0
cluster1_x1_end = 5
cluster1_x2_start = 2
cluster1_x2_end = 6
cluster1_x1 = numpy.random.random(size=(cluster1_num_samples))
cluster1_x1 = cluster1_x1 * (cluster1_x1_end - cluster1_x1_start) + cluster1_x
cluster1_x2 = numpy.random.random(size=(cluster1_num_samples))
cluster1_x2 = cluster1_x2 * (cluster1_x2_end - cluster1_x2_start) + cluster1_x
cluster2_num_samples = 10
cluster2_x1_start = 10
cluster2_x1_end = 15
cluster2_x2_start = 8
cluster2_x2_end = 12
cluster2_x1 = numpy.random.random(size=(cluster2_num_samples))
cluster2_x1 = cluster2_x1 * (cluster2_x1_end - cluster2_x1_start) + cluster2_x
cluster2_x2 = numpy.random.random(size=(cluster2_num_samples))
cluster2_x2 = cluster2_x2 * (cluster2_x2_end - cluster2_x2_start) + cluster2_x
```

In [4]:
```
c1 = numpy.array([cluster1_x1, cluster1_x2]).T
c2 = numpy.array([cluster2_x1, cluster2_x2]).T
data = numpy.concatenate((c1, c2), axis=0)
data
```

Out[4]:
```
array([[ 4.71957267,  3.3421973 ],
       [ 2.13899893,  5.5964932 ],
       [ 1.71978922,  5.9059297 ],
       [ 4.30544821,  4.20170886],
       [ 3.58060818,  3.55570246],
       [ 4.87400977,  3.4398681 ],
       [ 2.92718848,  3.24779641],
       [ 0.57201343,  3.54867029],
       [ 3.24631879,  4.47522238],
       [ 1.67603875,  5.38257037],
       [13.54297088,  9.9812031 ],
       [10.34086962, 10.84080038],
       [14.85021638, 10.89612418],
       [12.23752922, 10.55925508],
       [12.24618212, 10.02545591],
       [11.40389245, 11.72428224],
       [12.9524025 ,  8.49739914],
       [13.41203951, 11.6710607 ],
       [13.52078719, 11.6480618 ],
       [14.27371764,  9.08626223]])
```

In [5]:
```
matplotlib.pyplot.scatter(cluster1_x1, cluster1_x2)
matplotlib.pyplot.scatter(cluster2_x1, cluster2_x2)
matplotlib.pyplot.title("Optimal Clustering")
matplotlib.pyplot.show()
```



In [6]:
```
def euclidean_distance(X, Y):
    return numpy.sqrt(numpy.sum(numpy.power(X - Y, 2), axis=1))
```

In [8]:
```
def cluster_data(solution, solution_idx):
    global num_cluster, data
    feature_vector_length = data.shape[1]
    cluster_centers = []
    all_clusters_dists = []
    clusters = []
    clusters_sum_dist = []

    for clust_idx in range(num_clusters):
        cluster_centers.append(solution[feature_vector_length*clust_idx:featur
        cluster_center_dists = euclidean_distance(data, cluster_centers[clust_
        all_clusters_dists.append(numpy.array(cluster_center_dists))

    cluster_centers = numpy.array(cluster_centers)
    all_clusters_dists = numpy.array(all_clusters_dists)

    cluster_indices = numpy.argmin(all_clusters_dists, axis=0)
    for clust_idx in range(num_clusters):
        clusters.append(numpy.where(cluster_indices == clust_idx)[0])

        if len(clusters[clust_idx]) == 0:
            clusters_sum_dist.append(0)
        else:
            clusters_sum_dist.append(numpy.sum(all_clusters_dists[clust_idx, c
    clusters_sum_dist = numpy.array(clusters_sum_dist)
    return cluster_centers, all_clusters_dists, cluster_indices, clusters, clu
```

In [9]:
```
def fitness_func(ga_instance,solution, solution_idx):
    _, _, _, _, clusters_sum_dist = cluster_data(solution, solution_idx)
    fitness = 1.0 / (numpy.sum(clusters_sum_dist) + 0.00000001)
    return fitness
```

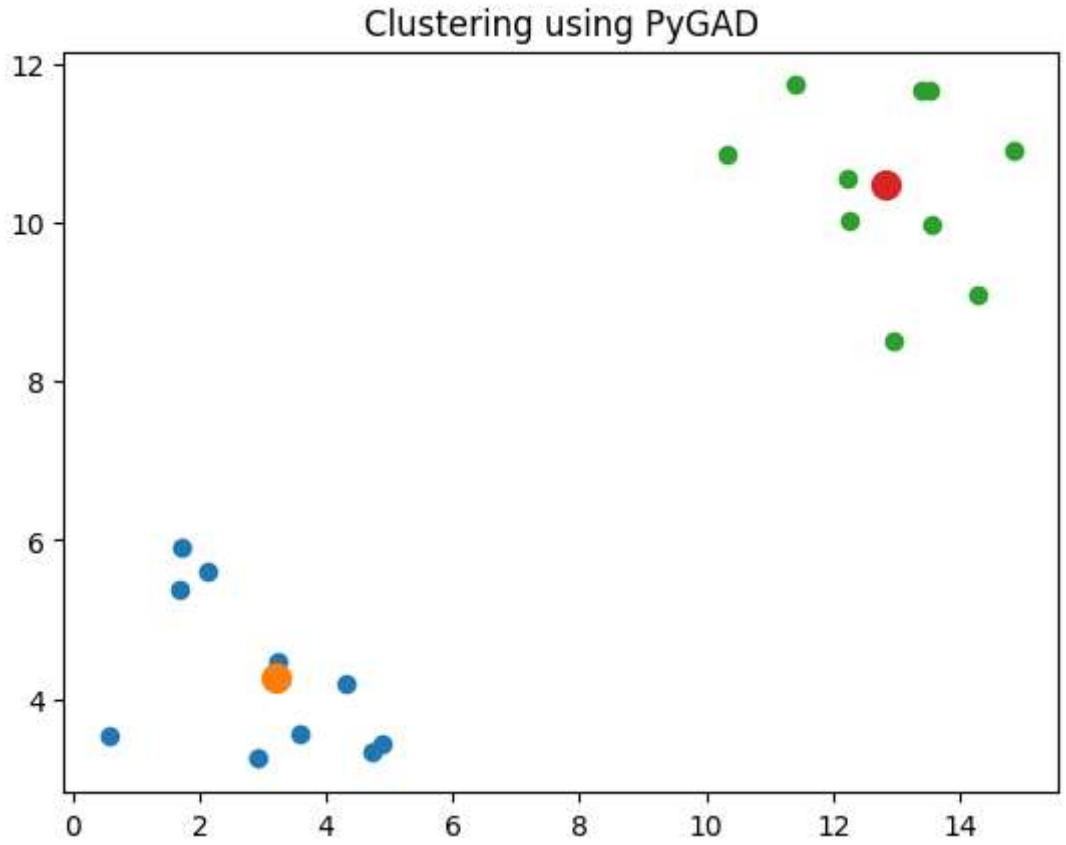
In [10]:
```
num_clusters = 2
num_genes = num_clusters * data.shape[1]

ga_instance = pygad.GA(num_generations=100,
                       sol_per_pop=10,
                       num_parents_mating=5,
                       init_range_low=-6,
                       init_range_high=20,
                       keep_parents=2,
                       num_genes=num_genes,
                       fitness_func=fitness_func,
                      suppress_warnings=True)
ga_instance.run()
```

In [11]:
```
best_solution, best_solution_fitness, best_solution_idx = ga_instance.best_sol
print("Best solution is {bs}".format(bs=best_solution))
print("Fitness of the best solution is {bsf}".format(bsf=best_solution_fitness
print("Best solution found after {gen} generations".format(gen=ga_instance.bes
```

```
Best solution is [ 3.21485788  4.26172861 12.83619304 10.47427056]
Fitness of the best solution is 0.03256901416783241
Best solution found after 59 generations
```

In [12]:
```
cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_d
```

In [13]:
```
for cluster_idx in range(num_clusters):
    cluster_x = data[clusters[cluster_idx], 0]
    cluster_y = data[clusters[cluster_idx], 1]
    matplotlib.pyplot.scatter(cluster_x, cluster_y)
    matplotlib.pyplot.scatter(cluster_centers[cluster_idx, 0], cluster_centers
matplotlib.pyplot.title("Clustering using PyGAD")
matplotlib.pyplot.show()
```



In [ ]:
```
```