

In [19]:

```
pip install pygad
```

```
Requirement already satisfied: pygad in c:\users\raja\appdata\local\progra
ms\python\python310\lib\site-packages (3.0.1)
Requirement already satisfied: cloudpickle in c:\users\raja\appdata\local
\programs\python\python310\lib\site-packages (from pygad) (2.2.1)
Requirement already satisfied: matplotlib in c:\users\raja\appdata\local\p
rograms\python\python310\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\raja\appdata\local\progra
ms\python\python310\lib\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\raja\appdata\l
ocal\programs\python\python310\lib\site-packages (from matplotlib->pygad)
(1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\raja\appdata\local
\programs\python\python310\lib\site-packages (from matplotlib->pygad) (0.1
1.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\raja\appdata
\local\programs\python\python310\lib\site-packages (from matplotlib->pyga
d) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\raja\appdata
\local\programs\python\python310\lib\site-packages (from matplotlib->pyga
d) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\raja\appdata\lo
cal\programs\python\python310\lib\site-packages (from matplotlib->pygad)
(23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\raja\appdata\loca
l\programs\python\python310\lib\site-packages (from matplotlib->pygad) (9.
5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\raja\appdata\l
ocal\programs\python\python310\lib\site-packages (from matplotlib->pygad)
(3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\raja\appda
ta\local\programs\python\python310\lib\site-packages (from matplotlib->pyg
ad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\raja\appdata\local\pro
grams\python\python310\lib\site-packages (from python-dateutil>=2.7->matpl
otlib->pygad) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [20]:

```
import numpy
import matplotlib.pyplot
import pygad
```

In [21]:

```
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_x1_start
cluster1_x2 = numpy.random.random(size=(cluster1_num_samples))
cluster1_x2 = cluster1_x2 * (cluster1_x2_end - cluster1_x2_start) + cluster1_x2_start
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_x1_start
cluster2_x2 = numpy.random.random(size=(cluster2_num_samples))
cluster2_x2 = cluster2_x2 * (cluster2_x2_end - cluster2_x2_start) + cluster2_x2_start
```

In [22]:

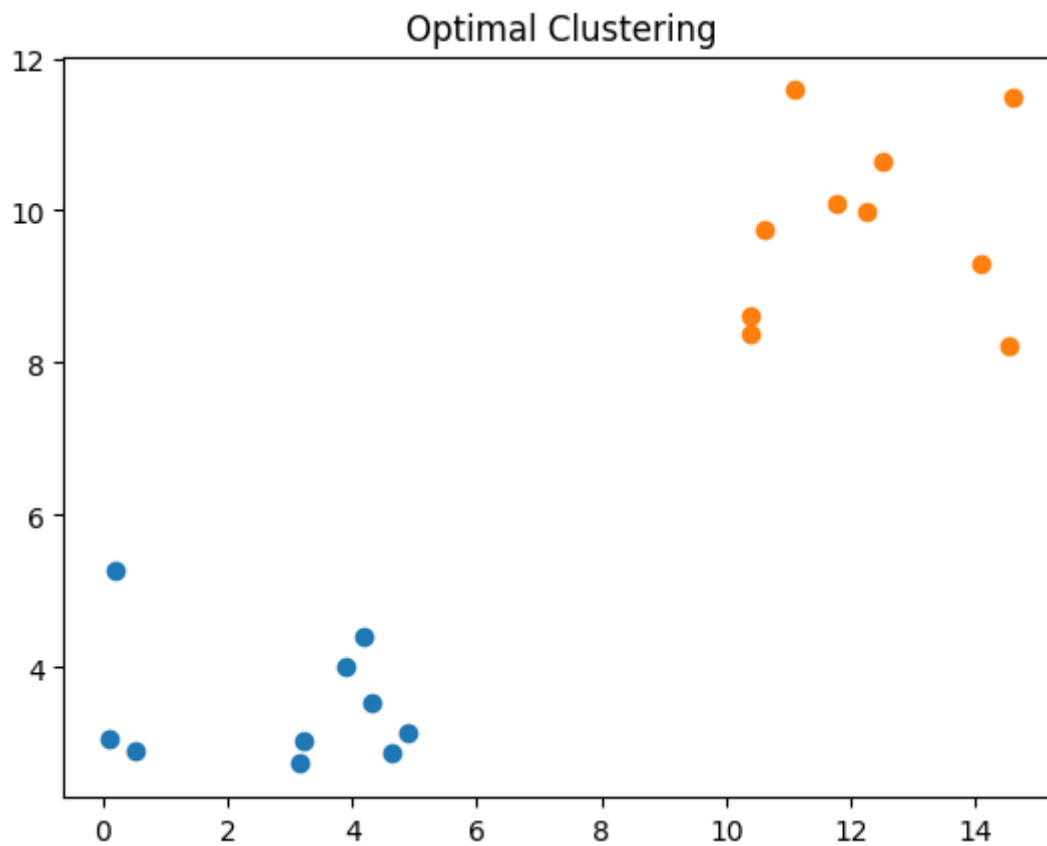
```
c1 = numpy.array([cluster1_x1, cluster1_x2]).T
c2 = numpy.array([cluster2_x1, cluster2_x2]).T
data = numpy.concatenate((c1, c2), axis=0)
data
```

Out[22]:

```
array([[ 0.50949723,  2.91345707],
       [ 0.19456949,  5.2697624 ],
       [ 3.22770048,  3.04365057],
       [ 3.89160338,  4.00513141],
       [ 3.15968591,  2.73358779],
       [ 4.88288691,  3.14010266],
       [ 4.19386993,  4.41302948],
       [ 0.09931089,  3.05336731],
       [ 4.30383156,  3.52316415],
       [ 4.63457454,  2.88616957],
       [11.09995088, 11.58009881],
       [14.09627587,  9.301346 ],
       [10.60263389,  9.74093018],
       [12.26060415,  9.98280478],
       [14.53026758,  8.22504215],
       [10.39399529,  8.38433426],
       [11.76317848, 10.08226297],
       [14.59623416, 11.48679457],
       [10.39743383,  8.60678157],
       [12.51600053, 10.63949219]])
```

In [23]:

```
matplotlib.pyplot.scatter(cluster1_x1, cluster1_x2)
matplotlib.pyplot.scatter(cluster2_x1, cluster2_x2)
matplotlib.pyplot.title("Optimal Clustering")
matplotlib.pyplot.show()
```



In [24]:

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

In [25]:

```
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:feature_vector_length*(clust_idx+1)])
        cluster_center_dists = euclidean_distance(data, cluster_centers[clust_idx])
        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, clusters[clust_idx]]))
    clusters_sum_dist = numpy.array(clusters_sum_dist)
    return cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_dist
```

In [26]:

```
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 [27]:

```
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 [28]:

```
best_solution, best_solution_fitness, best_solution_idx = ga_instance.best_solution()
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.best_solution_idx))
```

Best solution is [12.19391404 10.00056756 3.44634721 3.31312267]

Fitness of the best solution is 0.02933377438440531

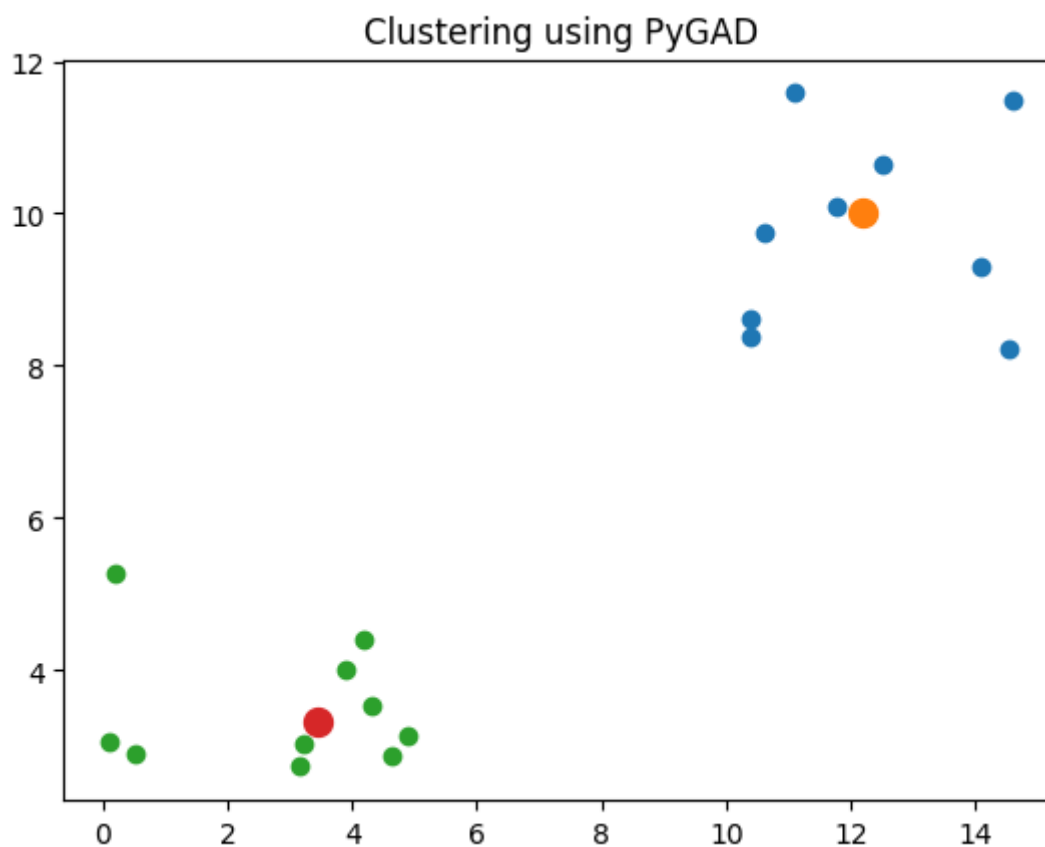
Best solution found after 96 generations

In [31]:

```
cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_dist = clus
```

In [32]:

```
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[cluster_i  
matplotlib.pyplot.title("Clustering using PyGAD")  
matplotlib.pyplot.show()
```



In []: