

In [1]:

```
pip install pygad
```

```
Requirement already satisfied: pygad in c:\users\samit\appdata\local\progr
ams\python\python311\lib\site-packages (3.0.1)
Requirement already satisfied: cloudpickle in c:\users\samit\appdata\loca
l\programs\python\python311\lib\site-packages (from pygad) (2.2.1)
Requirement already satisfied: matplotlib in c:\users\samit\appdata\local
\programs\python\python311\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\samit\appdata\local\prog
rams\python\python311\lib\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\samit\appdata
\local\programs\python\python311\lib\site-packages (from matplotlib->pyga
d) (1.0.7)
Requirement already satisfied: cyclor>=0.10 in c:\users\samit\appdata\loc
al\programs\python\python311\lib\site-packages (from matplotlib->pygad)
(0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\samit\appdat
a\local\programs\python\python311\lib\site-packages (from matplotlib->pyg
ad) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\samit\appdat
a\local\programs\python\python311\lib\site-packages (from matplotlib->pyg
ad) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\samit\appdata
\local\programs\python\python311\lib\site-packages (from matplotlib->pyga
d) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\samit\appdata\lo
cal\programs\python\python311\lib\site-packages (from matplotlib->pygad)
(9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\samit\appdata
\local\programs\python\python311\lib\site-packages (from matplotlib->pyga
d) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\samit\app
data\local\programs\python\python311\lib\site-packages (from matplotlib->
pygad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\samit\appdata\local\p
rograms\python\python311\lib\site-packages (from python-dateutil>=2.7->ma
tplotlib->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_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 [4]:

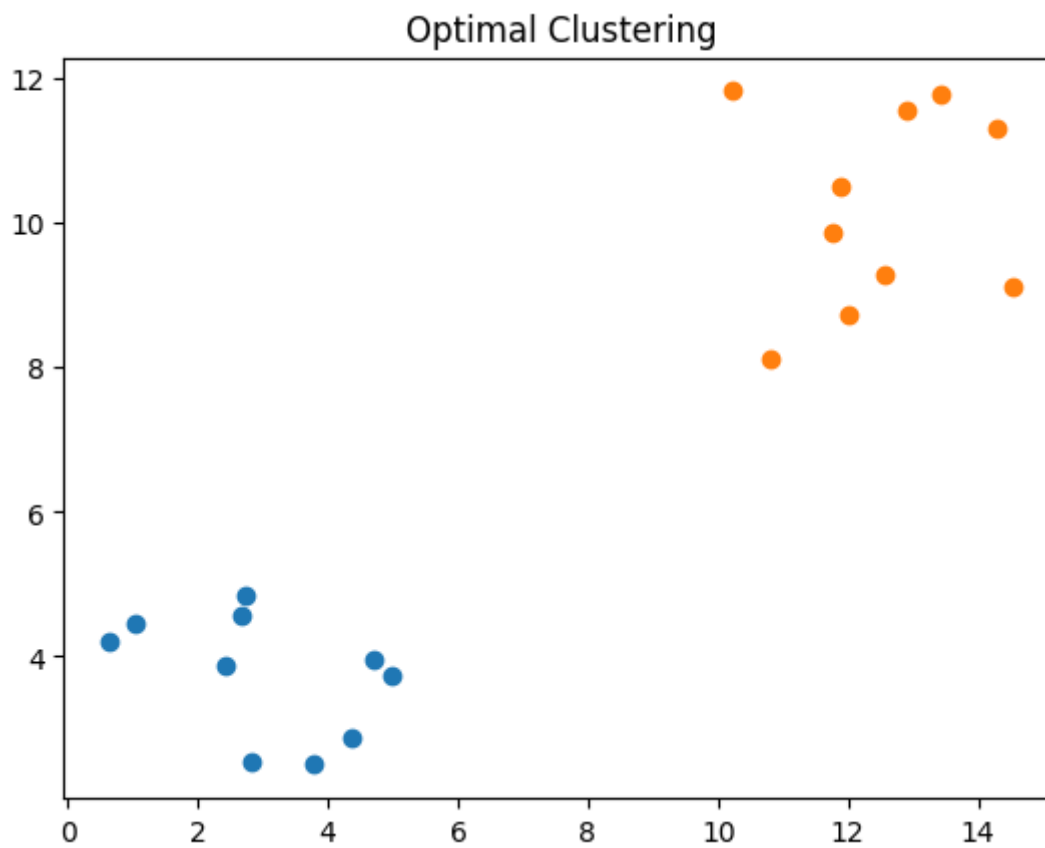
```
c1 = numpy.array([cluster1_x1, cluster1_x2]).T
c2 = numpy.array([cluster2_x1, cluster2_x2]).T
data = numpy.concatenate((c1, c2), axis=0)
data
```

Out[4]:

```
array([[ 2.6838736 ,  4.57052151],
       [ 2.41331132,  3.88530411],
       [ 2.8136292 ,  2.53853439],
       [ 2.7448541 ,  4.84535373],
       [ 4.69299685,  3.94589429],
       [ 4.97118278,  3.73488051],
       [ 1.02867791,  4.4463818 ],
       [ 3.77557119,  2.50280687],
       [ 0.63338203,  4.19461273],
       [ 4.35488691,  2.88900024],
       [10.81254781,  8.10615497],
       [10.21866051, 11.8200728 ],
       [11.74688494,  9.8629622 ],
       [12.02087866,  8.71645395],
       [13.43525957, 11.77682112],
       [12.90228501, 11.54732513],
       [14.29316367, 11.30762417],
       [14.52937306,  9.11093142],
       [11.88079298, 10.51179872],
       [12.56586353,  9.26866743]])
```

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 [28]:

```
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 [29]:

```
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 [30]:

```
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 [31]:

```
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 [12.25046525 10.13126786 2.89180653 3.89482825]

Fitness of the best solution is 0.031193199347366512

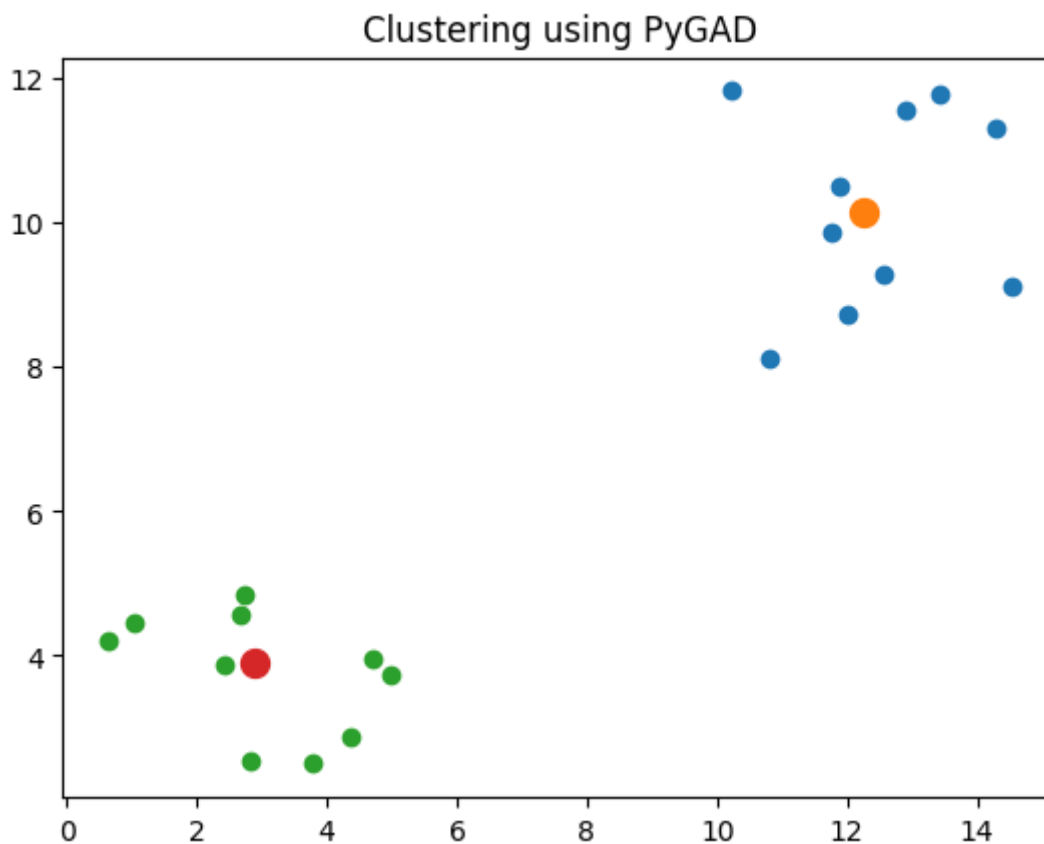
Best solution found after 87 generations

In [32]:

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

In [33]:

```
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_idx, 1])
matplotlib.pyplot.title("Clustering using PyGAD")
matplotlib.pyplot.show()
```



In []:

In []: