In [8]:

```
pip install pygad
```

```
Requirement already satisfied: pygad in c:\users\hp\appdata\local\programs\python\python310\lib\sit
e-packages (3.0.1)
Requirement already satisfied: cloudpickle in c:\users\hp\appdata\local\programs\python\python310\l
ib\site-packages (from pygad) (2.2.1)
Requirement already satisfied: matplotlib in c:\users\hp\appdata\local\programs\python\python310\li
b\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\hp\appdata\local\programs\python\python310\lib\sit
e-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hp\appdata\local\programs\python\python
310\lib\site-packages (from matplotlib->pygad) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\appdata\local\programs\python\python310
\lib\site-packages (from matplotlib->pygad) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hp\appdata\local\programs\python\pytho
n310\lib\site-packages (from matplotlib->pygad) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\appdata\local\programs\python\pytho
n310\lib\site-packages (from matplotlib->pygad) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\hp\appdata\local\programs\python\python3
10\lib\site-packages (from matplotlib->pygad) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hp\appdata\local\programs\python\python310
\lib\site-packages (from matplotlib->pygad) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\hp\appdata\local\programs\python\python
310\lib\site-packages (from matplotlib->pygad) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\appdata\local\programs\python\py
thon310\lib\site-packages (from matplotlib->pygad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\hp\appdata\local\programs\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 [9]:

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

In [10]:

```
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 [11]:

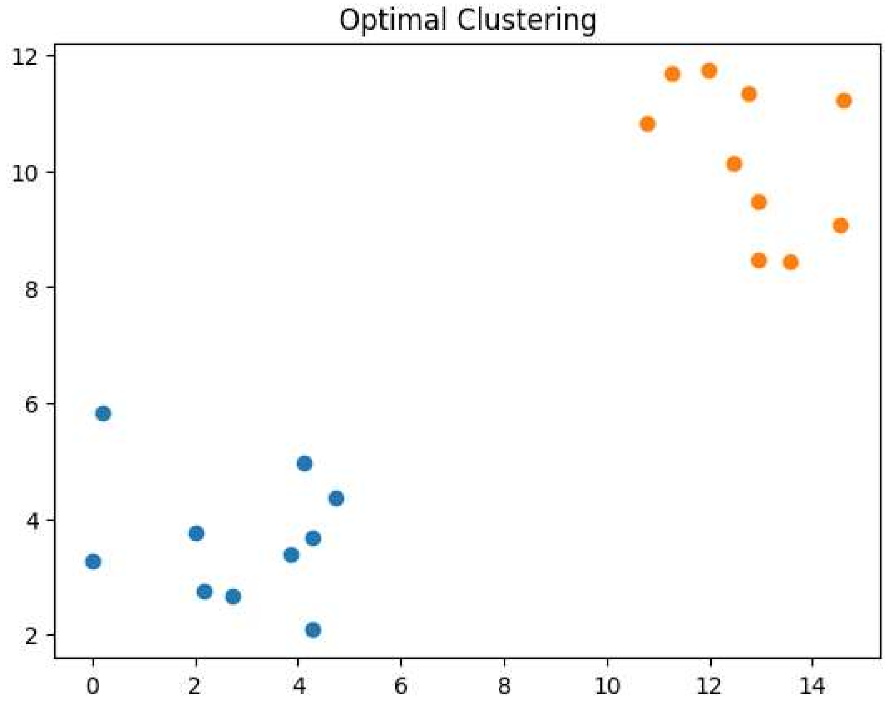
```
c1 = numpy.array([cluster1_x1, cluster1_x2]).T
c2 = numpy.array([cluster2_x1, cluster2_x2]).T
data = numpy.concatenate((c1, c2), axis=0)
data
```

Out[11]:

```
array([[2.17786607e+00, 2.74678615e+00],
       [4.10613462e+00, 4.95969744e+00],
       [4.72016843e+00, 4.35289196e+00],
       [4.27674429e+00, 2.08017132e+00],
       [3.84455791e+00, 3.39670935e+00],
       [4.27982348e+00, 3.68765059e+00],
       [2.12261897e-01, 5.81804898e+00],
       [3.79289173e-03, 3.26432130e+00],
       [2.00258908e+00, 3.76468366e+00],
       [2.73779903e+00, 2.66072426e+00],
       [1.45948198e+01, 1.12275550e+01],
       [1.29529704e+01, 9.48745437e+00],
       [1.24814130e+01, 1.01374201e+01],
       [1.27562151e+01, 1.13300084e+01],
       [1.07666886e+01, 1.08355134e+01],
       [1.12590135e+01, 1.16812355e+01],
       [1.29363324e+01, 8.47821836e+00],
       [1.35532864e+01, 8.42881536e+00],
       [1.45342781e+01, 9.06352838e+00],
       [1.19810464e+01, 1.17324677e+01]])
```

In [12]:

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



In [13]:

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

In [16]:

```
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 [24]:

```

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_
        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]) == clusters_sum_dist.append(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_generation)
```

```
Best solution is [12.58887079 10.16829963  3.08277653  3.46844113]
Fitness of the best solution is 0.029990016601585262
Best solution found after 79 generations
```

In [29]:

```
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_generation))
```

```
Best solution is [12.58887079 10.16829963  3.08277653  3.46844113]
Fitness of the best solution is 0.029990016601585262
Best solution found after 79 generations
```

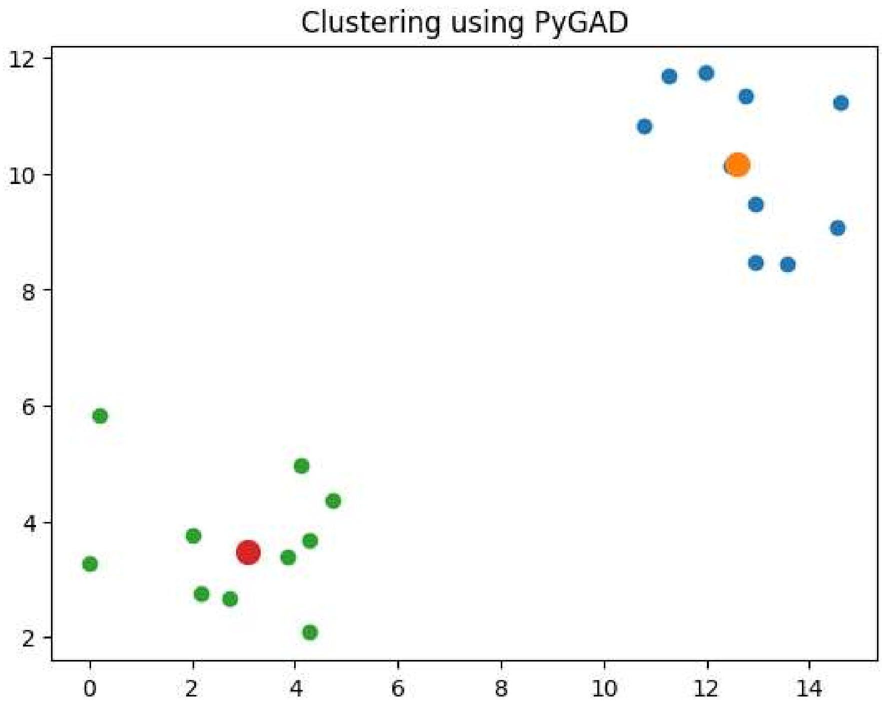
In [31]:

```
cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_dist= cluster_data(best_
```

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



In [ ]: