In [3]:

```
pip install pygad
```

```
Requirement already satisfied: pygad in c:\users\dell e5490\appdata\local
\programs\python\python310\lib\site-packages (3.0.1)
Requirement already satisfied: cloudpickle in c:\users\dell e5490\appdata
\local\programs\python\python310\lib\site-packages (from pygad) (2.2.1)
Requirement already satisfied: matplotlib in c:\users\dell e5490\appdata
\local\programs\python\python310\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\dell e5490\appdata\local
\programs\python\python310\lib\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\dell e5490\ap
pdata\local\programs\python\python310\lib\site-packages (from matplotlib-
>pygad) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\dell e5490\appdat
a\local\programs\python\python310\lib\site-packages (from matplotlib->pyg
ad) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell e5490\a
ppdata\local\programs\python\python310\lib\site-packages (from matplotlib
->pygad) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell e5490\a
ppdata\local\programs\python\python310\lib\site-packages (from matplotlib
->pygad) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\dell e5490\app
data\local\programs\python\python310\lib\site-packages (from matplotlib->
pygad) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell e5490\appda
ta\local\programs\python\python310\lib\site-packages (from matplotlib->py
gad) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\dell e5490\ap
pdata\local\programs\python\python310\lib\site-packages (from matplotlib-
>pygad) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell e549
0\appdata\local\programs\python\python310\lib\site-packages (from matplot
lib->pygad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\dell e5490\appdata\lo
cal\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 [4]:

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

In [5]:

```
 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 [6]:

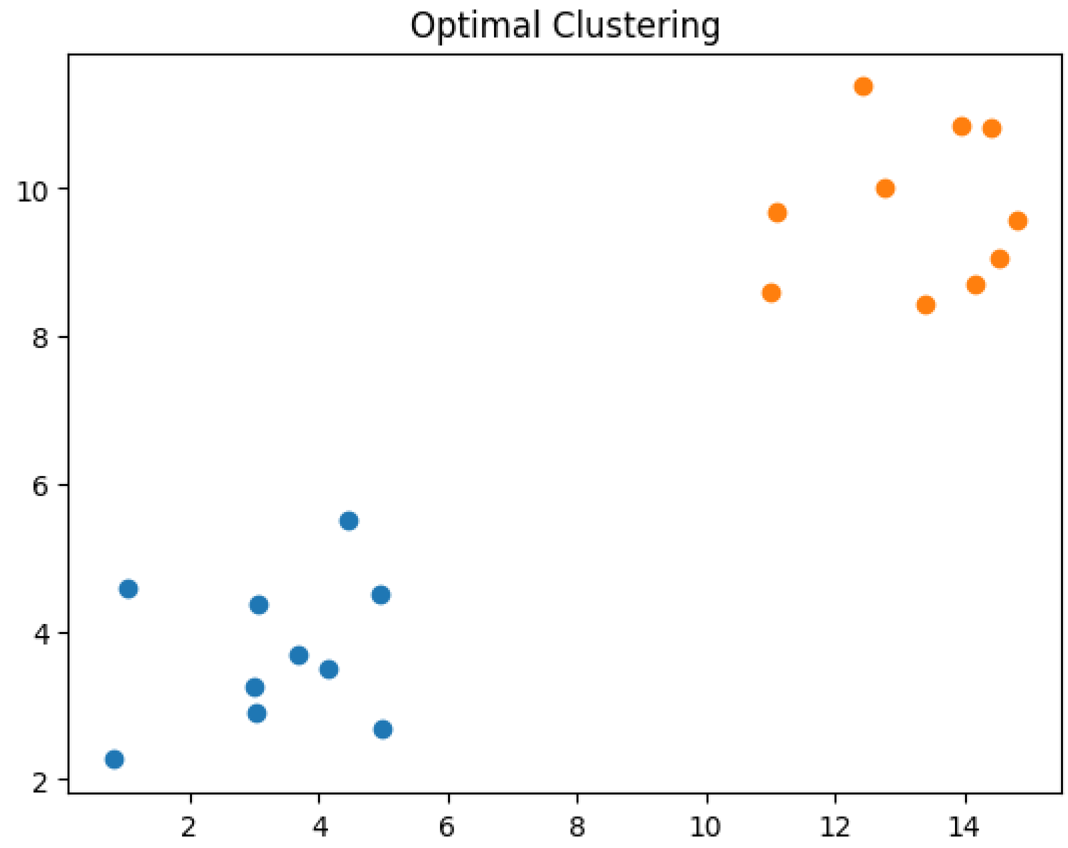
```
c1 = numpy.array([cluster1_x1, cluster1_x2]).T
c2 = numpy.array([cluster2_x1, cluster2_x2]).T
data = numpy.concatenate((c1, c2), axis=0)
data
```

Out[6]:

```
array([[ 3.06780728,  4.36609253],
       [ 4.98751565,  2.68334613],
       [ 4.16009145,  3.50008452],
       [ 3.69358275,  3.68010142],
       [ 3.01926484,  2.91123053],
       [ 4.44961828,  5.52214488],
       [ 0.82448464,  2.26968245],
       [ 1.05613427,  4.59957322],
       [ 4.96185169,  4.50192492],
       [ 3.00064632,  3.25593512],
       [12.77073021, 10.023862  ],
       [14.42606456, 10.8164137 ],
       [14.55234514,  9.05467048],
       [13.40448841,  8.43690346],
       [14.8127191 ,  9.56854161],
       [11.08973875,  9.67707566],
       [14.17177941,  8.70731751],
       [13.96269552, 10.86508633],
       [11.01700637,  8.61246333],
       [12.41574118, 11.38123536]])
```

In [7]:

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



In [8]:

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

In [9]:

```
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_l
        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[cl

    clusters_sum_dist = numpy.array(clusters_sum_dist)

    return cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_
```

In [10]:

```
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 [11]:

```
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 [12]:

```
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_solutior
```

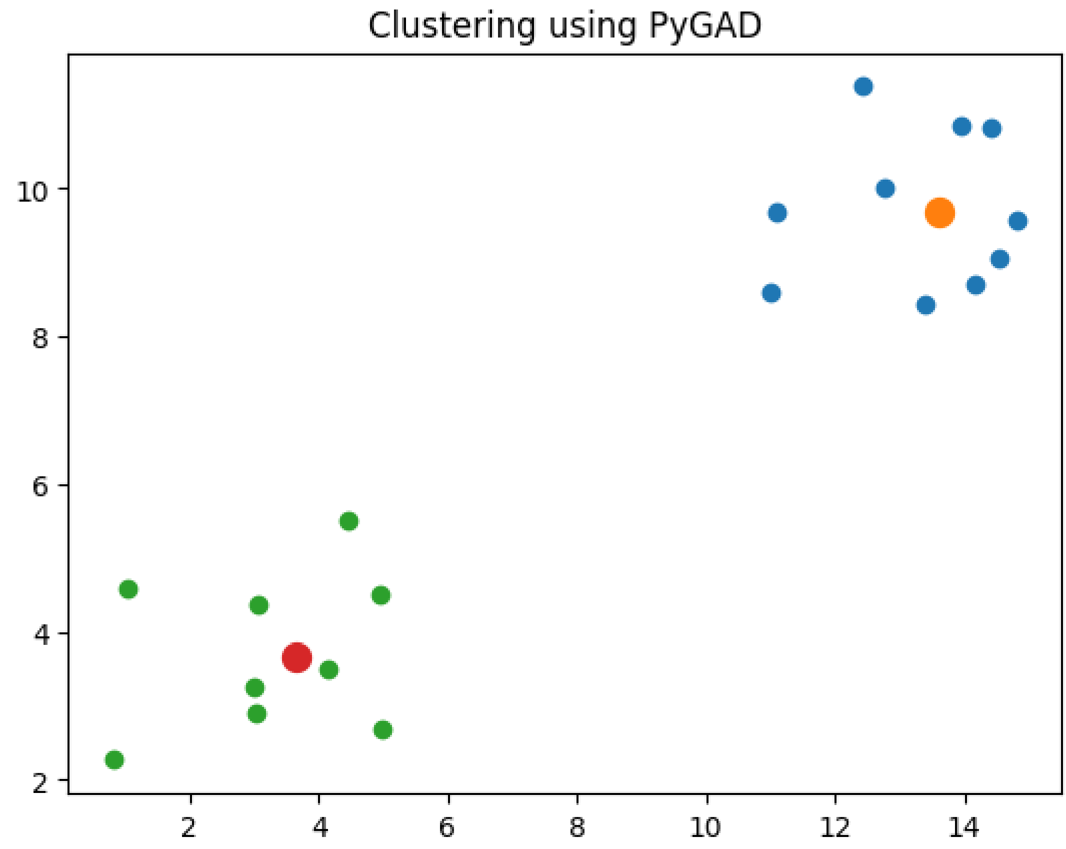
```
Best solution is [13.60312167  9.69497446  3.66016916  3.65565938]
Fitness of the best solution is 0.03326892869724182
Best solution found after 100 generations
```

In [13]:

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

In [14]:

```
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()
```


Clustering using PyGAD