In [1]:

```
1  pip install pygad
```

```
Collecting pygad
  Downloading pygad-3.0.1-py3-none-any.whl (67 kB)
                                          0.0/68.0 kB ? eta -:--:--
     ------------------                   30.7/68.0 kB ? eta -:--:--
     ---------------------------------    61.4/68.0 kB 825.8 kB/s eta 0:00:01
     ---------------------------------    61.4/68.0 kB 825.8 kB/s eta 0:00:01
     ------------------------------------ 68.0/68.0 kB 370.0 kB/s eta 0:00:00
Collecting cloudpickle (from pygad)
  Downloading cloudpickle-2.2.1-py3-none-any.whl (25 kB)
Requirement already satisfied: matplotlib in c:\users\hp\appdata\local\programs\python\pyt
hon311\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\hp\appdata\local\programs\python\python31
1\lib\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hp\appdata\local\programs\pyth
on\python311\lib\site-packages (from matplotlib->pygad) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\appdata\local\programs\python\p
ython311\lib\site-packages (from matplotlib->pygad) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hp\appdata\local\programs\pyt
hon\python311\lib\site-packages (from matplotlib->pygad) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\appdata\local\programs\pyt
hon\python311\lib\site-packages (from matplotlib->pygad) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\hp\appdata\local\programs\pytho
n\python311\lib\site-packages (from matplotlib->pygad) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hp\appdata\local\programs\python
\python311\lib\site-packages (from matplotlib->pygad) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\hp\appdata\local\programs\pyth
on\python311\lib\site-packages (from matplotlib->pygad) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\appdata\local\programs
\python\python311\lib\site-packages (from matplotlib->pygad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\hp\appdata\local\programs\python\pytho
n311\lib\site-packages (from python-dateutil>=2.7->matplotlib->pygad) (1.16.0)
Installing collected packages: cloudpickle, pygad
Successfully installed cloudpickle-2.2.1 pygad-3.0.1
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
1  import numpy
2  import matplotlib.pyplot
3  import pygad
```

In [3]:

```
1   cluster1_num_samples = 10
2   cluster1_x1_start = 0
3   cluster1_x1_end = 5
4   cluster1_x2_start = 2
5   cluster1_x2_end = 6
6   cluster1_x1 = numpy.random.random(size=(cluster1_num_samples))
7   cluster1_x1 = cluster1_x1 * (cluster1_x1_end - cluster1_x1_start) + cluster1_x1_start
8   cluster1_x2 = numpy.random.random(size=(cluster1_num_samples))
9   cluster1_x2 = cluster1_x2 * (cluster1_x2_end - cluster1_x2_start) + cluster1_x2_start
10  cluster2_num_samples = 10
11  cluster2_x1_start = 10
12  cluster2_x1_end = 15
13  cluster2_x2_start = 8
14  cluster2_x2_end = 12
15  cluster2_x1 = numpy.random.random(size=(cluster2_num_samples))
16  cluster2_x1 = cluster2_x1 * (cluster2_x1_end - cluster2_x1_start) + cluster2_x1_start
17  cluster2_x2 = numpy.random.random(size=(cluster2_num_samples))
18  cluster2_x2 = cluster2_x2 * (cluster2_x2_end - cluster2_x2_start) + cluster2_x2_start
```

In [4]:

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

Out[4]:

```
array([[ 4.1286135 ,  5.97128236],
       [ 1.32235587,  3.37135497],
       [ 1.16874564,  2.19878014],
       [ 3.32295083,  4.9603563 ],
       [ 3.27604046,  5.97977934],
       [ 1.71202013,  5.73469733],
       [ 0.29969341,  5.68571073],
       [ 0.2574191 ,  3.22833903],
       [ 0.87005668,  2.29981287],
       [ 4.6688935 ,  2.96015475],
       [10.24574522,  8.55864665],
       [10.26772964, 11.00698761],
       [10.29550531,  9.24603424],
       [13.99091308,  8.25549998],
       [12.14402662,  9.63927854],
       [11.21451388,  8.10565999],
       [11.44068358,  8.52025061],
       [13.81821889,  9.12265223],
       [12.61619651,  9.12402537],
       [11.63989626,  9.80798506]])
```

In [5]:

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

In [9]:
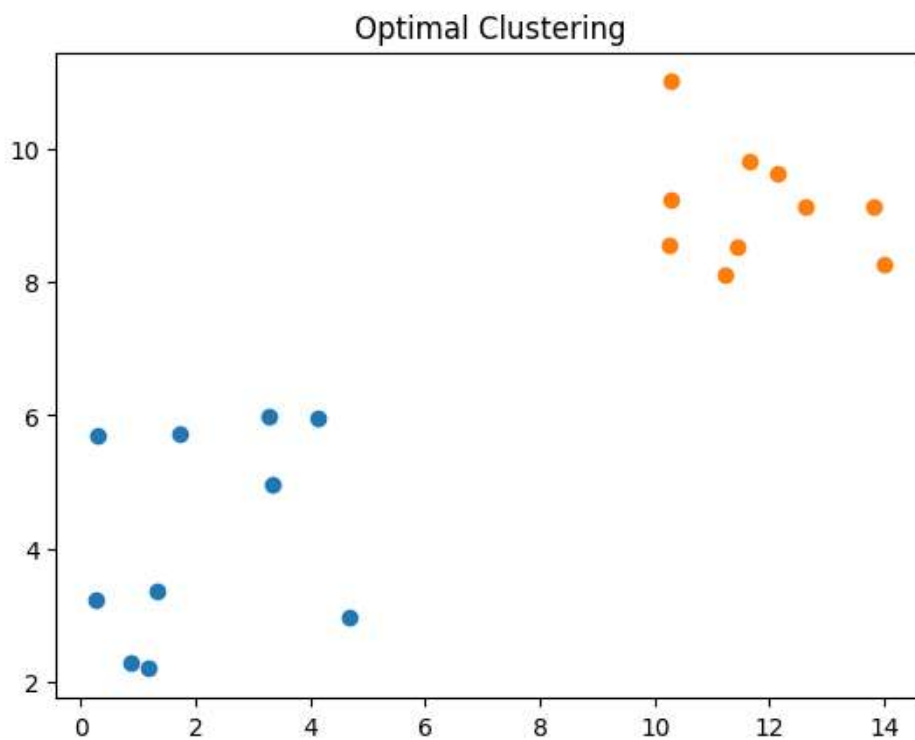
```python
def euclidean_distance(X, Y):
    return numpy.sqrt(numpy.sum(numpy.power(X - Y, 2), axis=1))
```

In [16]:

```python
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_id:
      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 [17]:

```python
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 [18]:

```python
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 [19]:

```python
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)
```
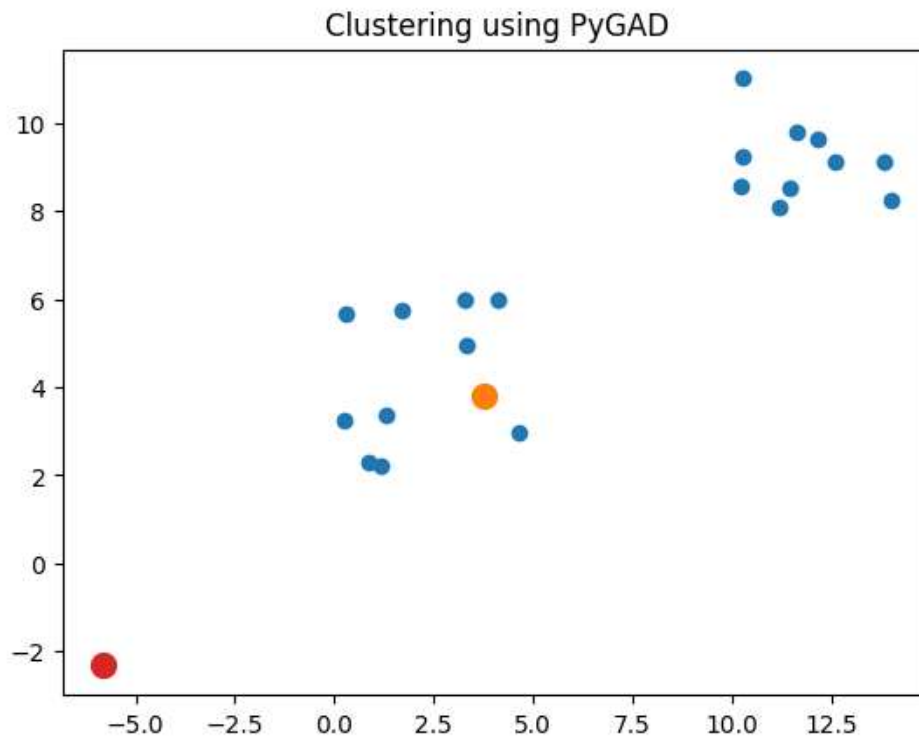
```
Best solution is [ 3.74919632  3.81106558 -5.83151371 -2.33031057]
Fitness of the best solution is 100000000.0
Best solution found after 2 generations
```

In [22]:

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

In [23]:

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


In [ ]:

```

```