In [29]:
```
pip install pygad
```

```
Requirement already satisfied: pygad in c:\users\sneha\appdata\local\programs\python\python310\lib
\site-packages (3.0.1)
Requirement already satisfied: cloudpickle in c:\users\sneha\appdata\local\programs\python\python31
0\lib\site-packages (from pygad) (2.2.1)
Requirement already satisfied: matplotlib in c:\users\sneha\appdata\local\programs\python\python310
\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\sneha\appdata\local\programs\python\python310\lib
\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\sneha\appdata\local\programs\python\pyt
hon310\lib\site-packages (from matplotlib->pygad) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\sneha\appdata\local\programs\python\python3
10\lib\site-packages (from matplotlib->pygad) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\sneha\appdata\local\programs\python\py
thon310\lib\site-packages (from matplotlib->pygad) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\sneha\appdata\local\programs\python\py
thon310\lib\site-packages (from matplotlib->pygad) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\sneha\appdata\local\programs\python\pyth
on310\lib\site-packages (from matplotlib->pygad) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\sneha\appdata\local\programs\python\python
310\lib\site-packages (from matplotlib->pygad) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\sneha\appdata\local\programs\python\pyt
hon310\lib\site-packages (from matplotlib->pygad) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\sneha\appdata\local\programs\python
\python310\lib\site-packages (from matplotlib->pygad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\sneha\appdata\local\programs\python\python310\l
ib\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 [66]:
```
import numpy
import matplotlib.pyplot
import pygad
```

In [67]:
```
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 [68]:
```
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[68]:
```
array([[ 3.48849071,  3.68321537],
       [ 0.89410827,  3.69865801],
       [ 1.68170152,  5.04202618],
       [ 1.03300509,  2.36098098],
       [ 0.14806579,  3.89760571],
       [ 1.13666516,  5.5644725 ],
       [ 3.81731246,  4.54850518],
       [ 3.3633942 ,  3.44683587],
       [ 3.78451704,  4.36554203],
       [ 1.70078058,  5.07983265],
       [13.70435955,  8.94556123],
       [14.20046337,  8.26087346],
       [13.06530403, 10.31295935],
       [11.91818294, 11.34460312],
       [14.80237152,  9.47112271],
       [12.24809216,  9.26136598],
       [11.10420154, 10.48975057],
       [10.19032728, 11.80799911],
       [12.77990065,  8.44551683],
       [11.25349125,  8.5560894 ]])
```

In [69]:
```
matplotlib.pyplot.scatter(cluster1_x1, cluster1_x2)
matplotlib.pyplot.scatter(cluster2_x1, cluster2_x2)
matplotlib.pyplot.title("Optimal Clustering")
matplotlib.pyplot.show()
```

In [70]:
```
def euclidean_distance(X, Y):
    return numpy.sqrt(numpy.sum(numpy.power(X - Y, 2), axis=1))
```

In [71]:
```
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]) == 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_
```

In [72]:
```
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 [73]:
```
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 [74]:
```
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))
```

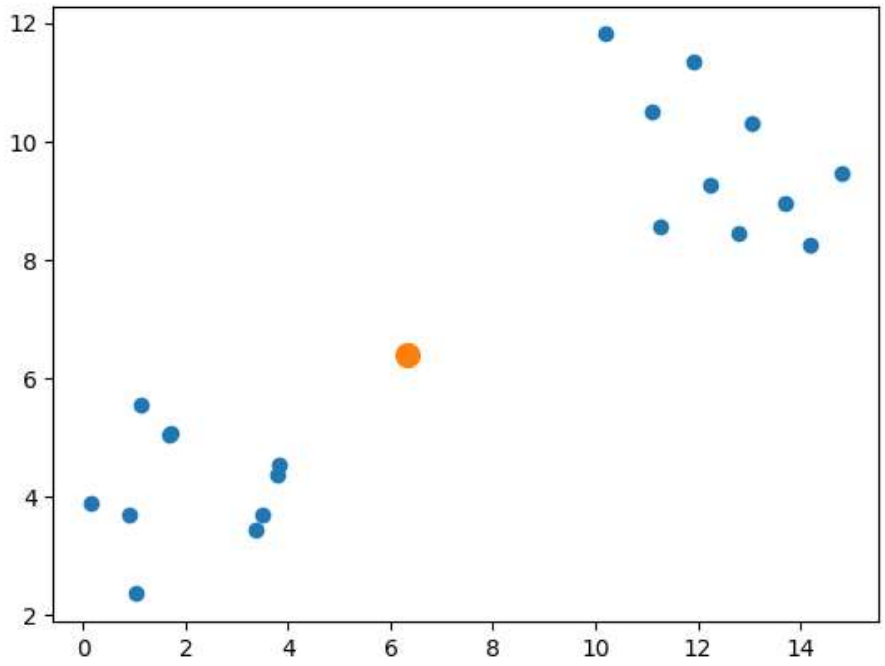
```
Best solution is [6.33364892 6.39789799 7.42711964 4.19925316]
Fitness of the best solution is 0.008291100564176328
Best solution found after 91 generations
```

In [75]:
```
cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_dist=
  cluster_data(best_solution, best_solution_idx)
```

In [78]:
```
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()
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[78], line 2
      1 for cluster_idx in range(num_clusters):
----> 2     cluster_x = data[clusters[cluster_idx], 0]
      3     cluster_y = data[clusters[cluster_idx], 1]
      4     matplotlib.pyplot.scatter(cluster_x, cluster_y)

IndexError: list index out of range
```



In [ ]:
```

```