In [15]:
```
pip install pygad
```

```
Requirement already satisfied: pygad in c:\users\jangidi veena\appdata\local
\programs\python\python311\lib\site-packages (3.0.1)
Requirement already satisfied: cloudpickle in c:\users\jangidi veena\appdata
\local\programs\python\python311\lib\site-packages (from pygad) (2.2.1)
Requirement already satisfied: matplotlib in c:\users\jangidi veena\appdata\l
ocal\programs\python\python311\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\jangidi veena\appdata\local
\programs\python\python311\lib\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\jangidi veena\app
data\local\programs\python\python311\lib\site-packages (from matplotlib->pyga
d) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\jangidi veena\appdata
\local\programs\python\python311\lib\site-packages (from matplotlib->pygad)
(0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\jangidi veena\ap
pdata\local\programs\python\python311\lib\site-packages (from matplotlib->pyg
ad) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\jangidi veena\ap
pdata\local\programs\python\python311\lib\site-packages (from matplotlib->pyg
ad) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\jangidi veena\appd
ata\local\programs\python\python311\lib\site-packages (from matplotlib->pyga
d) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\jangidi veena\appdat
a\local\programs\python\python311\lib\site-packages (from matplotlib->pygad)
(9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\jangidi veena\app
data\local\programs\python\python311\lib\site-packages (from matplotlib->pyga
d) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\jangidi veena
\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->
pygad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\jangidi veena\appdata\loc
al\programs\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.
```

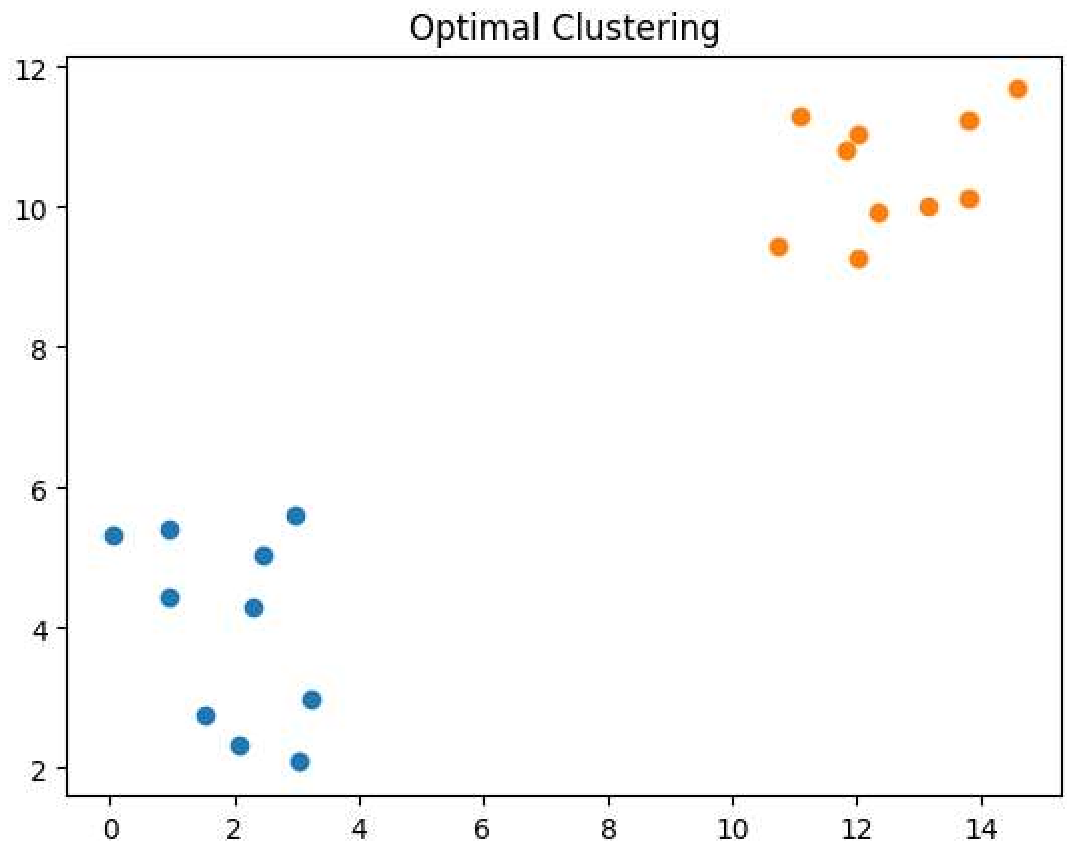
In [16]:
```
import numpy
import matplotlib.pyplot
import pygad
```

In [17]:
```
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_x
cluster1_x2 = numpy.random.random(size=(cluster1_num_samples))
cluster1_x2 = cluster1_x2 * (cluster1_x2_end - cluster1_x2_start) + cluster1_x
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_x
cluster2_x2 = numpy.random.random(size=(cluster2_num_samples))
cluster2_x2 = cluster2_x2 * (cluster2_x2_end - cluster2_x2_start) + cluster2_x
```

In [18]:
```
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[18]:
```
array([[ 2.09336441,  2.32772112],
       [ 0.04897798,  5.33355646],
       [ 3.04636592,  2.0860899 ],
       [ 2.29398181,  4.29948915],
       [ 0.95788618,  4.42940086],
       [ 2.45513308,  5.02936507],
       [ 1.54066568,  2.75071832],
       [ 3.24903361,  2.98535038],
       [ 0.9420665 ,  5.41600818],
       [ 2.98378154,  5.6005111 ],
       [13.78695107, 10.12219975],
       [14.55941508, 11.68440148],
       [11.07687602, 11.30541696],
       [12.01714075, 11.03487982],
       [13.78859359, 11.25259296],
       [12.35691852,  9.93746698],
       [12.01981487,  9.26192305],
       [13.14232466, 10.02667021],
       [11.84645729, 10.80740673],
       [10.72437936,  9.43289626]])
```

In [19]:
```
matplotlib.pyplot.scatter(cluster1_x1, cluster1_x2)
matplotlib.pyplot.scatter(cluster2_x1, cluster2_x2)
matplotlib.pyplot.title("Optimal Clustering")
matplotlib.pyplot.show()
```



In [20]:
```
def euclidean_distance(X, Y):
    return numpy.sqrt(numpy.sum(numpy.power(X - Y, 2), axis=1))
```

In [21]:
```
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:featur
        cluster_center_dists = euclidean_distance(data, cluster_centers[clust_
        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, c
    clusters_sum_dist = numpy.array(clusters_sum_dist)
    return cluster_centers, all_clusters_dists, cluster_indices, clusters, clu
```

In [22]:
```
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 [23]:
```
 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 [24]:
```
best_solution, best_solution_fitness, best_solution_idx = ga_instance.best_sol
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.bes
```

```
Best solution is [12.3875381  10.38812195  1.97370938  4.13562247]
Fitness of the best solution is 0.035164760402395556
Best solution found after 87 generations
```

In [25]:
```
cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_d
```

In [ ]:
```
```