Genetic Algorithm

In [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 544.7 kB/s eta 0:
00:01
     ---------------------------------------- 68.0/68.0 kB 524.7 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\thara\appdata\local
\programs\python\python310\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\thara\appdata\local\progr
ams\python\python310\lib\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\thara\appdata
\local\programs\python\python310\lib\site-packages (from matplotlib->pyga
d) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\thara\appdata\loca
l\programs\python\python310\lib\site-packages (from matplotlib->pygad) (0.
11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\thara\appdata
\local\programs\python\python310\lib\site-packages (from matplotlib->pyga
d) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\thara\appdata
\local\programs\python\python310\lib\site-packages (from matplotlib->pyga
d) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\thara\appdata\l
ocal\programs\python\python310\lib\site-packages (from matplotlib->pygad)
(23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\thara\appdata\loc
al\programs\python\python310\lib\site-packages (from matplotlib->pygad)
(9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\thara\appdata
\local\programs\python\python310\lib\site-packages (from matplotlib->pyga
d) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\thara\appd
ata\local\programs\python\python310\lib\site-packages (from matplotlib->py
gad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\thara\appdata\local\pr
ograms\python\python310\lib\site-packages (from python-dateutil>=2.7->matp
lotlib->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 [3]:

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

In [4]:

```
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 [5]:

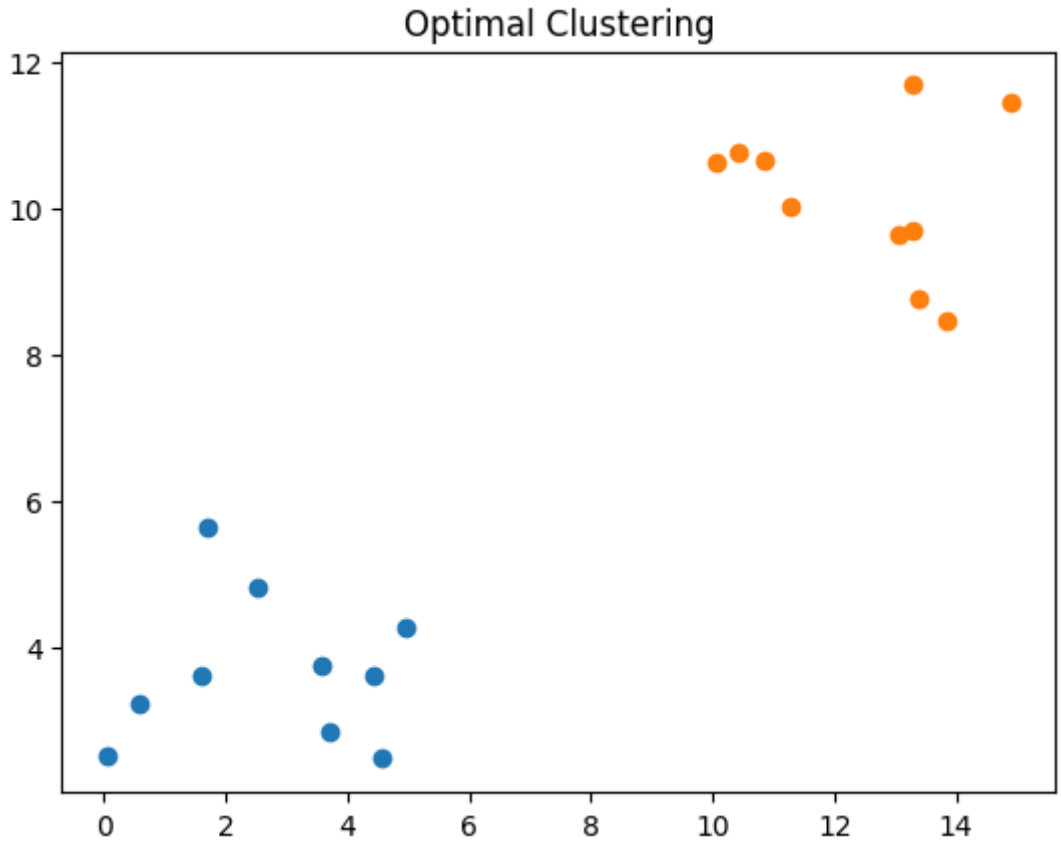
```
c1 = numpy.array([cluster1_x1, cluster1_x2]).T
c2 = numpy.array([cluster2_x1, cluster2_x2]).T
data = numpy.concatenate((c1, c2), axis=0)
data
```

Out[5]:

```
array([[ 3.70266961,  2.83282492],
       [ 0.59269819,  3.21946049],
       [ 1.59646192,  3.61395257],
       [ 4.43382235,  3.60213884],
       [ 3.56347006,  3.75351308],
       [ 0.05007531,  2.50188659],
       [ 2.5261812 ,  4.80203719],
       [ 1.6960587 ,  5.64704178],
       [ 4.95488368,  4.26878318],
       [ 4.56953517,  2.46767506],
       [13.06903331,  9.64160668],
       [13.28338105,  9.70886774],
       [13.84671167,  8.46380211],
       [13.38303252,  8.76429595],
       [11.29140386, 10.0159489 ],
       [10.04835843, 10.62525832],
       [10.86216901, 10.66704858],
       [14.89497663, 11.44974596],
       [10.42469679, 10.75956355],
       [13.28672031, 11.68482915]])
```

In [6]:

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



In [31]:

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

In [38]:

```
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 [39]:

```
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 [40]:

```
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 [41]:

```
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
```

```
Best solution is [12.82484474  9.80055445  3.20714229  3.68628991]
Fitness of the best solution is 0.028528889605355393
Best solution found after 96 generations
```

In [42]:

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

In [43]:

```
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()
```