Genetic Algorithm

In [1]:

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

In [2]:

```
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 [3]:

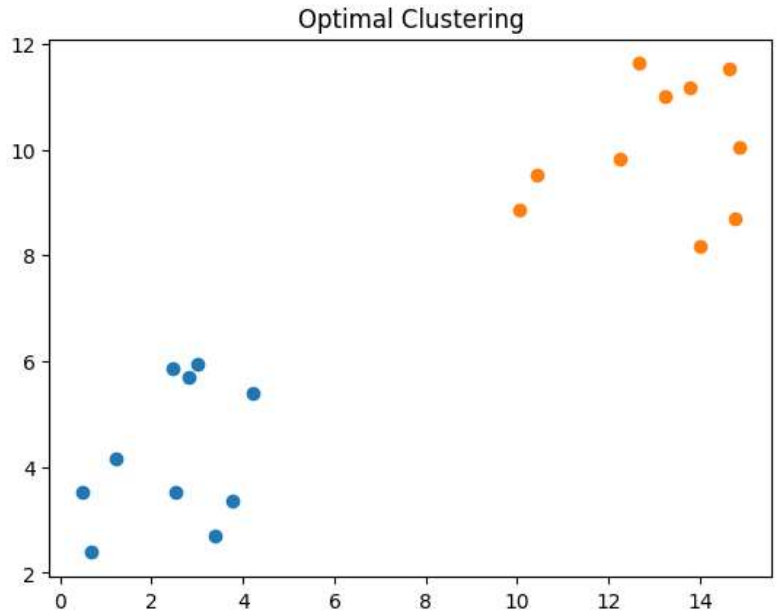
```
c1 = numpy.array([cluster1_x1, cluster1_x2]).T
c2 = numpy.array([cluster2_x1, cluster2_x2]).T
data = numpy.concatenate((c1, c2), axis=0)
data
```

Out[3]:

```
array([[ 3.00264221,  5.93375455],
       [ 0.48327329,  3.53363056],
       [ 3.7558403 ,  3.35963136],
       [ 4.22567701,  5.40562058],
       [ 3.39676668,  2.70150515],
       [ 2.46920386,  5.86589617],
       [ 2.80689432,  5.69472621],
       [ 0.66670969,  2.38349588],
       [ 1.21713382,  4.16657262],
       [ 2.52911449,  3.51467418],
       [13.23443828, 11.01830549],
       [10.05447313,  8.85781984],
       [12.25514676,  9.82347582],
       [12.65726058, 11.62787966],
       [14.76452837,  8.69395335],
       [14.64744415, 11.53776307],
       [13.77283963, 11.18043033],
       [10.43192971,  9.52419836],
       [14.85815936, 10.05367599],
       [13.9916268 ,  8.18256824]])
```

In [4]:

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



In [5]:

```
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_length*(clust_idx+1)])
        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]):
          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 [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_solution_generation))
```

```
Best solution is [-4.32091757  3.50939899  0.50759003 14.91881954]
Fitness of the best solution is 100000000.0
Best solution found after 0 generations
```

In [13]:

```
ıster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_dist= cluster_data(best_solution,best_solution_idx)
```

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_idx,1],linewidths=5)
matplotlib.pyplot.title("Clustering using PyGAD")
matplotlib.pyplot.show()
```