

In [1]: `pip install pygad`

```
Requirement already satisfied: pygad in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (3.0.1)
Requirement already satisfied: cloudpickle in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from pygad) (2.2.1)
Requirement already satisfied: matplotlib in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\dell\appdata\local\programs\python\python311\lib\site-packages (from python-dateutil>=2.7->matplotlib->pygad) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip available: 22.3.1 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [2]: `import numpy`  
`import matplotlib.pyplot`  
`import pygad`

```

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

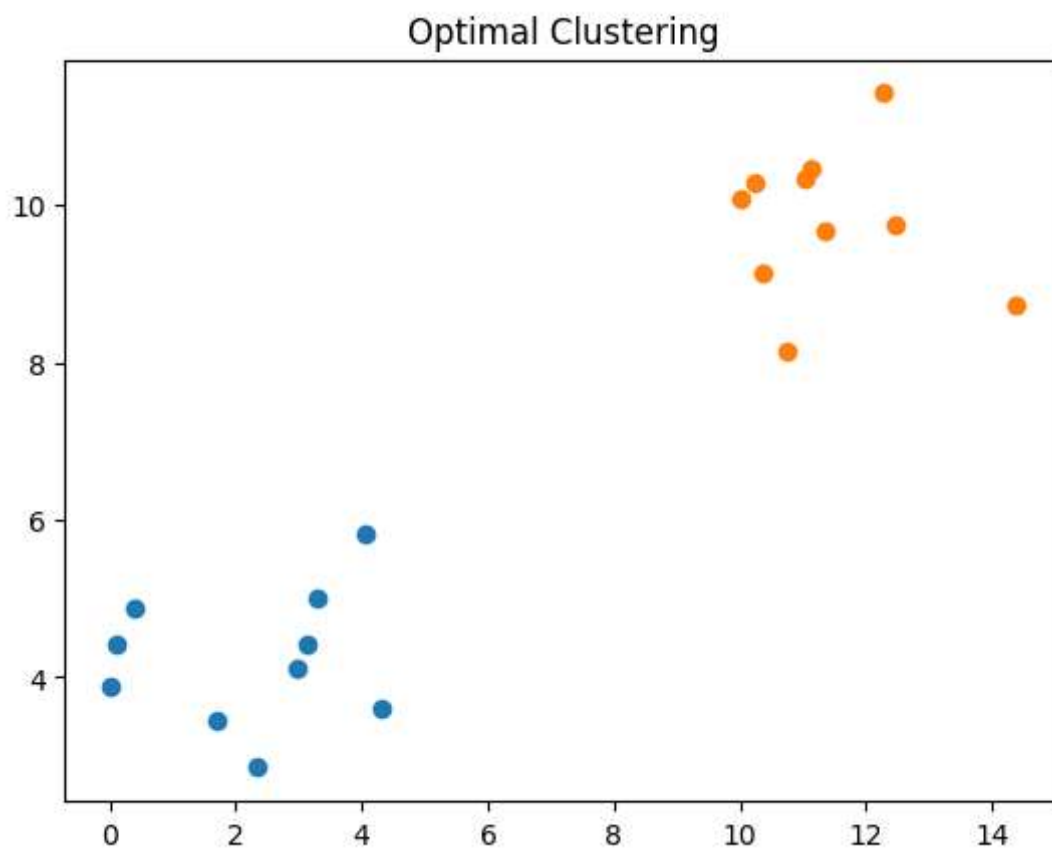
```

```

Out[4]: array([[3.29237477e+00, 5.00237458e+00],
 [1.04957564e-01, 4.42647677e+00],
 [3.13882344e+00, 4.42084035e+00],
 [4.05157884e+00, 5.81874439e+00],
 [4.31573033e+00, 3.59463565e+00],
 [2.97819542e+00, 4.10509704e+00],
 [2.35028454e+00, 2.84950298e+00],
 [3.92206673e-01, 4.86956710e+00],
 [1.07407886e-02, 3.89365743e+00],
 [1.70630086e+00, 3.46008940e+00],
 [1.02226242e+01, 1.02905284e+01],
 [1.43682020e+01, 8.72110833e+00],
 [1.13402235e+01, 9.66156599e+00],
 [1.03730843e+01, 9.14054450e+00],
 [1.22677625e+01, 1.14213582e+01],
 [1.10450243e+01, 1.03348922e+01],
 [1.24621889e+01, 9.76195779e+00],
 [1.00131051e+01, 1.00691236e+01],
 [1.07299150e+01, 8.14545149e+00],
 [1.11217261e+01, 1.04537969e+01]])

```

```
In [5]: matplotlib.pyplot.scatter(cluster1_x1, cluster1_x2)
matplotlib.pyplot.scatter(cluster2_x1, cluster2_x2)
matplotlib.pyplot.title("Optimal Clustering")
matplotlib.pyplot.show()
```



```
In [6]: def euclidean_distance(X, Y):
         return numpy.sqrt(numpy.sum(numpy.power(X - Y, 2), axis=1))
```

```
In [7]: 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]) == 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 [8]: 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 [9]: 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 [10]: 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_idx))
```

```
Best solution is [19.45274657  5.06667068  3.41657283  3.4488444 ]
Fitness of the best solution is 100000000.0
Best solution found after 0 generations
```

```
In [11]: cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_di
```

```
In [12]: 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[  
matplotlib.pyplot.title("Clustering using PyGAD")  
matplotlib.pyplot.show()
```

