

In [1]: `pip install pygad`

Collecting pygadNote: you may need to restart the kernel to use updated packages.

```

    Downloading pygad-3.0.1-py3-none-any.whl (67 kB)
      0.0/68.0 kB ? eta -:--:--
    -----
      30.7/68.0 kB 1.3 MB/s eta 0:00:01
    -----
      68.0/68.0 kB 930.9 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\ubini\appdata\local\programs\python\python311\lib\site-packages (from pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\ubini\appdata\local\programs\python\python311\lib\site-packages (from pygad) (1.24.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\ubini\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (1.0.7)
Requirement already satisfied: cyclor>=0.10 in c:\users\ubini\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\ubini\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\ubini\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\ubini\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\ubini\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\ubini\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\ubini\appdata\local\programs\python\python311\lib\site-packages (from matplotlib->pygad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\ubini\appdata\local\programs\python\python311\lib\site-packages (from python-dateutil>=2.7->matplotlib->pygad) (1.16.0)
Installing collected packages: cloudpickle, pygad
Successfully installed cloudpickle-2.2.1 pygad-3.0.1

```

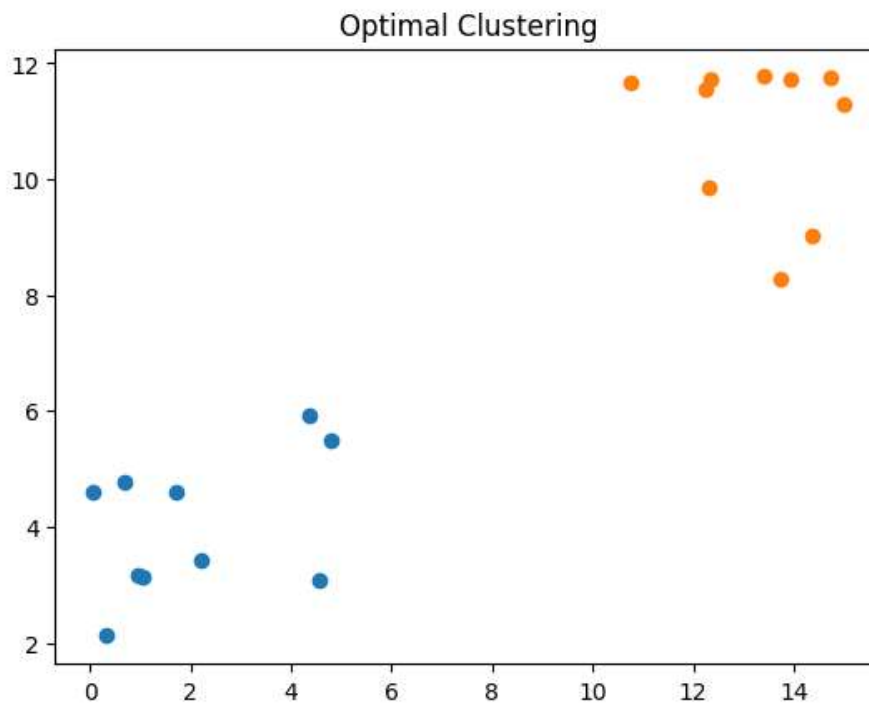
In [5]: `import numpy`
`import matplotlib.pyplot`
`import pygad`

In [7]: `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 [8]: 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[8]: array([[ 1.7106287 ,  4.61640124],
 [ 0.94363899,  3.15789328],
 [ 1.04145253,  3.12464939],
 [ 0.67675425,  4.76999551],
 [ 4.57236211,  3.08361092],
 [ 4.37249143,  5.9251955 ],
 [ 0.31792395,  2.11767171],
 [ 4.78784277,  5.48614836],
 [ 2.20133045,  3.42333392],
 [ 0.05221063,  4.60331804],
 [13.74306532,  8.28066689],
 [13.93054064, 11.73608245],
 [10.77089723, 11.6702638 ],
 [12.23788283, 11.54717854],
 [14.99075065, 11.30982646],
 [12.33571727, 11.73711214],
 [14.36834905,  9.03533083],
 [14.73941955, 11.76121141],
 [13.40019505, 11.77048339],
 [12.31769618,  9.87264091]])
```

```
In [9]: matplotlib.pyplot.scatter(cluster1_x1, cluster1_x2)
matplotlib.pyplot.scatter(cluster2_x1, cluster2_x2)
matplotlib.pyplot.title("Optimal Clustering")
matplotlib.pyplot.show()
```



```
In [10]: def euclidean_distance(X, Y):
return numpy.sqrt(numpy.sum(numpy.power(X - Y, 2), axis=1))
```

```
In [12]: 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 [18]: 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 [20]: 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 [21]: 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 [13.33582167 11.30739946 1.67438545 3.9122935 ]
Fitness of the best solution is 0.02847008220121051
Best solution found after 76 generations
```

```
In [23]: cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_dist= cluster_data(best
```

```
In [26]: 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], line  
matplotlib.pyplot.title("Clustering using PyGAD")  
matplotlib.pyplot.show()
```

