

```
!pip install pygad
import pygad
import numpy
import numpy as np
import matplotlib.pyplot as plt
```

```
def func_3D(x,y):
    wynik= 3
    for i in range(1,4):
        wynik+=x*x - np.cos(2*np.pi*x)
    wynik = -wynik
    return wynik
```

```
x = np.linspace(-4, 4, 100)
y = np.linspace(-4, 4, 100)
X, Y = np.meshgrid(x, y)
Z = func_3D(X,Y)
plt.rcParams["figure.figsize"] = (6,6)
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
surface = ax.plot_surface(X, Y, Z, cmap='coolwarm')
plt.show()
```

```
def fitness_function(ga_instance, solution, solution_idx):
    return func_3D(solution[0],solution[1])
```

```
last_fitness = 0
```

```
def on_generation(ga_instance):
    global last_fitness
    print(f"Generation = {ga_instance.generations_completed}")
    print(f"Fitness = {ga_instance.best_solution(pop_fitness=ga_instance.last_generation_fitness)[1]}")
    print(f"Change = {ga_instance.best_solution(pop_fitness=ga_instance.last_generation_fitness)[1] - last_fitness}")
    print()
    last_fitness = ga_instance.best_solution(pop_fitness=ga_instance.last_generation_fitness)[1]
```

```
ga_instance = pygad.GA(
    num_generations=100,
    num_parents_mating=5,
    sol_per_pop=100,
    num_genes=2,
    init_range_low=-4,
    init_range_high=4,
    fitness_func=fitness_function,
    crossover_type="single_point",
    parent_selection_type = "rws",
    mutation_type = "random",
    mutation_percent_genes=5,
    mutation_by_replacement=True,
    random_mutation_min_val=-4,
    random_mutation_max_val=4,
```

```
)
```

```
print(ga_instance.population)
```

```
ga_instance.run()
```

```
solution, solution_fitness, solution_idx = ga_instance.best_solution()
print(f"Argumenty najlepszego rozw : {solution}")
print(f"Wartosc dla najlepszego rozw: {solution_fitness}")
```

```
def wykres_populacji(ga_instance):
    x = np.linspace(-4, 4, 100)
    y = np.linspace(-4, 4, 100)
```

```
X, Y = np.meshgrid(x, y)
Z = func_3D(X,Y)
plt.contour(X, Y, Z, cmap='coolwarm')
plt.plot(ga_instance.population[:,0], ga_instance.population[:,1], 'bo')
plt.show()
```

```
wykres_populacji(ga_instance)
```

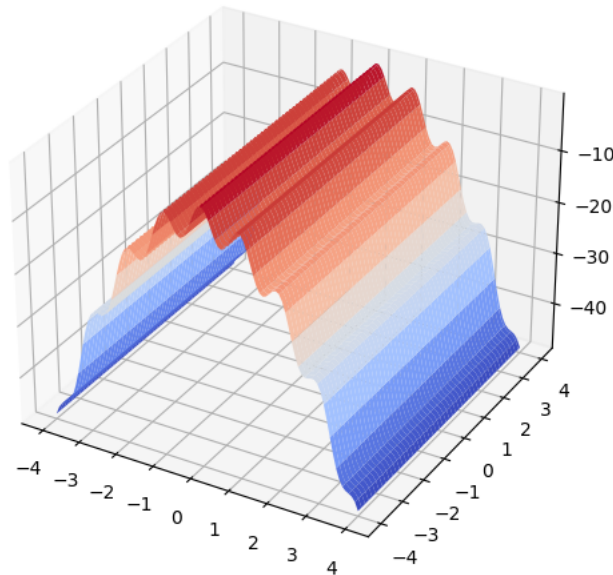


Collecting pygad

Downloading pygad-3.2.0-py3-none-any.whl (80 kB)

80.8/80.8 kB 576.6 kB/s eta 0:00:00

Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-packages (from pygad) (2.2.1)  
 Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from pygad) (3.7.1)  
 Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from pygad) (1.23.5)  
 Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pygad) (1.2.0)  
 Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pygad) (0.12.1)  
 Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pygad) (4.47.2)  
 Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pygad) (1.4.5)  
 Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pygad) (23.2)  
 Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pygad) (9.4.0)  
 Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pygad) (3.1.1)  
 Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pygad) (2.8.2)  
 Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->pygad) (1.16).  
 Installing collected packages: pygad  
 Successfully installed pygad-3.2.0



/usr/local/lib/python3.10/dist-packages/pygad/pygad.py:748: UserWarning: The percentage of genes to mutate (mutation\_percent\_genes=5) re  
 If you do not want to mutate any gene, please set mutation\_type=None.

warnings.warn(f"The percentage of genes to mutate (mutation\_percent\_genes={mutation\_percent\_genes}) resulted in selecting ({mutation\_n  
 [[ 3.71394401 -2.21020831]  
 [-2.22055136 0.09544376]  
 [-2.60275911 2.32906075]  
 [-3.04086165 -3.02825741]  
 [-0.97366463 0.32427915]  
 [ 1.4816084 -2.01837178]  
 [ 3.40714184 3.02658144]  
 [ 0.21316026 1.77623626]  
 [-3.33799884 0.53677301]  
 [ 1.80051236 2.45743232]  
 [ 2.43315704 -0.82473271]  
 [ 3.45540804 3.45552818]  
 [-3.13787462 2.29778534]  
 [ 0.78407542 3.57772666]  
 [-3.54406358 -0.49454028]  
 [ 3.56526368 -1.06836785]  
 [ 2.88112257 -0.88342154]  
 [-3.31966846 3.43702909]  
 [-0.98687277 0.06106612]  
 [ 0.47101149 1.17984113]  
 [-1.28655777 3.1331558 ]  
 [-3.40055857 -0.22059904]  
 [-3.28529429 0.81991136]  
 [ 2.44403717 -0.86480333]  
 [-2.67272961 0.28799183]  
 [ 3.04848473 -2.15198972]  
 [-1.87497271 0.52670116]  
 [ 1.93331413 -2.07929311]  
 [ 2.21065959 0.34362385]  
 [ 1.810041 3.22139238]  
 [ 3.83901762 1.92346421]  
 [-3.55993752 -0.09752996]  
 [-2.14155336 0.95928504]  
 [ 3.11554226 -0.94561066]  
 [-3.03620912 0.15852478]  
 [-3.24949699 0.95429288]  
 [ 0.00000000 0.00000000]