Lab 11: Evolutionary Computation (week 6,7)

1 – Objectives

In this lab you will learn to use the "Deap Library" to solve some problems using evolutionary algorithms.

2 – DEAP

DEAP, https://pypi.org/project/deap/, is a novel evolutionary computation framework for rapid prototyping and testing of ideas. It seeks to make algorithms explicit and data structures transparent. It works in perfect harmony with parallelization mechanisms:

You can find all information about this library at:

https://deap.readthedocs.io/en/master/.

3 – GA: One Max Example

We will start with the One Max example. Read the documentation available at:

https://deap.readthedocs.io/en/master/examples/ga_onemax.html

Run the example with the debugger and see what it does. See how the population, each chromosome, and each gene are created. See how the crossover and mutation functions are chosen. Then how the population evolves.

Now lets try to make some modifications to the program. Print the best individual in each generation. Compare it with the HallOfFame.

Make 10 runs of the program and compute how much time it took to perform each run.

Change the crossover function and mutation functions and see if there is any difference in the run time.

4 – PSO: Examples

Now let's move to PSO. Try the "basics" example available at:

https://deap.readthedocs.io/en/master/examples/pso_basic.html

Afterwards, see how PSO performs with the Moving Peaks Benchmark:

https://deap.readthedocs.io/en/master/examples/pso_multiswarm.html

5 – Function Optimization

Now try to solve the problem of finding the maximum value of $f1(X1,X2)$ using both GA and PSO

$$Z1 = \sqrt{X1^2 + X2^2}$$

$$Z2 = \sqrt{(X1-1)^2 + (X2+1)^2}$$

$$f1 = (\sin(4 * Z1)\ /Z1) + (\sin\ (2.5 * Z2)/Z2)$$

6 – <u>Optional</u> Report for Bonus points

By submitting a pdf containing the code you developed and the results you obtained in section 5, you will get a 1 val bonus on Project 3 (available soon). The optional report should be submitted together with Project 3.

Appendix: 3D plot with Matplotlib (also useful for Proj2)

In order to visualize how evolutionary algorithms work, it is useful to plot the results in 3D. The following example plots a 3D figure of a function. You can also use the function ax.scatter(x, y, f1, c='red', marker='x') to plot a point in the figure. You control the transparency of the figure with the alpha parameter in the plot_surface function.

```
def plotFigure():
    fig = plt.figure()
    ax = fig.gca(projection='3d')

    # Make data.
    X = np.arange(-5, 5, 0.05)
    Y = np.arange(-5, 5, 0.05)
    X, Y = np.meshgrid(X, Y)

    Z1=np.sqrt((X**2)+(Y**2))
    Z2=np.sqrt(((X-1)**2)+((Y+1)**2))

    f1=(np.sin(4*Z1)/Z1)+(np.sin(2.5*Z2)/Z2)
    f2=1-(np.sin(5*Z1)/Z1)
    xlen = len(X)
    ylen = len(Y)
    # Create an empty array of strings with the same shape as the meshgrid, and
    # populate it with two colors in a checkerboard pattern.
    colortuple = ('y', 'b')
    colors = np.empty(X.shape, dtype=str)
    for y in range(ylen):
        for x in range(xlen):
            colors[x, y] = colortuple[(x + y) % len(colortuple)]

    # Plot the surface with face colors taken from the array we made.
    surf = ax.plot_surface(X, Y, f1, facecolors=colors, alpha=0.75, linewidth=0)

    plt.show()
    return fig, ax
```