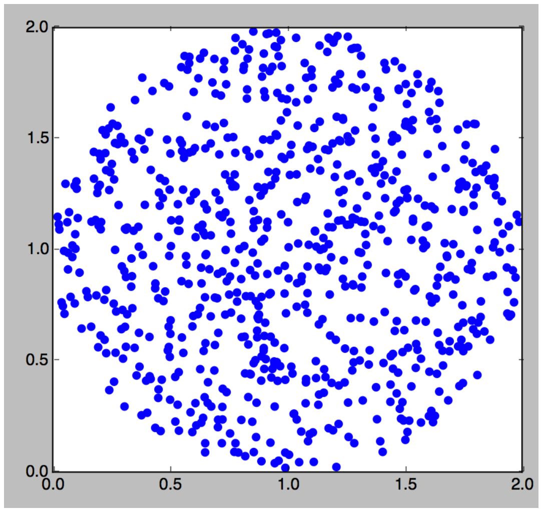
ECI 289I: Applied Evolutionary Computing
Assignment #5

Please include your code with your assignment. Feel free to use any programming language.

(1) Nondominated sorting

The file `circle-points.txt` contains several hundred "candidate solutions" to a 2-objective optimization problem, which are conveniently arranged in a circle of radius 1 centered at (1,1):



Write a function `nondom_sort(points)` which accepts a Numpy matrix of points—where each row is a solution and each column is an objective—and returns a matrix of only the non-dominated points. A few criteria:

- Your function should not modify the matrix `points` directly (no side effects!)
- Your function should work for an arbitrary number of objectives, not just two
- Assume all objectives are to be minimized

Test your function using the `circle-points.txt` data and include a scatter plot showing the non-dominated points.

(2) Fix my code!

Imagine a hypothetical scenario where your instructor wrote an incorrect `archive_sort()` function in the file `L9-multiobj-archive.py`. Recall that this function should accept the current archive and population matrices in the algorithm, and return an updated archive **only** containing the nondominated solutions. Please fix it so that all dominated solutions are removed. You can start writing the function from scratch; your code from Problem 1 may be useful here!

Once you have it working, try optimizing the example `mymulti1` function and make a scatter plot of the objective space and decision space after some number of evaluations. Most of what you need is already there—just remove the animation commands to make only a single plot.

(3) Application

Liao et al. (2007) describe a multi-objective optimization problem involving the crash safety of a car, beginning in Section 5 of the paper. The decision variables are the thicknesses of five reinforced members in the front of the car (each between 1mm and 3mm, see Figure 8), and the three objectives are: to minimize the mass of the car (Mass), to minimize a measure of damage due to deceleration (Figure 9), and to minimize the "toe board intrusion" (Figure 10).

The finite element model to calculate these objectives would be far too complicated to run for this assignment! But fortunately the authors have provided regression approximations of the objective functions that we can use here (Eqns. 14-16). This objective function is included in the file `eci289I-hw5-p3.py` so you don't have to type in all of the numbers.

Set up this multi-objective function and optimize it using a MOEA of your choice from the Platypus library. Report your experimental settings: the algorithm used, NFE, and parameter values.

Once you have your estimate of the Pareto front, create both a 3D scatter plot and a parallel axis plot. Compare (visually) to the Pareto set that the authors report in Table 1 of the paper. Can you approximately reproduce their results?

(4) Read a paper

Do some searching for a paper in your field of study that uses a MOEA. It should be an application-focused paper, not one that develops or proposes a new algorithm. It may take some time on Google Scholar or the like to find one that catches your attention! Especially because not all papers that use MOEAs will say so in the title.

Once you read the paper, provide a short summary and see if you can answer the following:

- Describe the optimization problem. How many decisions and objectives?
- What problem properties caused the authors to use an evolutionary algorithm?
- What is the experimental setup? How many function evaluations / random trials?
- Algorithm parameter values (if they are given)?
- How do they assess if convergence is reached?
- Is the resulting Pareto set used somehow for decision support? How is it plotted?
- Any other relevant details that you find interesting

You are welcome to provide a critique if you feel that any important aspects could be improved.