Homework 6

MSSC 6000 — Scientific Computing — Spring 2023

Due Friday, May 5, on D2L, by 11:59pm

1. Recall the job scheduling problem from a previous assignment: You have n jobs $J_1, \ldots, J_n$. Each job J_i has a duration t_i, a deadline d_i and a profit p_i, and your goal is to find the schedule that leads to the largest profit.

   In a previous assignment, you constructed a branch-and-bound solution to this problem. On this assignment, your goal is to find a dynamic programming solution. Figure out the recurrence, just like we did in the examples in class, and make sure that you fully explain how you derived it.

   Then, use your recurrence to find the optimal solution to the job list below by hand, on paper, showing all of your steps. You do not need to code anything for this assignment.

Job #	1	2	3	4	5
Duration	2	5	1	1	2
Deadline	5	8	2	3	2
Profit	3	2	1	3	5

2. In class, we discussed an optimization problem involving spring design (you can refer to the lecture notes and the jupyter notebook). The goal is to design a spring whose weight is minimal while still satisfying conditions that ensure it will work as needed. There are three parameters we can change: d, the diameter of the coil; L, the length of the spring; w, the thickness of the wire.

   The quantity we wish to minimize is $(2+L)dw^2$. Our parameters must satisfy the boundary conditions

$$0.05 \leqslant w \leqslant 2.0 \qquad 0.25 \leqslant d \leqslant 1.3 \qquad 2.0 \leqslant L \leqslant 15.0$$

   as well as the constraints

$$g_1(L,d,w) = 1 - \frac{d^3 L}{7178 w^4} \leqslant 0$$

$$g_2(L,d,w) = \frac{4d^2 - wd}{12566 dw^3 - w^4} + \frac{1}{5108 w^2} - 1 \leqslant 0$$

$$g_3(L,d,w) = 1 - \frac{140.45 w}{d^2 L} \leqslant 0$$

$$g_4(L,d,w) = \frac{w+d}{1.5} - 1 \leqslant 0$$

   that ensure the spring will work.

   In this exercise, you will write an implementation that uses Particle Swarm Optimization to solve this problem. You should submit:

   - A file called `PSO_spring.py` with all of your code. When I run your file, it should start executing your code. I would like your code to print the global best score and the corresponding quantities (w, d, L) each time the global best score improves. I will run your code for a minute or two, so make sure it can find reasonable solutions in that time.

- A document describing the design of your algorithm. For example, you should discuss how you are handling the boundary conditions and constraints to prevent particles from moving into illegal solutions, how you have determined the parameters α, β, γ, etc.

One additional note:

- You may wish to look up and use the package “numpy” for its arrays that will make vector arithmetic a little simpler, but you might also choose to just do it manually so you don’t have to learn about numpy. Your choice!