HW8

Wednesday, April 9, 2025 6:20 PM

8.1 Knapsack Problem: You are a supervisor overseeing 5 subject matter experts who have been hired by your company for one day of work. Collectively, they have 40 hours (5x8) available for work tomorrow. You are scheduling what projects they will work on tomorrow to maximize the value of the available hours. Currently, at your company there are 27 projects or tasks that need attention and that the export could work on. Of course, there isn't enough time to get them all done but you want to optimize their schedule and assignments as they are each receiving $1200 from the company for the one day of work. These projects have been evaluated with respect to how long they will take to complete (t_i) and their value v_i to the company. No task can be split up across two different days as the experts are flying home tomorrow evening, but two experts can work on the same project.

Which projects should you select from among the following (see table)?

Include your V matrix and S matrix with your solution in your write-up.

Discuss what you would do if you applied a greedy algorithm to this problem. Would you get the same result? Why or why not?

i	t_i (hours)	v_i ($)
1	7	16000
2	4	1000
3	2	7000
4	2	14000
5	3	6000
6	4	10000
7	7	12000
8	5	18000
9	4	5000
10	3	11000
11	2	6000
12	3	13000
13	2	18000
14	2	5000
15	2	17000
16	7	17000
17	3	7000
18	2	12000
19	1	14000
20	4	9000
21	4	9000
22	1	6000
23	4	3000
24	7	6000
25	6	12000
26	7	8000
27	2	19000

We select the following tasks based off of the tabulated dynamic programming optimization:

```
1  selected_items
✓ 0.0s
[27, 22, 19, 18, 16, 15, 14, 13, 12, 11, 10, 8, 6, 4, 3]
```

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14000	14000	14000	14000	14000	14000	14000	14000	14000
0	0	0	7000	14000	14000	14000	14000	14000	14000	14000	14000	14000	18000	18000	18000	18000	18000	18000	18000	18000	18000	20000	20000	20000	20000	20000	20000
0	0	0	7000	14000	14000	14000	14000	14000	14000	14000	14000	14000	18000	18000	18000	18000	18000	18000	32000	32000	32000	32000	32000	32000	32000	32000	33000
0	0	1000	7000	21000	21000	21000	21000	21000	21000	21000	21000	21000	32000	32000	35000	35000	35000	35000	35000	35000	35000	38000	38000	38000	38000	38000	39000
0	0	1000	7000	21000	21000	21000	21000	21000	21000	25000	25000	27000	32000	32000	35000	35000	35000	35000	49000	49000	49000	49000	49000	49000	49000	49000	51000
0	0	1000	8000	21000	21000	24000	24000	24000	24000	25000	27000	27000	39000	39000	49000	49000	49000	49000	49000	49000	49000	55000	55000	55000	55000	55000	57000
0	16000	16000	16000	21000	27000	27000	27000	32000	32000	32000	32000	34000	45000	45000	49000	49000	49000	49000	63000	63000	63000	63000	63000	63000	63000	63000	68000
0	16000	16000	16000	22000	27000	31000	31000	32000	32000	32000	32000	38000	45000	45000	56000	56000	56000	61000	63000	63000	63000	69000	69000	69000	69000	69000	74000
0	16000	16000	23000	30000	30000	31000	31000	39000	39000	39000	39000	40000	52000	52000	62000	62000	62000	62000	75000	75000	75000	75000	75000	75000	75000	75000	82000
0	16000	16000	23000	30000	30000	31000	31000	39000	39000	43000	43000	45000	56000	56000	62000	62000	62000	68000	76000	76000	76000	81000	81000	81000	81000	81000	88000
0	16000	17000	23000	37000	37000	37000	37000	42000	42000	43000	45000	45000	58000	58000	69000	69000	69000	74000	82000	82000	82000	82000	82000	82000	82000	82000	94000
0	16000	17000	23000	37000	37000	37000	37000	45000	45000	50000	50000	52000	63000	63000	73000	73000	73000	74000	88000	88000	88000	88000	88000	88000	88000	88000	100000
0	16000	17000	24000	37000	37000	40000	40000	49000	49000	50000	50000	56000	63000	63000	75000	75000	75000	81000	88000	88000	88000	94000	94000	94000	94000	94000	101000
0	16000	17000	24000	37000	43000	43000	43000	49000	49000	53000	56000	58000	70000	70000	80000	80000	80000	85000	95000	95000	95000	95000	95000	95000	95000	95000	107000
0	16000	17000	24000	38000	43000	47000	47000	49000	49000	56000	56000	63000	74000	74000	80000	80000	80000	87000	99000	99000	99000	101000	101000	101000	101000	101000	113000
0	16000	17000	24000	38000	43000	47000	47000	55000	55000	60000	60000	63000	76000	76000	87000	87000	87000	92000	101000	101000	101000	105000	105000	105000	105000	105000	114000
0	16000	17000	24000	38000	43000	47000	47000	55000	55000	60000	62000	69000	81000	81000	91000	91000	91000	92000	106000	106000	106000	107000	107000	107000	107000	107000	120000
0	16000	17000	24000	38000	44000	53000	53000	58000	58000	60000	66000	69000	81000	81000	93000	93000	93000	99000	106000	106000	106000	112000	112000	112000	112000	112000	124000
0	16000	17000	24000	38000	44000	53000	53000	61000	61000	66000	66000	73000	87000	87000	98000	98000	98000	103000	113000	113000	113000	113000	113000	113000	113000	113000	126000
0	16000	17000	24000	38000	44000	53000	53000	65000	65000	66000	66000	75000	87000	87000	98000	98000	98000	105000	117000	117000	117000	119000	119000	119000	119000	119000	131000
0	16000	17000	24000	38000	44000	53000	55000	65000	65000	69000	72000	79000	91000	92000	104000	104000	104000	110000	119000	119000	119000	123000	123000	123000	123000	123000	132000
0	16000	17000	24000	38000	44000	54000	59000	65000	65000	72000	72000	79000	93000	93000	104000	104000	105000	110000	124000	124000	124000	125000	125000	125000	125000	125000	138000
0	16000	17000	24000	38000	44000	54000	59000	71000	71000	76000	76000	79000	97000	97000	109000	109000	109000	116000	124000	124000	124000	130000	130000	130000	130000	130000	142000
0	16000	17000	24000	38000	44000	54000	59000	71000	71000	76000	78000	85000	97000	98000	110000	110000	111000	117000	130000	130000	130000	130000	130000	130000	130000	130000	144000
0	16000	17000	24000	38000	44000	54000	65000	71000	71000	76000	82000	85000	97000	102000	114000	114000	114000	121000	131000	131000	131000	136000	136000	136000	136000	136000	149000
0	16000	17000	24000	38000	44000	54000	65000	73000	73000	82000	82000	89000	103000	103000	115000	115000	116000	123000	135000	135000	135000	137000	137000	137000	137000	137000	149000
0	16000	17000	24000	38000	44000	54000	65000	77000	77000	82000	82000	91000	103000	103000	119000	119000	119000	126000	137000	137000	137000	141000	141000	141000	141000	141000	155000
0	16000	17000	24000	38000	44000	54000	65000	77000	77000	82000	88000	95000	107000	108000	120000	121000	121000	128000	140000	140000	140000	143000	143000	143000	143000	143000	156000
0	16000	17000	24000	38000	44000	54000	66000	77000	77000	84000	88000	95000	109000	109000	120000	121000	122000	131000	142000	142000	142000	146000	146000	146000	146000	146000	160000
0	16000	17000	24000	38000	44000	54000	66000	83000	83000	88000	88000	95000	113000	113000	125000	126000	126000	133000	145000	145000	145000	148000	148000	148000	148000	148000	162000
0	16000	17000	24000	38000	44000	54000	66000	83000	83000	88000	90000	101000	113000	114000	126000	127000	128000	134000	147000	147000	147000	151000	151000	151000	151000	151000	165000
0	16000	17000	24000	38000	44000	54000	66000	83000	83000	88000	94000	101000	113000	118000	130000	131000	131000	138000	148000	149000	149000	153000	153000	153000	153000	153000	167000
0	16000	17000	24000	38000	44000	54000	66000	83000	83000	94000	94000	101000	119000	119000	131000	132000	133000	140000	152000	152000	152000	155000	155000	155000	155000	155000	170000
0	16000	17000	24000	38000	44000	54000	66000	84000	88000	94000	94000	103000	119000	119000	135000	136000	136000	143000	154000	154000	154000	158000	158000	158000	158000	158000	172000
0	16000	17000	24000	38000	44000	54000	66000	84000	88000	94000	100000	107000	119000	124000	136000	137000	138000	145000	157000	157000	157000	160000	160000	160000	160000	160000	174000
0	16000	17000	24000	38000	44000	54000	66000	84000	88000	94000	100000	107000	121000	124000	136000	137000	139000	148000	159000	159000	159000	163000	163000	163000	163000	163000	177000
0	16000	17000	24000	38000	44000	54000	66000	84000	88000	99000	100000	107000	125000	125000	141000	142000	143000	150000	162000	162000	162000	165000	165000	165000	165000	165000	179000
0	16000	17000	24000	38000	44000	54000	66000	84000	89000	99000	100000	113000	125000	126000	141000	143000	144000	151000	164000	164000	164000	168000	168000	168000	168000	168000	182000
0	16000	17000	24000	38000	44000	54000	66000	84000	89000	99000	105000	113000	125000	130000	142000	147000	147000	155000	165000	166000	166000	170000	170000	170000	170000	170000	184000
0	16000	17000	24000	38000	44000	54000	66000	84000	89000	99000	105000	113000	131000	131000	143000	148000	149000	156000	169000	169000	169000	172000	172000	172000	172000	172000	187000

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	0	1	1	1	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	1	1	0	1	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1
0	1	0	0	1	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	0	1	1	0	1	0	1	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1
0	1	0	1	1	0	1	0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	0	0	0	1	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	1
0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	1
0	1	1	1	1	0	1	0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	0	0	1	0	1	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	1
0	1	1	1	1	1	1	0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	0	1	0	1	1	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	0	1	0	1	1	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	1
0	1	1	1	1	1	1	0	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1	0	1	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1	0	1	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	0	0	0	0	1

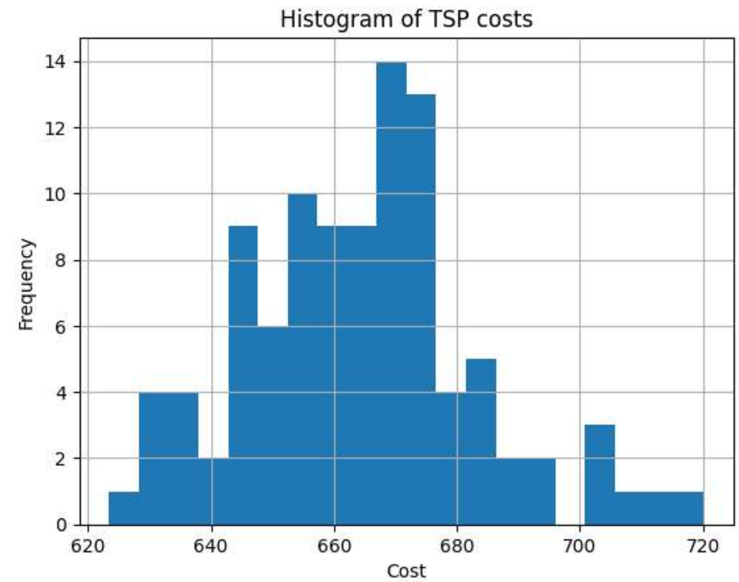
This algorithm performs better than and gets a different value than greedy (greedy got V = 170000), as greedy essentially sorts by value and takes the highest value tasks until there are no available hours. In reality some tasks are worth more per hour than others. And different combinations of tasks can yield higher overall value than even the value/hour would.

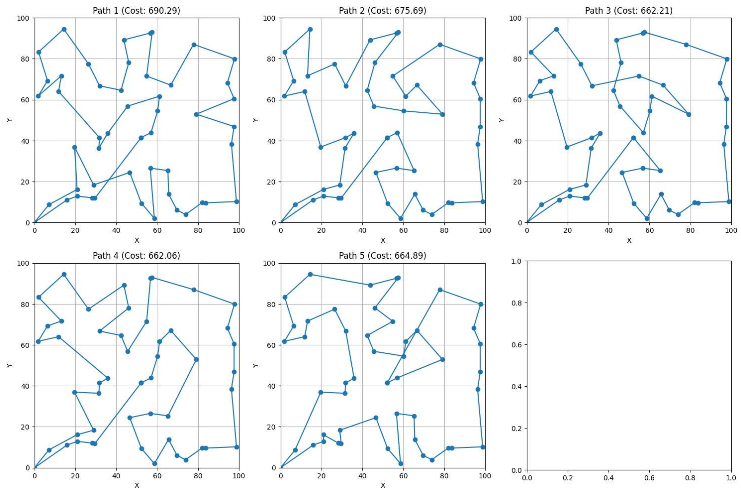
8.2 Simulated Annealing: Develop a simulated annealing program that optimizes the same Traveling Salesman problem from Homework #7 (i.e. visiting 49 points in an area 100 x 100 units square with the additional starting point (0,0)).

Once your program is working you should repeat the optimization 100 times and plot the histogram of the minimum path lengths achieved. Note: for each of these 100 repetitions, you need to have the same 50 point locations! How similar are the solutions? Plot 5 of the 100 paths you solved for.

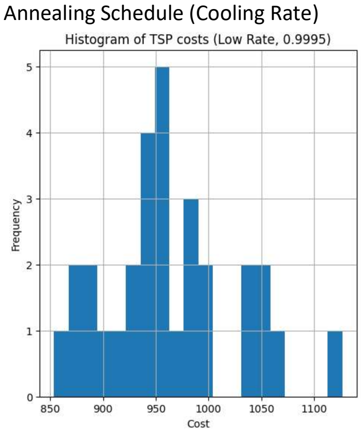
Next, explore 1) the impact of the annealing schedule, 2) the impact of the maximum number of iterations, and 3) the influence of the starting point. For each of those three aspects, you should run a comparison between two or more different strategies (i.e. fast annealing vs slow annealing). Since this optimization technique is considered stochastic, you should repeat each strategy multiple times (30 times or more is a good starting point) to obtain the mean behavior. Have a comparison plot for each of the three aspects.

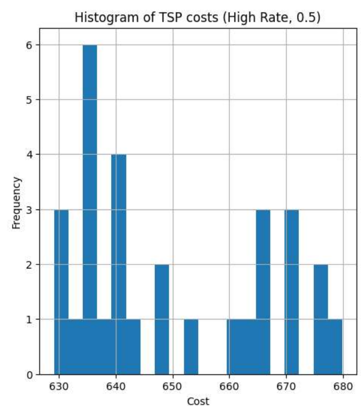
Lastly, discuss your definition of how you found a "neighboring design" at each iteration and how your overall process could be improved in 200 to 400 words.



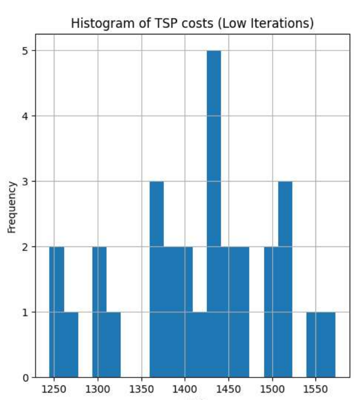


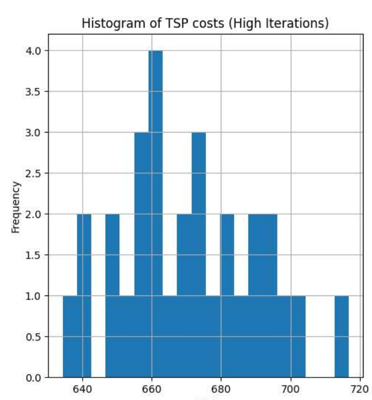
Annealing Schedule (Cooling Rate)



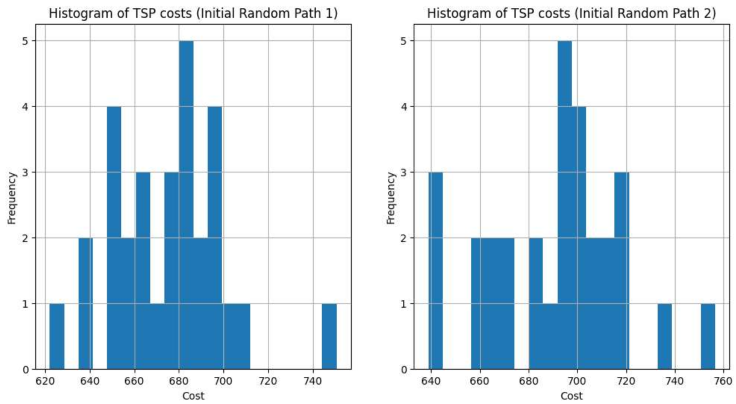


You can see that for the low cooling rate the final costs were all higher than the high cooling rate, and the results kind of look like a bell curve distribution. For the high cooling rate, it converged to better values, but it seems a little more random, it is possible that an even higher cooling rate would have made it converge to a worse solution quicker.





For iterations it can be seen that low iterations yield much worse solutions that are also more spread out compared to higher iterations. More iterations definitely help to hone in on a better solution.



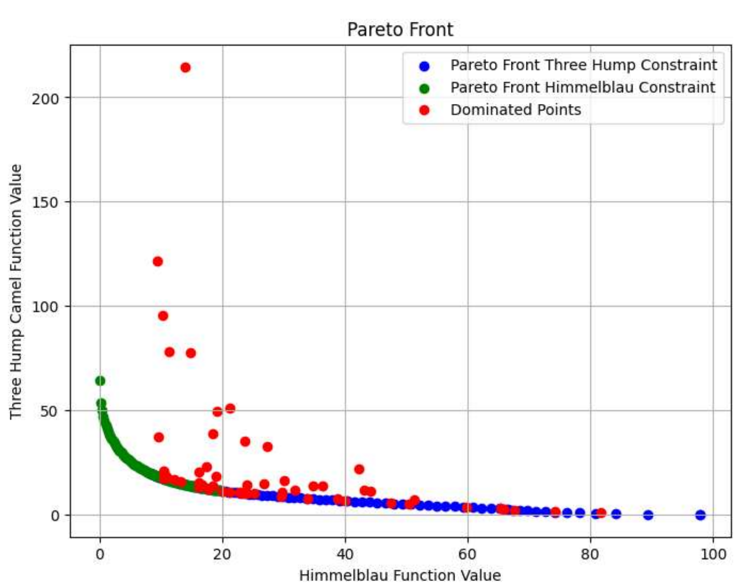
These are the results from starting with two different initial random paths. I was surprised that it did make a difference, while the values overlap etc the mean of the first one was 674, and the second was 690, so clearly your initial random starting path for the optimization does make a difference. It would be important to random start from a lot of different spots.

Neighboring Design: To find the neighboring design I followed the method in the book- I randomly choose between one of two options, the first is to reverse the order of a segment, so a random start and end point are chosen, then the order of the points in between are reversed. The second option was to randomly select a segment by the start and end like above, then move that segment behind another randomly chosen index/point. This helps beyond reversing order by allowing the connections themselves to also change randomly. To improve the overall process, it would be nice to take away some of the randomness, maybe I would implement a greedy algorithm randomly as one of the two options or add it as a third option, and then have that be kept and changed. Maybe making it closer to GA style could help too, like taking two of the best options and combining them somehow.

8.3 Multi-Objective Optimization: Select **two** functions found here: https://en.wikipedia.org/wiki/Test_functions_for_optimization
Make adjustments if necessary so the optimal point of the first function is not the same as the second function. (i.e. the optimal point should not be, say, (0,0) for both functions). If you want, you can define your own function for this problem or use two objective functions from your project.

Using a method of your choice define the Pareto front of the multi-objective function $f = [f_1 \;\; f_2]^T$. Plot at least 10 points on the Pareto front and then keep or add in 50 or more points that are dominated (i.e. not on the Pareto front).

Discuss the method you selected and its pros and cons in defining the Pareto-front in 200 to 400 words.



I used the Epsilon Constraint Method to find the Pareto Front. This basically chooses a function as the objective, and makes the other one a constraint. Epsilon is the constraint value, ie saying that this function needs to be <= some epsilon value. I had a lot of different epsilon values and looped through using scipy.minimize for each function constraint set. I also changed which function was the constraining one, but that wouldn't have made a difference if I used a larger epsilon range for just one of them. A good perk of this is that it finds the pareto front very easily and is simple to code. However, it would not scale well as it requires a minimization for each point found on the pareto front. My dominated points were found by adding noise to the pareto (x,y) points and evaluating those, They visually appear all dominated, and were checked using AI generated code that compares the values of the dominated points to those of the pareto points, and if one fails, it was removed.