Activity 1 – Local Search (15%)
Activity is due on Friday, March 11, 2022, 11:59 pm.

Consider the following problem: Given a set of n packages with profit p_j and weight w_j , and a set of m containers with weight capacity c_i, select m disjoint subsets of packages so that the total profit of the selected packages is maximum, while ensuring the containers' capacity is never exceeded.

Consider the following two instances to be generated in python (other software can be used to code the algorithms, but the instances should be generated in python using random.seed(1)):

Instance 1	**Instance 2**

random.seed(1)
n = 100 #number of packages
m= 5 #number of containers
c=50
 	
#Generate random locations
pj = random.choices(range(10, 100), k=n)
wj = random.choices(range(5, 20), k=n)
	random.seed(1)
	n = 10000 #number of packages
	m= 200 #number of containers
	c=50
	
	#Generate random locations
	pj = random.choices(range(10, 100), k=n)
	wj = random.choices(range(5, 20), k=n)

Exercise 1: Formulate the problem mathematically

Exercise 2: Solve the problem using pyomo

Exercise 3: Propose and apply a random sampling algorithm for the problem

Exercise 4: Propose and apply a local search algorithm for the problem (consider two different strategies for constraint handling; the initial solution must be randomly generated)

Exercise 5: Propose and apply a simulated annealing algorithm for the problem (consider two different strategies for constraint handling; the initial solution must be randomly generated)

Exercise 6: Propose and apply a greedy algorithm for the problem

Exercise 7: Solve exercise 5 again, this time using the greedy solution as initial solution. Try to achieve the best solution possible within 1 minute (for instance 1) and 15 minutes (for instance 2).

Deliverables: (1) textfile (or screenshot) of the code used to run each exercise**;** (2) pdf report including: (2.1) mathematical formulation of the problem; (2.2.) solution encoding used ; (2.3) explanation of the search operator and constraint handling techniques used ; (2.4) explanation of any other heuristic rules used ; (2.5.) explanation of the greedy heuristic applied in exercise 6; (2.6.) results obtained for each exercise. The results should include: (2.6.1) a table with information of the objects included in each container, the total weight in each container, the profit in each container; (2.6.2) a plot with the cpu performance. See example below:

Table 1 - Table of results

Container	**Packages**	**Profit**	**Weight**
1	3,5,7	54	50
2	10,24,1	64	49
3	89,21	48	49
4	55,63,29	61	50
5	8,14,13,47	75	48
Total		**302**	

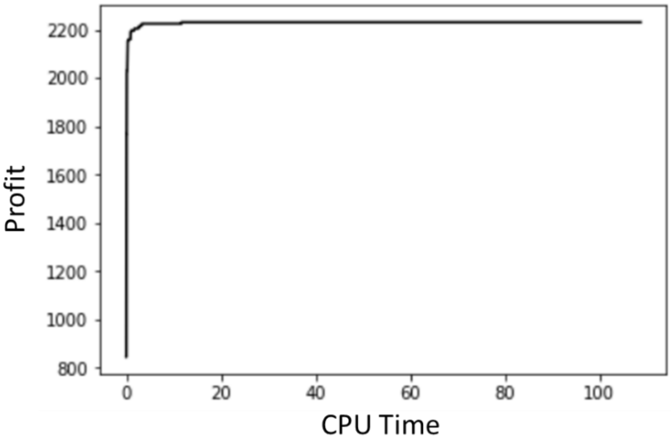

Fig 1 – CPU performance