Knapsack Problem

Write a genetic algorithm to solve the knapsack problem. Given size of a knapsack, a number of items, weight & benefit of each item. You should select the maximum number of items that could fit in the knapsack and at the same time provide/ensure the maximum benefit.

For more info about the problem please refer to this link: https://en.wikipedia.org/wiki/Knapsack_problem.

Input will be in following format:

C Number of test cases (Maximum 20 test).

N Number of items (Maximum 50 item).

S Size of knapsack.

N pair values representing the weight and benefit of each knapsack (weight, benefit).

Input (Example):

1

3

10

4 4

7 6

5 3

(e.g. we have 1 test case, 3 items, size of the knapsack is 10, and 3 pair values. in the second pair value, we have an item with weight 7 and benefit 6)

Output:

Case ID: total benefit.

M: number of items that could be placed in the knapsack.

M pair values (weight, benefit) of selected items.

Output (Example):

Case 1: 7

2

4 4

5 3

Algorithm Steps:

1. First thing to think about is how you will encode the solution in your chromosome. After figuring that out you will be able to randomly generate your initial population.
2. We know that our chromosomes will consist of 0's and 1's. We can say that each chromosome's length will equal to the given number of items, where each element/gene in the chromosome will correspond to an item. That item will be represented by 1 if it will be included in the knapsack.
3. Now you can generate your initial population. Generation mechanism is up to you. For example your can generate a random number, and if that random number is < 0.5 then the item will be included in the knapsack, hence you will add 1, otherwise add 0. Population size is determined by you. And you can actually try different population sizes to see how it will affect your results.
4. Evaluate fitness function which will be the sum of the included items' weights. Example: suppose we have 3 items (4, 4), (7, 6), and (5, 3), and our chromosome is [1, 0, 1]. Then our fitness function will equal = 4 * 1 + 6 * 0 + 3 * 1= 7. **NOTE:** you need to add a condition to check if the chromosome is a feasible solution or not, if not you need to handle this case. How you will handle it, is up to you. Surprise us!
5. Perform Selection.
6. Perform crossover.
7. Perform mutation.
8. Replace old generation by new one.
9. Repeat steps 4 through 8, till you reach your termination condition.
10. Don't forget to print the final output. When printing the output, you need to decode the 1's back to the items' pair values.

Notes:

- You will be graded based on your output (the higher the values, the higher the grade).
- Number of team members: 3-4.
- Assignment will be delivered and discussed in next week's labs.