Evolutionary Computing - 2020
Assignment #2: Genetic Algorithm

This home work is an abstract war game that generalizes to many situations involving competing strategies of resource distribution. In the military context, it goes as follows:

Each colonel has a limited number of soldiers, S, and a fixed number of battles, B, to be fought during a war (against one opposing colonel). In each battle, the commander who shows up with the most troops wins, and the commander who wins the majority of the B battles wins the war. The colonel's job is to decided, prior to the first battle, the distribution of troops for all the battles.

Your task here is to implement the genetic algorithm (GA) to evolve the colonels' strategies.

A representation for each strategy is simply a list of B non-negative integers (that sum to S). Each strategy will compete (i.e., have a war) with every other strategy in the population. Each war win is worth 2 points, and each tie is worth 1; losses are worth 0. Fitness should be directly based on the total number of war points. The degree of battle victory does not count, only the binary act of winning or losing.

Use the following parameters in the default implementation:

- Random initialization
- Population size: 50
- Tournament selection
- 10 % elitism.

Colonel	Battle 1	Battle 2	Battle 3	Battle 4
A	8	1	6	5
B	9	1	0	10
C	6	6	4	4

Then make an experiment with different parameter setups. Here are some parameters that you should experiment with:

1. Implement different mutation operator
2. Implement different selection mechanism
3. Incorporate the following factor into your code for simulating a war

For your third experiment you should consider redistributing the soldiers to have higher winning chance.

Troop redistribution - If Colonel A wins a battle with X troops versus Colonel B's Y troops, then the $(X - Y)$ extra troops can be re-deployed in all succeeding battles (in Colonel A's strategy), by evenly distributing $R_f(X - Y)$ resources among the remaining battles, where R_f is in range of [0,1] .

The visualization of each GA run must involve at least these 3 things:
1. A fitness plot, showing:
   The fitness of the best individual in the current generation
   The average fitness across the whole population
2. A listing of the winning strategy for each generation.
3. Take 3 of the runs that are significantly different from one another and describe them in detail. In your report, include fitness along with your descriptions, for each of these 3 signature runs.

Notes:
- Your implementation should be functional.
- You should write a complete report for the results you get.
- Allowed programming languages: MATLAB, python, Java, …
- Any sign of cheating would be result in the zero grade for the assignment.
- Your codes should be self-commented.
- Send you codes in a ZIP file named "LASTNAME - FIRSTNAME.zip"