ECI 289I: Applied Evolutionary Computing
Assignment #4: Boolean Satisfiability

Please include your code with your assignment. Feel free to use any programming language.

(a) In the SAT problem, we are aiming to satisfy a compound Boolean statement, for example:

$$E = (A \vee B) \wedge (\neg A \vee C) \wedge (C \vee \neg D)$$

The symbols $\vee$, $\wedge$, $\neg$ stand for "or, and, not" respectively. In this example, E will evaluate to "True" (i.e. the statement will be satisfied) for a few different combinations of A, B, C, D.

This statement is an instance of 2-SAT written in conjunctive normal form (CNF), which means that it contains "or" clauses with 2 terms each, separated by "and" symbols indicating that all clauses must be satisfied. (This is the opposite of disjunctive normal form, where the "ands" and "ors" are switched around, and therefore only one clause must be satisfied, which is a much easier problem!)

In this problem we'll solve an instance of 3-SAT from the SATLIB library of benchmark problems. The data for this problem is given in the file `uf20-01.txt`, and you can use `np.loadtxt('filename')` to read it into a matrix. This problem contains 20 binary decision variables (x_1 through x_{20}) and 91 clauses. Each row in the matrix is a clause. For example, the first row [4, -18, 19] represents the clause $(x_4 \vee \neg x_{18} \vee x_{19})$.

Use a binary GA to solve this problem (see code from Lecture 7). Your objective function should be the number of clauses satisfied (maximize), where the optimal value is 91. You will likely need to experiment with the population size, crossover/mutation probabilities, tournament size, etc. to improve the search. Report results for a few random seeds—are you able to solve it? If so, how reliably can you reach the optimal result?

(b) Now that you have the objective function written, try optimizing the much larger problem described by uf250-01.txt. This problem is still an instance of 3-SAT, but it has 250 binary decision variables and 1065 clauses. How close can you get?

(c) You may have noticed that the initial random solutions perform well. Gent and Walsh (1993), which is posted on Canvas, offer a reason why this is. Based on their reasoning, how many clauses would you expect to be satisfied by the initial random solutions in (a) and (b), respectively? Does this approximately match your findings?