Habib University
CS 451 - Computational Intelligence
Spring' 2023
Assignment 1 – Evolutionary Algorithm

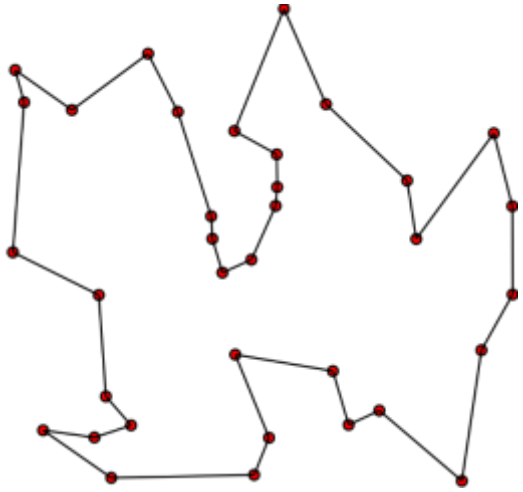
Objective:

The purpose of this assignment is to provide students an insight of stochastic optimization using Evolutionary Algorithms (EA). This exercise will enable them to address some popular computing problems that map to several real-world problems and are known to be computationally hard.

Q 1 – Evolutionary Algorithms

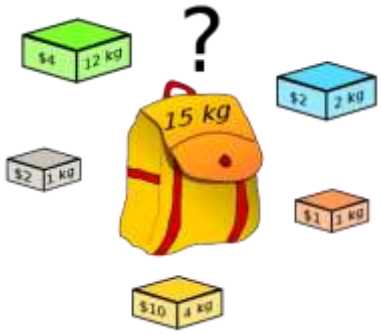
You have learned Evolutionary Algorithms (EA) and have seen their applications in optimization. In this question, you will implement EA and will use it to address the following benchmark problems.

- ***Travelling Salesman Problem (TSP):*** TSP asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?"



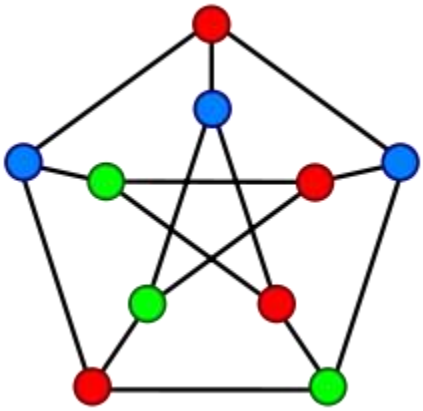
  You will use the dataset of Qatar that contains 194 cities. The optimal tour reported so far for this dataset is of length 9352 (and the best we have achieved by any CI student is 9939). The dataset and its related details can be found at: http://www.math.uwaterloo.ca/tsp/world/countries.html (look for Qatar dataset on this page).
- ***Knapsack Problem:*** The knapsack problem is a problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to

include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.



You will use knapsack lower dimensional data instances available at http://artemisa.unicauca.edu.co/~johnyortega/instances_01_KP/. You can try with different data instances but will report your results on *'f8_l-d_kp_23_10000'* data instance which contains 23 items and a knapsack of capacity 10000.

- ***Graph coloring:*** Graph Coloring is a classic problem in Computer Science in which you are required to color the vertices of a graph (vertex coloring) with minimum colors such that no two adjacent vertices are of same color (as shown in the image below).



Several problem instances of Graph Coloring are available here from where you can download the data file (gcol1.txt) for your testing.

You will ensure that your implementation is modular, generic and is structured in an objected oriented model with common classes being reused for above three problems.

EA Implementation:

Your EA will perform the following cycle:

- Step 0: Analyse the problem that you are addressing and decide your chromosome representation and fitness function.
- Step 1: Initialize the population randomly or with potentially good ***solutions***.

- Step 2: Compute the ***fitness*** of each individual in the population.
- Step 3: Select parents using a ***selection procedure***.
- Step 4: Create offspring by ***crossover*** and ***mutation*** operators.
- Step 5: Compute the ***fitness*** of the new offspring.
- Step 6: Select members of population to die using a ***selection procedure***.
- Step 7: Go to Step 2 until the termination criteria are met.

The following selection schemes will be implemented:

- Fitness Proportional Selection
- Rank based Selection
- Binary Tournament
- Truncation
- Random

Following parameters will be configurable in your program with some default values given below that you are free to change and observe the behavior of your algorithm:

- Population size: 30
- Number of offspring to be produced in each generation :10
- No. of generations: 100
- Mutation rate: 0.5
- No of Iterations: 10

EA Evaluation:

In each generation, you will note the average fitness (avg) of the generation and the best fitness so far (BSF). Suppose you run the process for 40 generations. You will have following observations:

Generation #	Best Fitness so far (BSF)	Average Fitness so far (ASF)
1		
2		
..		
..		
..		
40		

You need to repeat this exercise (for a single combination of parent and survival selection schemes) K times where K is your number of iterations. Each run will start with a new randomly initialized population.

You will have the following table after K iterations:

Generation #	Run # 1 BSF	Run # 2 BSF	..	..	Run # 10 BSF	Average BSF
1						
2						
..						
..						
..						
40						

A similar table will be obtained for average 'average fitness'.

- Once you have calculated (a) average best-so-far values and b) average average-fitness, you need to plot these values against generation # and analyze their pattern.
- You will repeat the process for different combinations of parent and survival selections such as (not limited to):
  - FPS and Random
  - Binary Tournament and Truncation
  - Truncation and Truncation
  - Random and Random
  - FPS and Truncation
  - RBS and Binary Tournament
  - Random and Truncation

Finally, you will provide your analysis on the behavior of EA under different parameters and selection schemes. Which scheme did work best in your case and which one performed worst?

Grading:

Problem Formulation (for three problems)	15%
EA Implementation (EA Process, Selectin Schemes) (Correctness of code, Proper Structuring of code, Generic Implementation)	40%
Fine-tuning of parameters and final solution	10 %
Gathering statistics and plotting graph	15 %
Analysis and Findings	20 %

Submission:

The assignment will be done in pairs. Please sign up for your Assignment 1 team on Canvas. You will submit your code and a pdf file describing:

- Chromosome Representation and Fitness Functions
- Plots of Avg. BSF and Avg. ASF for various combinations of schemes
- Your analysis for various schemes and the scheme that worked best in your case.