

Habib University
CS 451 - Computational Intelligence
Spring' 2021

Assignment 1 – Evolutionary Algorithm

Due on: Feb 14, 2021

Objective:

The purpose of this assignment is to provide students an insight of stochastic optimization using Evolutionary Algorithms (EA). This exercise will enable students to address some complex real-world problems that cannot be solved by greedy methods or analytical approaches.

Q 1 – Evolutionary Algorithms [40 Points]

You have learned Evolutionary Algorithms and have seen their applications in optimization. In this question, you will implement EA and will use it to address two different benchmark problems i.e. Traveling Salesman Problem (TSP), and Knapsack Problem.

- **Travelling Salesman Problem (TSP):** You will use the dataset of Qatar that contains 194 cities. The optimal tour reported so far for this dataset is of length 9352. 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:** 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 'f2_l-d_kp_20_878' data instance which contains 20 items and a knapsack of capacity 878.

You will ensure that your implementation is modular and generic to be used for multiple problems with minimal changes.

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 the 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 FPS
 - 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?

Grading:

| | |
|--|------|
| EA Implementation (Correctness of code, Proper Structuring of code, Generic Implementation) | 55 % |
| Fine-tuning of parameters and final solution | 20 % |
| Gathering statistics and plotting graph | 15 % |
| Analysis and Findings | 10 % |

Submission:

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.

Q-2 Evolutionary Art [15 Points]

Modify the implementation of EA in Question 1 to evolve a human image using polygons. You can look at the example of [evolving Mosa Lisa using polygons](#) and its [preliminary implementation](#). This implementation does not apply the traditional cycle of evolution and still takes quite long to evolve. Can you come up with an efficient yet effective implementation using EA? Try with pictures other than Mona Lisa to see if your implementation is generic enough to work with any human image. You are welcome to incorporate ideas that result in better/quick evolution.

Grading:

| | |
|--|-----|
| Problem Representation | 25% |
| Fitness Computation | 25% |
| EA Process - Modification/Optimization | 25% |
| Image Rendering/Results | 25% |

Submission:

You will submit your code and final graphs/images generated as a pdf file.