**SE 3368 – SOFT COMPUTING**

**Spring, 2023**

**ASSIGNMENT 1**

**Given:** March 17, 2023 **Due:** March 31, 2023

Apply a genetic algorithm (GA) on the “Consecutive Ones Problem (COP)” according to the given instructions below.

**Definition of the “COP”:** It aims to find the n-length binary strings that contain the longest consecutive ones. The fitness function for this problem is: the length of the substring that contains the longest consecutive ones in the binary string. The objective function is to maximize this fitness value. For example, let n = 10 and the number of solutions be as follows,

S1 = 000**111**0101, fitness = 3

S2 = 110**11111**01, fitness = 5

S3 = **11**00100**11**0, fitness = 2

The best solution is S2.

**Task:** You will apply a GA as shown in class. Code your solution to the problem in JAVA and fill the report below. Check the explanations and pseudocodes in the slides. Here are some steps you need to take:

Parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Problem 1** | **Problem 2** | **Problem 3** | **Problem 4** |
| **n** | 10 | 100 | 100 | 100 |
| **initial population size (m)** | 10 | 100 | 100 | 1000 |
| **# of iterations** | 10 | 100 | 1000 | 100 |
| **crossover type** | One Point Crossover | | | |
| **crossover ratio** | 70% | | | |
| **mutation type** | bitflip (flipping **n / 3** bits at random) | | | |
| **mutation ratio** | 30% | | | |
| **mating selection** | roulette wheel or tournament | | | |
| **survival selection** | elitism (transferring **m /10** individuals)  +  roulette wheel or tournament (for transferring **9\*m / 10** individuals) | | | |

Instructions:

* Use the parameters above in your GA.
* Generate the initial population randomly.
* Do not use any local search.
* At each iteration, the exactly number of offspring should be same as the size of the initial population (that is *n*).
* Run each Problem 1 through 4 for **10 times** and fill in the “Table” in the report below.
* In the “Concluding Remarks” section, write down your observation about your algorithm’s success by comparing the results of Problems 1 through 4.
  + How did the change in the values of “*n”*, “population size”, and “# of iterations” affect the success of your GA? Comment with a few brief sentences.
* After finishing your GA coding, running it, and filling your report, put your **JAVA** **project (with all your codes)** and the **report file** (convert to **PDF file**) in a folder. Name this folder with this format: “A1\_Firstname\_Lastname” (e.g. A1\_Kazim\_Erdogdu). Zip your folder and upload it to “Assignment1” in Moodle.

***NOTE:*** *Late submissions will get a* ***deduction*** *of* ***25 points per day****.*

**SE 3368 – SOFT COMPUTING**

**ASSIGNMENT 1 – REPORT**

**Student ID:**

**Student Name:**

**Department:**

| **Best fitness for** | **Problem 1** | **Problem 2** | **Problem 3** | **Problem 4** |
| --- | --- | --- | --- | --- |
| **Run 1** |  |  |  |  |
| **Run 2** |  |  |  |  |
| **Run 3** |  |  |  |  |
| **Run 4** |  |  |  |  |
| **Run 5** |  |  |  |  |
| **Run 6** |  |  |  |  |
| **Run 7** |  |  |  |  |
| **Run 8** |  |  |  |  |
| **Run 9** |  |  |  |  |
| **Run 10** |  |  |  |  |
| **Average of all runs** |  |  |  |  |
| **Best fitness of 10 runs** |  |  |  |  |
| **The genotype of the best fitness** |  |  |  |  |

**Concluding Remarks:**