# Evolutionary Computing - Fall 2019

## Assignment #2: GA

Shiraz University

Due Date: 12/Nov

Assume a cell phone company needs to setup antennas to cover a rectangular $m \times n$ area. Company has $k$ different type of antenna. $c_i$ is the cost of a $i$ typed antenna. $r_i$ is the coverage radius of a $i$ typed antenna. Antennas are installed at grid positions. When the Euclidian distance between the antenna and the point is less than or equal to $r_i$ , that point is covered by that antenna. A point is considered covered when at least one antenna covers it.

Company can install up to $s$ antennas. However, they want to cover as much of the $m \times n$ grid as possible with minimum cost.

You are asked to design and implement a genetic algorithm for the problem. $m$, $n$, $s$, $k$, $c_i$ and $r_i$ $i$ = 1, …, $k$ are given as input. The fitness value $f$ to maximize and summary of parameters are given as:

$m \times n$ = grid dimensions $k$ = number of different types of antenna

$c_i$ = cost of type $i$, $i = 1,2, \dots, k$

$r_i$ = radius of type $i$, $i = 1,2,\dots,k$

$u(x)$ = number of covered points in grid for individual $x$

$p(x)$ = set of installed antennas for individual $x$

$s$ = maximum number of antennas

$t(j)$ = type of antenna $j$

$$f(x) = u(x) - \sum_{j}^{p(x)} c_t(j)$$

As default encoding, use an array of size $s$ as your genotype. Each allele is a triple $(i, x, y)$. $i$ is a number in $[0, k]$. 0 means antenna is not installed, otherwise $i$ is the type of antenna. $x$ is an integer in $[0, m]$ and $y$ is an integer in $[0, n]$ denoting the position of the antenna. In this way, you expect to find the positions and types of antennas to install in the grid. Also use the following parameters in the default implementation:

- Random initialization of each integer value. Only valid triples.
- Population size 100
- Tournament selection
- 5% mutation probability. Randomly set one of the elements of the triple.
- One-point crossover
- 1% elitism. Replace worst offspring with best parent in the next generation.

Then make an experiment with different parameter setups. Here are some parameters for experiment:

- Different crossover operators (arithmetic crossovers etc.)
- Different selection mechanisms/pressures
- Another possible encoding or another idea you have (just ask me before)

Choose at least 3 parameter setups.

Repeat experiment for each setup for at least 10 times and collect results for best and average fitnesses for each generation and number of moves required to guess the number. Plot average charts for:

- Generation vs best fitness for 3 setups
- Generation vs average fitness for 3 setups

Notes:

- Your implementation should be functional.
- Allowed programming languages: MATLAB, PYTHON, JAVA.
- Feel free to change the model parameters.
- Any sign of cheating would be result in the zero grade for the assignment.
- Your codes should be self-commented.