CI - Laboratory 3: Evolutionary Algorithms

Jordi Armengol Estapé - Joan Llop Palao

1 Introduction and methodology

In this exercise, we have to minimize the Rosenbrock function by comparing different values of some operators. We will study their effect and find the minimum. For each configuration, we run the same algorithm 3 times and took the average fitness function. We set a random seed for reproducibility purposes. Specifically, we will fix some parameters to the default ones and we will perform a grid search over a set of different values for the required variables: 1. Population size: 10, 20, 50, 100. 2. Number of generations: 15, 50, 150, 30. 3. Range of the initial populations: [[-5, -5], [5, 5]], [[-2, -2], [2, 2]], [[-1.5, -1.5], [1.5, 1.5]]. 4. Selection functions: `selectionstochunif`, `selectionremainder`, `selectionuniform`, `selectionroulette`, `selectiontournament`. 5. Reproduction (crossover vs mutation ratio): 0, 0.2, 0.4, 0.6, 0.8, 1.

2 Results and discussion

We tried as many as **1440** combinations. In the table, we provide the correlations between the different numerical variables (all of them but the selection function) and the fitness score. Recall that we are minimizing the function. Thus, negative correlations imply that the corresponding variables are related to improving the fitness when increased, and the reverse holds for positive ones.

Variable	Correlation with FVal
Population	-0.200
Number of generations	-0.003
Initial range	0.243
Reproduction	0.15

In figure 1, we show how the fitness with respect of one of the variables (for brevity, we only show one of them). Apparently, the bigger the crossover vs. mutation ratio (recall that the bigger this ratio, the more crossover), the greater the fitness (which is bad since we are minimizing).

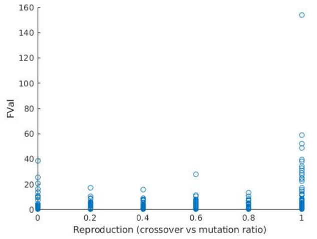

Figure 1: Scatter plot: Fitness with respect to the reproduction ratio.

The optimal parameter, which corresponds to a fitness score of almost 0 (specifically, `fval` = 0.0003479), is found with the following parameters: Population = 100, NGenerations = 50, InitialRange = 1.5, Selection = `selectiontournament`, Reproduction = 0. Analytically, the minimum of the function is found at x = 1, y = 1, almost what we obtained (x = 0.9910, y = 0.9818).

The population size seems to be important. The bigger the population, the less (better) fitness, because we have more individuals to experiment with. The number of generations does not seem to be very important, perhaps because with the minimum value we tested, the evolution has already enough iterations to converge. The initial range is positively correlated with the fitness, which means that with smaller range we obtain better fitness, because we are searching in a more reduced space near to the real optimum value. Finally, mutation seems to work better for this problem than crossover (so perhaps this operation is breaking good solutions), and the best selection function appears to be `selectiontournament`.