Evolutionary Computation

Notes for the PL 4

Ernesto Costa
João Macedo

Class PL 4

Design of Experiments: first steps

4.1 General Goal

We know that defining a solution for a problem based on an evolutionary algorithm is just the first step. Once we have one or more solution(s) we have to provide evidence that it is one of good quality, both in terms of **efficacy** (it solves the problem) and in terms of **efficiency** (it solve it fast). Unfortunately, we all know that the **No Free Lunch Theorem** states that there is not a general algorithm that is better than all the others for all class of problems. So, what we in fact seek is a good evolutionary algorithm for a particular class of problems. Today, we will analyze different solutions for an old friend, **João Brandão's Numbers**. Our analysis will proceed with the use of appropriate and sound statistics, following the elements provided in class. It is clear that it will be hard to finish all the exercises in two hours, so we may have to split the tasks among the groups, and than share the results among all of you.

4.2 Practice

4.2.1 JBN: Binary Representation

Here we will use a binary representation for the JBN problem, with random initial population of fixed size (e.g., 150), deterministic tournament selection of the parents (e.g., tournament size of 3), flip mutation, one point crossover, generational with elitism survival's selection (e.g., elite size of 10%), and stop criterion fixed by the number of generations (e.g., 1500). Use a chromosome size of 100. The code for the simple evolutionary algorithm (`sea_bin`) was already provided.

Problema 4.1 M

Let's study the importance of the probability of mutation for the quality of the result. For that, we define several values, for example $p_m = \{0.01, 0.05, 0.1\}$. We keep the probability of crossover the same for all experiments (e.g., 70%). We run 30 times[1] for each situation and keep in a file the results, i.e., the **best at the end of the run**. Then we must choose the appropriate statistical test, use it, and state the conclusions.

Problema 4.2 M

Some times the results attained by the different configurations and/or algorithms are not statistically significantly different. Nevertheless, it may happen that one approach is more **efficient** than the others. Repeat the previous experiment to study that issue. Now the value to store for each run is the **number of the generation** where we found the best individual. Note that you may have to modify the code that was provided.

4.2.2 JBN: Integer Representation

Being creative is one major characteristic of humans. Let's propose a different approach for the JBNP, both in terms of representation and variation operators. Now, an individual will be **represented** by a set of integers, i.e., the genotype is equal to the phenotype. **Mutation** will have three possibilities: changing one value, adding one value, or deleting one value. For the **crossover**, we take two parents, define a size n, based on the length of the parents, sample each parent choosing n elements and construct an offspring equal to **union** of the two samples. It seems a crazy approach . . . but one never knows when it turn out to be a good one!

Problema 4.3 D

Devise an EA for that representation and operators. Try to identify values for the parameters that give a reasonable solution, based on a population of small size, and a small number of generations. That is, construct a solution that works . . . .

Problema 4.4 M

[1] Be aware that this experiment may take a lot of time. So, to do this during the class you must reduce the number of runs to a much lower number, say 10.

What solution for the JBNP is best? The one using a binary representation or the one based on a representation consisting of a set of integers? To clarify this issue, use the best result for the binary representation and the one you choose for the integer representation, and do a proper statistical analysis.