Evolutionary Algorithms: Peer review report

For the group of Andru Onciul, Alexios Konstantopoulos, and Alexandros Kyrloglou

Awouters Andreas, Christiaensen Hanne, Claessens Nina and Van Steenbergen Tom

November 19, 2021

1 One strong aspect

Removing one element from your representation is a small but smart optimization. I assume you store every path in a list, creating a matrix. This is probably a good idea, as operations on matrices are fast. The numpy library has low level c-implementation of matrix operations, making them even faster.

2 Three weak aspects

The following aspects might be weak points in your algorithm:

1. Complex recombination operator: slow computation
2. Having a constant number of mutations: static programming
3. Duplicate paths: fast convergence and stagnation

2.1 Complex recombination operator

This is an interesting recombination operator. But it seems quite complex. I would recommend looking at simpler operators. Maybe dynamically switch depending on the population.

2.2 Using different mutation options

Scramble mutation can be useful if the algorithm is about to converge. It might be an idea to switch between swap and scramble mutation depending on the population. Currently, you chose a fixed number of paths that are to be mutated. It might be useful to do this in a probabilistic way. Mutate every path with a predefined probability.

2.3 Duplicate paths

Having duplicate paths might become a problem. If you keep selecting the same paths over and over again, you will converge more quickly to a solution. This is not always a good thing, as this can happen for bad paths. On the other hand, checking for duplicates is a computationally heavy task. The graph in your report converges quite quickly. This could be because of all the duplicate paths in your population. You could check for duplicates, or increase the mutation probability when you detect stagnation. With the smart convergence test, you could predict stagnation.

3 One suggestion

Keep in mind that your program will run for 5 minutes. Introducing more diversity is a good solution, as stated in the report. But also, take care of your duplicate paths.

The convergence criterion is very smart. This method uses a lot of history, compared to just looking at the previous value. You might be able to use this as a method for predicting when you will converge. Convergence might be an indicator of stagnation, so you could use linear regression to dynamically switch operators. This means that when you have exploited the local space, you switch to a more global exploration mode.