Evolutionary Robotics
Prof. Dario Floreano

Evolutionary Robotics Laboratory

Exercise Sheet 4: Chase a goal on flat ground

Euan Judd (euan.judd@epfl.ch)
Krishna Manaswi Digumarti (krishna.digumarti@epfl.ch)

Goal.

Use RoboGen to evolve robots able to solve more complex tasks. The evolved robot must follow a signal, in this case a moving light, in the environment.

Learning objectives.

In this laboratory, you should learn:

- How to co-evolve controllers and morphologies for more complex tasks that include sensor readings.
- How to use penalization in fitness functions.
- How to evolve robots capable of making the jump from simulation to reality, i.e. how to avoid evolving robots that exploit some feature of the simulation and therefore don't work in the real world.

Assignments.

- Go through the provided PDF instruction file during the laboratory class.
- Every team will submit the best robot (text file) found with evolution on moodle by the 5th of May. **NOTE:** The performance of the robot will not be graded.

Getting Started

To get started, visit http://robogen.org/app and upload the files provided in moodle into Robogen2022/es4.

Tip if the software is having problems, try refreshing the page.

Important:

- Remember, all data is being saved to a virtual filesystem within your web browser. If you want to save anything for later use, be sure to download it to your home directory!

- As mentioned previously, this year (including the Robogen Grand Challenge) you will not be allowed to use wheels. The available parts are for this exercise sheet are the ***CoreComponent, FixedBrick, ParametricJoint, PassiveHinge, ActiveHinge, LightSensor****, and* ***IrSensor*** (if you specify addBodyPart=All when evolving morphologies, these will be the parts that your robots will be composed of).

Exercise 1

You will now perform a body-brain evolution of a robot that makes more complex usage of sensors.

Try a simulation with the **es4/myConf.txt**, you will see an example of an arena where a fixed light is placed. The command lightSourcesConfigFile = lights.txt is used to add information on where the light is placed. See the documentation on the Robogen website for more details. Scenario = chasing is used in the myConf file, this is a custom scenario using a fitness function that will try to minimize the distance between the robot and the light. However, if the light is always in the same position during evolution, it is possible that the robot will just learn to move along a specific path rather than use its light sensors to detect where the light source is and navigate towards it. To avoid this, different starting positions should be used during evolution. Alternatively, use the provided **myConf_movingLight.txt** file which uses the **arena2-scenario-random.js** scenario file**.** In this scenario, the light moves in random directions during the simulation so only robots that utilize their light sensors will be able to chase it.

How to proceed?

It is difficult to evolve a robot for two different tasks simultaneously, e.g. the locomotion and light following tasks in this example. We therefore suggest that you perform a **multi-step evolution**. Multi-step evolution is usually performed when multiple distinct behaviours should be evolved in a robot. As the name implies, multiple evolutionary runs are performed in series, each step uses a different fitness function to evolve the robot for one of the desired behaviours. At the end of each step, the best robot from an evolutionary step is given as a json file which needs to be converted (using **json_converter.py**) to a txt file so it can be input as the starting point in the next step of the evolution. For example, in this problem a first step could be to evolve a robot able to locomote and turn, a second step would be to evolve a robot able to locomote and follow a light thanks to the usage of light sensors in the body and the relative adjustment of the NN controller.

NOTE: A possible strategy is to evolve a locomoting robot first and then allow a body-brain evolution that will add only light sensors as body components.

You may see the fitness plateau after a certain number of generations. If this happens, try the following:

- **Keep the evolution running for more generations. Big improvements can sometimes happen after 50-100 generations, even after the fitness seems to have plateaued already.**
- **Explore the fitness landscape more, i.e. increase the population size, mutation rate, crossover rate, or make the tournament size smaller.**
- **Improve the fitness function. For instance, modify the equation by changing some of the terms or give weights to different terms in the existing fitness function to change their relative importance.**

Exercise 2

Real sensor data usually includes some noise. In addition, physical phenomena that are not modelled in the simulator may influence the real system, e.g. reflections, different lights in the environment, or inaccurate sensors. Therefore, to make the evolved robot generalizable when transferring the model from simulation to reality, noise should be added to sensor readings in the simulator. Try to add noise of 0.1 to the sensors and check if the performance is still good. If not, try an additional step of the multi-step evolution where you perform a brain only evolution in the presence of sensor noise.

NOTE: See the documentation to add sensor noise.

You can now try to analyse the performance of the multi-step evolution by putting all the data from the different steps together in a single file and opening it with the plot_results.py file provided in Exercise 1.

Exercise 3 (optional)

If your robot can follow the light when the terrain is flat, you could try to evolve a new one using an arena with a movable light and a more challenging terrain of your choosing.

Exercise 4

Submit the final robot that you evolved to locomote and follow a light source in presence of sensors noise. It should be submitted as a txt file so use **json_converter.py** to convert the json file to txt. If you haven't already, use plot_results.py to analyse your evolution.

Good luck with your evolution!