Modern Robots: Evolutionary Robotics

Programming Assignment 8 of 8*

Description

In this assignment you will employ a multi-objective evolutionary algorithm to evolve organisms that are both long (in any direction) and that move as far as possible.

Tasks

1. Copy all your code for assignment 7 to the folder you will use for assignment 8. Make sure to keep a working backup of assignment 7.

2. Extract `hw_8_code.zip` and move all files to your assignment 8 folder.

3. Open `EA_MultiObj.hpp` and implement the `init` and `epoch` functions according to an NSGA-II based multi-objective algorithm. The algorithm will use Pareto-based tournament selection for parent selection and elitist Pareto-based rank selection for survivor selection. You do not have to implement crowding, but you have to make sure that, if there are multiple individuals occupying the exact same position on the Pareto front, only one of those individuals is added to each layer. Note that, because the selection operators are very specific for this algorithm, we will not pass selectors to the algorithm.

4. Next, implement the `evaluate` function in `Fit_MultiObjDistLong.hpp` as a fitness function that measures both the distance that a robot can move, as well as the maximum distance between any of its parts. Adding these measures as objectives to be maximized should result in robots that range from being very long (in any direction) to being very fast.

5. Now it is time to evolve the robots. Open `main.cpp`, comment out or delete all previous code (backup!) and include all necessary header files.

Use the following type as your individual:

```
typedef IndivMultiObj<EvolvingRobot> indiv_t;
```

*Original material was graciously provided by Josh Bongard. Jeff Clune slightly modified it. Joost Huizinga heavily modified it.

Start with the following parameters:

- Population size: 50
- Generations: 25

While running the algorithm, log the entire Pareto front at each generation. This means you will have to write the following information at each generation: for each individual on the Pareto front write the performance on the first and second objectives to the log file, separated by a space. Write each generation to a single line. For example, if the current population has two individuals on the Pareto front, where the first individual has the performance (1.72, 0.2), and the second individual has the performance (0.76, 0.95), you should write the line:

```
1.72 0.2 0.76 0.95
```

Plot the resulting file with `plotPareto.py` and add the result to your document. The result should look like figure 1.

Lastly, visualize the entire Pareto front in a single simulator window. To do so you can use the following code:

```
sim.deleteRobots();
ea_t::pop_t paretoFront = ea.getParetoFront();
lexicalSort<indiv_t> sort;
std::sort(paretoFront.begin(), paretoFront.end(), sort);
btVector3 center(0,0,0);
btVector3 incr(2.5,0,0);
btVector3 current = center + -incr*btScalar(paretoFront.size()-1)*0.5;

for(size_t i=0; i<paretoFront.size(); ++i){
    sim.buildRobot(paretoFront[i], current);
    current += incr;
}

sim.getDynamicsWorld()->setDebugDrawer(&gDebugDrawer);
glutmain(argc, argv,640,480,"Simulator",&sim);
```

Where `sim` is an instance of the simulator, `ea` is an instance of the evolutionary algorithm, and `ea_t` is the type of the evolutionary algorithm.

Take a screenshot in which you can see all individuals on the Pareto front and add it to your document. This figure should look like figure 2. Unpause the simulation and wait until all organisms have moved some distance. Pause the simulation, make another screenshot, and add it to your document. This figure should look like figure 3.

Deliverables

A pdf document containing the figures resembling figures 1, 2 and 3 and any files you changed.

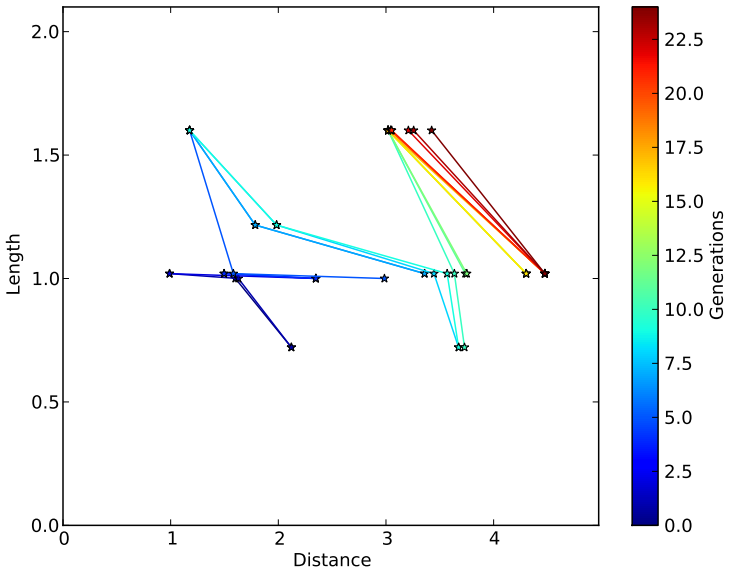

Figure 1: The Pareto front over time. Note that your Pareto front may be much wider.

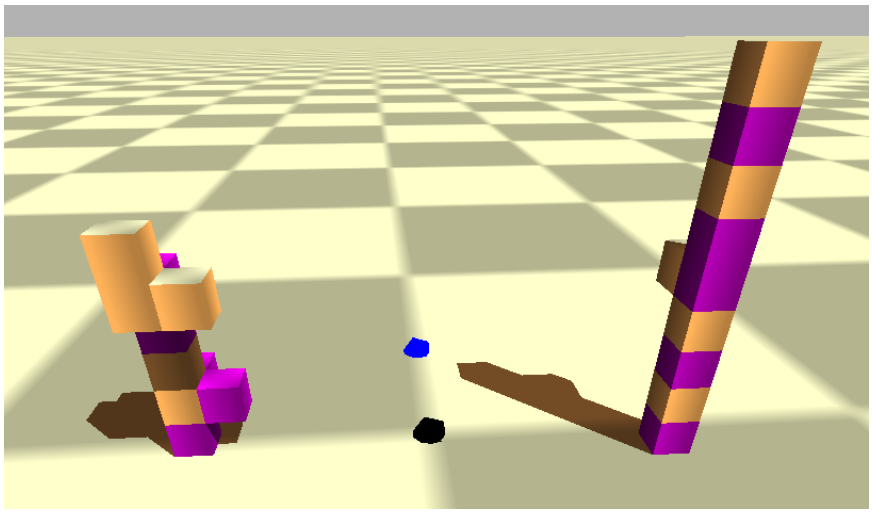

Figure 2: The final individuals on the Pareto front. Your individuals may look quite different and you may have more than just two individuals.

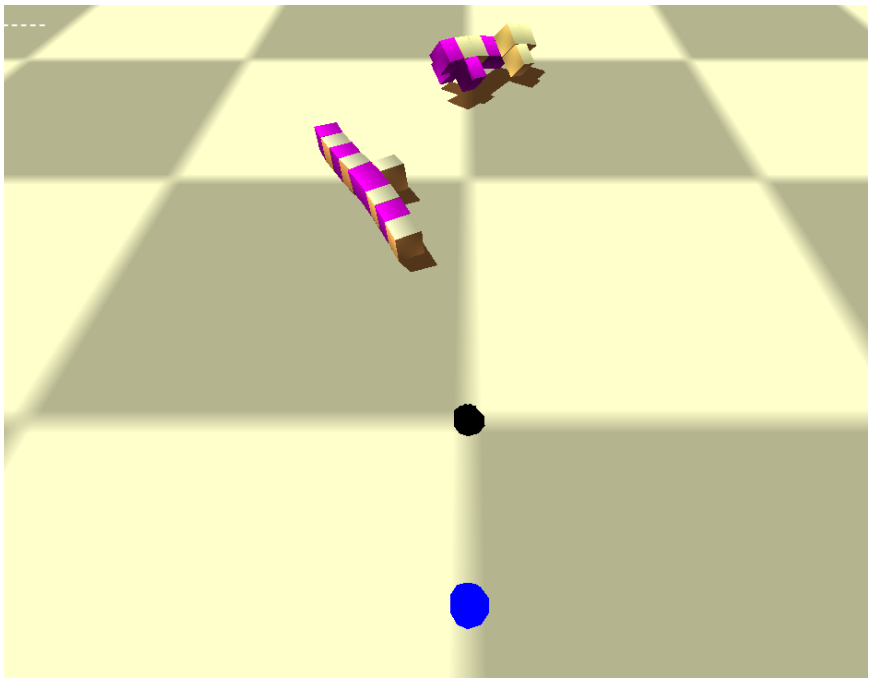

Figure 3: The final individuals after moving some distance. Your individuals may look quite different and you may have more than just two individuals.