

# Introduction to Complexity

## Unit 5 Homework

(optional)

**Note that this week we will not be providing solutions to the Homework, since the exercises are all open-ended experiments.**

(There is a video in the next segment with hints on how to do Intermediate Question 1.)

You can choose whether you want to do the Beginner, Intermediate, or Advanced level. You are free to discuss this part of the homework with anyone, or to ask questions about it on the course forum. You can also use any resources you like, including our course NetLogo models. Note that homework in this course is not turned in or graded.

### **Beginner Level:**

**1. Experiment with Different Mutation Rates:** Download RobbyGA.nlogo from the Course Materials page. Set the speed slider to its fastest value, and turn off “view updates” (checkbox next to speed slider). This will allow the program to run at its fastest speed.

For each parameter setting below, run the GA five times: for each run, click Setup, then go-n-generations. For each run, record the final best fitness in the population (from the output window on the right side of the interface) in the table below. Each run may take a long time (one run with the first parameter setting took six minutes on my computer). You may want to do something else while you’re waiting for each run to finish! If you want, you can automate doing these runs via NetLogo’s Behavior Space module: <http://ccl.northwestern.edu/netlogo/docs/behaviorspace.html>

Then fill in the rest of table below.

Number of Generations	Population Size	Mutation Rate	Best Fitness in Final Population	Average Best Fitness in Final Population (over 5 runs)
200	100	0.05	Run1:	
			Run2:	
			Run 3:	
			Run 4:	
			Run 5:	
200	100	0.1	Run1:	
			Run2:	
			Run 3:	
			Run 4:	
			Run 5:	
200	100	0.01	Run1:	
			Run2:	
			Run 3:	
			Run 4:	
			Run 5:	

What effect do you see on the final best fitness with the different mutation rates?

## 2. Experiment with Different Population Sizes:

Do a similar set of experiments testing the effect of increasing the population size.

### Intermediate Level:

1. **Experimenting with “hidden” parameters.** The program RobbyGA.nlogo has a number of parameters that are set “behind the scenes”. Some of these are the rewards and penalties for various actions (`wall-penalty`, `can-reward`, `pick-up-penalty`), the number of “environments” each strategy is tested on to compute its

fitness (num-environments-for-fitness), and the number of actions each strategy takes in each environment (num-actions-per-environment). Modify the code so that these parameters can be set by the user (via sliders in the interface). Experiment with different settings of these parameters. (If you haven't already, take a look at the "behavior space" module of Netlogo, that allows you to do experiments with many parameter settings: <http://ccl.northwestern.edu/netlogo/docs/behaviorspace.html>).

Experiment to see if and how changing these parameters changes the behavior of the best evolved strategies.

**2. Allowing user to set crossover probability.** In the current version of RobbyGA.nlogo, each child is created via a crossover between the parents. However, in many GAs, the user can set a "crossover probability", which gives the probability that two parents will create a child via crossover, or via cloning of one of the parents. Modify RobbyGA.nlogo to include a user-settable crossover probability. Then, using the Behavior Space module, test several crossover probabilities (with a set population size and mutation rate) to see the effect of crossover probability on the performance of the GA.

### **Advanced Level:**

**1.** Read CommonSelectionMethods.pdf (on the Course Materials page) Note that RobbyGA.nlogo currently uses a "tournament selection" method for choosing parents to create offspring. Modify RobbyGA.nlogo so that the user can choose from a variety of selection methods (using a drop-down menu on the interface). The ones I suggest that you implement are:

Fitness-Proportionate Selection (with Roulette-Wheel sampling)

Elitism: At each generation, copy the best individual from the previous generation

Linear Rank selection

Also, for tournament selection, allow the user to set the number in the tournament.

Experiment with these various selection methods—does the particular selection method make a big difference in the GA's performance for a given set of parameter settings?

**2.** Modify RobbyGA.nlogo to allow users to put "walls" in any square in Robby's world. Experiment to see how putting in walls in different places affects the fitness and behavior of the best individuals.