

Homework Set 2

Due date: Wednesday, Jan. 22.

Written Exercises

HW2.1. Textbook, Chapter 2, Thought Exercise 1. Check your answer with the answers in the “Selected solutions” handout. Turn in answer and how you obtained it. (10 points)

HW2.2. Define a function set and terminal set that you could use to represent strategies for the Prisoners Dilemma (with a memory of three previous games—i.e., 64 bits in the traditional GA representation) as parse trees that could be used by a genetic programming algorithm. (Hint: You can use something similar to the solution of Thought Exercise 1 in Chapter 2 in the textbook.) Use these functions and terminals to construct a parse tree that represents the TIT-FOR-TAT strategy. Note that there is no single correct answer here; many different solutions could be correct. (10 points)

HW 2.3. Textbook, Chapter 2, Thought Exercise 3. Check your answer with the answers in the “Selected solutions” handout. Turn in answer and how you obtained it. (10 points)

HW2.4. Consider the schema $1^{***}\dots^{**}0$ (128 loci). Interpret instances of this schema as cellular automaton rules, as described in the readings and lecture. Prove (in words) that the average fitness of instances of this schema on the density-classification task for cellular automata (the task described in the book and in the lecture) is zero. (10 points)

Computer Exercises

HW2.5. Run the Prisoner’s Dilemma code with the parameters given in the files in

```
nfs: /home/projects/class/cse580ec/prisoners-dilemma
cifs: \\hopper\clss-prj\cse580ec\prisoners-dilemma
```

Do 10 runs, and find the single best individual evolved by the GA (set the `LONG_PRINT` option to `TRUE` to save the chromosomes). Write a function that will calculate this individual's total score when run against TIT-FOR-TAT over 10 games (memory of last three moves). How does this total score compare with its total score when run for 10 games against a single randomly generated strategy?

Turn in your new version of `fitness.c`, plus the scores calculated by your code.

(20 points)

HW2.6. Modify the Prisoner's Dilemma code so that the fitness of an individual is the total score over 10 games against TIT-FOR-TAT. (In `fitness.c`, you will need to set `NUM_TRAINING_STRATEGIES` to 1, and replace the code that generates random strategies with code that generates TIT-FOR-TAT's bit string as `training_strategies[0].chrom.`) Do 10 runs of the GA, and find the highest fitness individual. What is this individual's score over 10 games with a randomly generated strategy? What is its score over 10 games with the "always defect" strategy (all 1s)?

Turn in your new version of `fitness.c`, plus the scores calculated by your code.

(20 points)

HW2.7. Modify the Prisoner's Dilemma code so that the fitness of an individual is the total score over 10 games each against the 20 individuals in the population, including the individual itself. (In `fitness.c`, you will need to set `NUM_TRAINING_STRATEGIES` to 20, and replace the code that generates random strategies with code that copies the current population into the training strategies.) Do 10 runs of the GA, and find the highest fitness individual evolved during these runs. What is this individual's score over 10 games with a randomly generated strategy? What is its score over 10 games with the "TIT FOR TAT" strategy? What is its score over 10 games with the "always defect" strategy (all 1s)?

(20 points)

Turn in your new version of `fitness.c`, plus the scores calculated by your code.