

Application of Genetic Programming to the "Snake Game"

Problem: The fundamental problem of the snake game is to eat the maximum number of food pieces before "dying" by running into either a wall or a segment of the snake's body. The problem being addressed is to provide a function and terminal set that will allow for the evolution of a GP that will maximize the number of food pieces eaten by the snake. The maximum goal for the particular configuration of the snake game used in the paper is 211 pieces of food ($220 - 9 = 211$, where 220 is total board position and 9 is initial snake size).

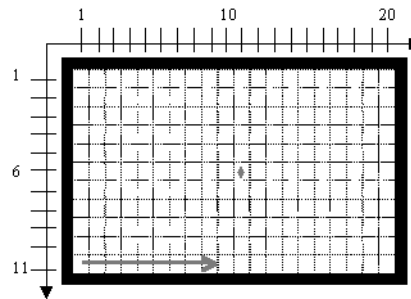


Figure 1: Snake game board

Objective:	Find a computer program that eats the maximum possible pieces of food.
Terminal set:	(forward), (left), (right)
Function set:	ifFoodAhead, ifDangerAhead, ifDangerRight, ifDangerLeft, progn2
Fitness cases:	One fitness case.
Raw Fitness:	Pieces of food eaten.
Standardized fitness:	Maximum possible pieces of food eaten (211) minus the raw fitness.
Hits:	Total pieces of food eaten during a run of the program, same as raw fitness.
Parameters:	Population = 10000. Max generations = 500.
Success predicate:	A program scores 211 hits.

Additional Functions: *ifDangerTwoAhead*, *ifFoodUp*, *ifFoodRight*, *ifMovingRight*, *ifMovingLeft*, *ifMovingUp* and *ifMovingDown*.

Fitness Cases: For initial runs of the problem, only a single fitness case was used to determine the fitness for each individual. Because the food placement is random both during a single run, and from one run to another, occasionally individuals would score a number of hits because of fortuitous placement of the food, and not as much on the merit of their function tree.

To better ensure that the most successful individuals achieved high fitness measures primarily on the basis of their function tree, new GP runs were often made featuring a "**primed**" population in which the fitness was measured as the average of four runs of an individual. The procedure for this is as follows: once a run had completed without obtaining a solution, or if a run had stalled on a single individual for a large number (100 or more) of generations, a new run was begun with this final individual as one of the initial individuals. For this new run, however, the fitness was taken as the average fitness of an individual over four runs instead of merely a single run. The averaging of the fitness over four runs helped eliminate the possibility of an individual having a high fitness due simply to lucky placement of the food. Using this averaging method to determine fitness was only used in primed populations because it increased the time of a GP run fourfold. Furthermore, it was common for the generations that timed out to feature an individual who had scored a high fitness as a result of a lucky run. By beginning a new run with this individual in the initial population, it not only assured a more realistic fitness measure, but it introduced an entirely new mix of randomly generated schemata that could potentially benefit the stalled individual. Details of results produced by primed runs are given in the results section.

Results: There were three types of GP runs made in an attempt to evolve a solution to the snake game: runs using the initial function set, the final function set, and primed runs, also using the final function set. The highest number of hits generated by a run using the initial function set was 123. Three separate solutions were generated using the final function set, although none of them were found to consistently generate a solution. The number of hits achieved by each solution depended on the placement of the food. It was not until the method of "priming" a run, was used that a consistent solution was generated. Of ten primed runs, using various initial seeds, exactly five of them evolved a solution, all of which were consistent solutions over multiple runs. Comparatively, over twenty runs using the full function set were made, and only three of them produced solutions, none consistent.

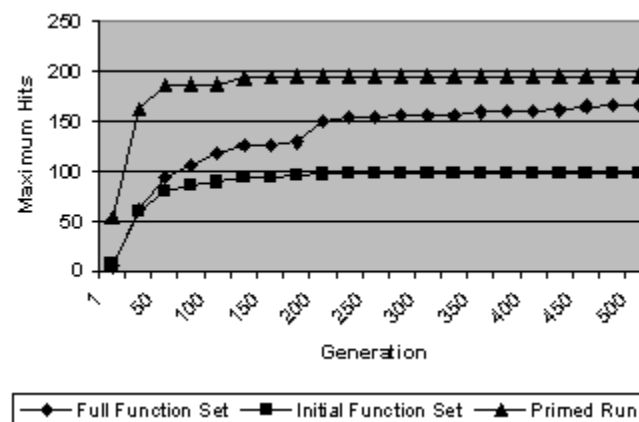


Figure 2: Average result of ten runs

By Tobin Ehlis