CS5401 FS2018 Assignment 2c Report

Ruicheng Xu
rx8qd@mst.edu

Missouri S&T

Methodology:

This section the author will explain the implementation of the EA in two major aspects, including GP tree terminal and function set, tree evaluation and its related optimizations.

GP tree terminal and function set

In this assignment the pacman and the ghosts are controlled by GP trees which provides a numerical evaluation of a given control options, the highest valued option will be chosen to be the next move of pacman and ghost respectively. The GP Tree used in this experiment series is a binary parse tree consists of binary operator functions on non leaf nodes and terminal functions on the leaf nodes. GP Tree of Ghost and Pacman are identical except with different Terminal Set. The specific function and terminal descriptions can be found in the assignment description.

Tree evaluation and its related optimizations

Since GP tree(parse tree) is used to evaluate control options in every single step of the game for all roles(Pacman and Ghosts), it is the most used portion of the entire program, thus the following optimization approach is being used to speed up the program:

- **Dynamically Compiled Function from Parse Tree –** Compared to the normal interpreter approach for tree evaluation, the author designed the program in such a way that the parse tree will be compiled and defined into a python function(which will be directly excuted by the python interpretor) right before every game play, which significantly reduced the runtime.
- **Terminal Caching –** The terminal result is being cached within a single game step, as is is computationally expensive to calculate the terminal result and each parse tree is likely to have identical terminal function on multiple leaves. Such approach is also expected to reduce the run time.

Experimental Setup:

In this assignment, every experiment is controlled and specified by a set of control variables stored as a configuration file. There many variables within a configuration file, such as Pacman/Ghost stable population size, Pacman/Ghost offspring population size, Pacman/Ghost parent population size, and various Parent/Survival selection methods with corresponding parameters. To reproduce a specific experiment, one can find the configuration data from the beginning of the log file as a python dictionary format(this includes the specific seed for a specific experiment) and write it in a standalone .cfg file, then call python3 run.py -f [config file name] to reproduce the experiment.

Results:

In order to describe each experiment's configuration in a clear and organized manner, a table contains configuration details of all experiments conducted by the author is constructed below.

Experiment Name	Pacman Stable Population Size	Pacman Offspring Population Size	Pacman Mating Pool Size	Ghost Stable Population Size	Ghost Offspring Population Size	Ghost Mating Pool Size	Pacman Controller	Ghost Controller	number of Pacman	number of Ghost
formal_exp_0	5	50	2	5	50	2	shared	shared	1	3
formal_exp_1	5	50	3	5	50	2	shared	shared	1	3
formal_exp_2	5	50	4	5	50	2	shared	shared	1	3
formal_exp_1_offspring_100	5	100	3	5	100	2	shared	shared	1	3
formal_exp_1_offspring_150	5	150	3	5	150	2	shared	shared	1	3
bonus_1	5	50	4	5	50	3	shared	individual	3	3
bonus_2a	5	50	4	5	50	4	shared	shard	3	3
bonus_2b	5	50	4	5	50	5	individual	shared	3	3
bonus_2c	5	50	4	5	50	6	shared	individual	3	3
bonus_2d	5	50	4	5	50	7	individual	individual	3	3

After a brief testing of several different parameters, the author decided to run a set of experiments to discover the impact of different parent population size(mating pool size) on the Pacman performance. Different parent population also relates to different parent selection pressure (smaller the mating pool size, higher the pressure).

The following three experiments results are being compared: formal_exp_0, formal_exp_1 and formal_exp_2(using mating pool size 2,3 and 4 respectively)please refer the table for detailed configuration. The highest Pacman scores within the last generation of each run is being recorded and compared using the following statistical test.

Treatment 1 formal_exp_0(mating pool size 2) *Treatment 2* formal_exp_1(mating pool size 3)	*Treatment 1* formal_exp_1(mating pool size 3) *Treatment 2* formal_exp_2(mating pool size 4)	*Treatment 1* formal_exp_0(mating pool size 2) *Treatment 2* formal_exp_2(mating pool size 4)
Treatment 1 N_1: 30 $df_1 = N - 1 = 30 - 1 = 29$ M_1: 33.43 SS_1: 14633.37 $s^2_1 = SS_1/(N - 1) = 14633.37/(30-1) = 504.6$ *Treatment 2* N_2: 30 $df_2 = N - 1 = 30 - 1 = 29$ M_2: 28.4 SS_2: 12761.2 $s^2_2 = SS_2/(N - 1) = 12761.2/(30-1) = 440.04$ <u>T-value Calculation</u> $s^2_p = ((df_1/(df_1 + df_2)) * s^2_1) + ((df_2/(df_2 + df_2)) * s^2_2) = ((29/58) * 504.6) + ((29/58) * 440.04) = 472.32$ $s^2_{M1} = s^2_p/N_1 = 472.32/30 = 15.74$ $s^2_{M2} = s^2_p/N_2 = 472.32/30 = 15.74$ $t = (M_1 - M_2)/\sqrt{}(s^2_{M1} + s^2_{M2}) = 5.03/\sqrt{}31.49 = 0.9$ The *t*-value is 0.89698. The *p*-	*Treatment 1* N_1: 30 $df_1 = N - 1 = 30 - 1 = 29$ M_1: 28.4 SS_1: 12761.2 $s^2_1 = SS_1/(N - 1) = 12761.2/(30-1) = 440.04$ *Treatment 2* N_2: 30 $df_2 = N - 1 = 30 - 1 = 29$ M_2: 35.07 SS_2: 10031.87 $s^2_2 = SS_2/(N - 1) = 10031.87/(30-1) = 345.93$ <u>T-value Calculation</u> $s^2_p = ((df_1/(df_1 + df_2)) * s^2_1) + ((df_2/(df_2 + df_2)) * s^2_2) = ((29/58) * 440.04) + ((29/58) * 345.93) = 392.98$ $s^2_{M1} = s^2_p/N_1 = 392.98/30 = 13.1$ $s^2_{M2} = s^2_p/N_2 = 392.98/30 = 13.1$ $t = (M_1 - M_2)/\sqrt{}(s^2_{M1} + s^2_{M2}) = -6.67/\sqrt{}26.2 = -1.3$ The *t*-value is -1.30247. The *p*-	*Treatment 1* N_1: 30 $df_1 = N - 1 = 30 - 1 = 29$ M_1: 33.43 SS_1: 14633.37 $s^2_1 = SS_1/(N - 1) = 14633.37/(30-1) = 504.6$ *Treatment 2* N_2: 30 $df_2 = N - 1 = 30 - 1 = 29$ M_2: 35.07 SS_2: 10031.87 $s^2_2 = SS_2/(N - 1) = 10031.87/(30-1) = 345.93$ <u>T-value Calculation</u> $s^2_p = ((df_1/(df_1 + df_2)) * s^2_1) + ((df_2/(df_2 + df_2)) * s^2_2) = ((29/58) * 504.6) + ((29/58) * 345.93) = 425.26$ $s^2_{M1} = s^2_p/N_1 = 425.26/30 = 14.18$ $s^2_{M2} = s^2_p/N_2 = 425.26/30 = 14.18$ $t = (M_1 - M_2)/\sqrt{}(s^2_{M1} + s^2_{M2}) = -1.63/\sqrt{}28.35 = -0.31$ The *t*-value is -0.30676. The *p*-

value is .373437. The result is *not* significant at $p < .05$.	value is .197905. The result is *not* significant at $p < .05$.	value is .760128. The result is *not* significant at $p < .05$.

The following plot visualize the performance distribution of experiments with three different pacman mating pool population sizes

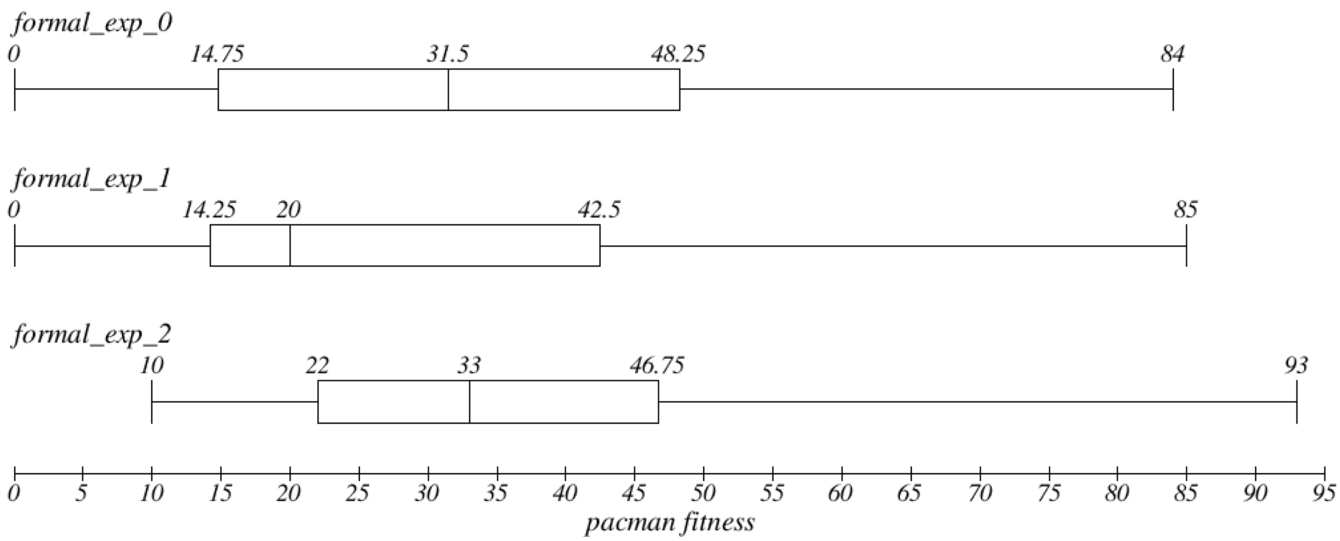

Illustration 1: Pacman performance distribution of three different mating pool size

Unfortunately none of the comparison group shows a significant result.

The author then conducted another set of experiments observe the impact of Pacman performance with three sets of different Pacman and Ghost offspring population.

The following three experiments results are being compared: formal_exp_1, formal_exp_1_offspring_100 and formal_exp_1_offspring_150
(using offspring size 50,100 and 150 respectively for both Pacman and Ghost)please refer the table for detailed configuration. The highest Pacman scores within the last generation of each run is being recorded and compared using the following statistical test.

Treatment 1 formal_exp_1 (50 offsprings) *Treatment 2*	*Treatment 1* formal_exp_1_offspring_100 *Treatment 2*	*Treatment 1* formal_exp_1 (50 offsprings) *Treatment 2*

formal_exp_1_offspring_100	formal_exp_1_offspring_150	formal_exp_1_offspring_150
Treatment 1 N_1: 30 $df_1 = N - 1 = 30 - 1 = 29$ M_1: 28.4 SS_1: 12761.2 $s^2_1 = SS_1/(N - 1) = 12761.2/(30-1) = 440.04$ *Treatment 2* N_2: 30 $df_2 = N - 1 = 30 - 1 = 29$ M_2: 42.07 SS_2: 22023.87 $s^2_2 = SS_2/(N - 1) = 22023.87/(30-1) = 759.44$ T-value Calculation $s^2_p = ((df_1/(df_1 + df_2)) * s^2_1) + ((df_2/(df_2 + df_2)) * s^2_2) = ((29/58) * 440.04) + ((29/58) * 759.44) = 599.74$ $s^2_{M1} = s^2_p/N_1 = 599.74/30 = 19.99$ $s^2_{M2} = s^2_p/N_2 = 599.74/30 = 19.99$ $t = (M_1 - M_2)/\sqrt{(s^2_{M1} + s^2_{M2})} = -13.67/\sqrt{39.98} = -2.16$ **The t-value is -2.16135. The p-value is .034809. The result is significant at $p < .05$.**	*Treatment 1* N_1: 30 $df_1 = N - 1 = 30 - 1 = 29$ M_1: 42.07 SS_1: 22023.87 $s^2_1 = SS_1/(N - 1) = 22023.87/(30-1) = 759.44$ *Treatment 2* N_2: 30 $df_2 = N - 1 = 30 - 1 = 29$ M_2: 42.77 SS_2: 14327.37 $s^2_2 = SS_2/(N - 1) = 14327.37/(30-1) = 494.05$ T-value Calculation $s^2_p = ((df_1/(df_1 + df_2)) * s^2_1) + ((df_2/(df_2 + df_2)) * s^2_2) = ((29/58) * 759.44) + ((29/58) * 494.05) = 626.75$ $s^2_{M1} = s^2_p/N_1 = 626.75/30 = 20.89$ $s^2_{M2} = s^2_p/N_2 = 626.75/30 = 20.89$ $t = (M_1 - M_2)/\sqrt{(s^2_{M1} + s^2_{M2})} = -0.7/\sqrt{41.78} = -0.11$ The t-value is -0.10829. The p-value is .914137. The result is *not* significant at $p < .05$.	*Treatment 1* N_1: 30 $df_1 = N - 1 = 30 - 1 = 29$ M_1: 28.4 SS_1: 12761.2 $s^2_1 = SS_1/(N - 1) = 12761.2/(30-1) = 440.04$ *Treatment 2* N_2: 30 $df_2 = N - 1 = 30 - 1 = 29$ M_2: 42.77 SS_2: 14327.37 $s^2_2 = SS_2/(N - 1) = 14327.37/(30-1) = 494.05$ T-value Calculation $s^2_p = ((df_1/(df_1 + df_2)) * s^2_1) + ((df_2/(df_2 + df_2)) * s^2_2) = ((29/58) * 440.04) + ((29/58) * 494.05) = 467.04$ $s^2_{M1} = s^2_p/N_1 = 467.04/30 = 15.57$ $s^2_{M2} = s^2_p/N_2 = 467.04/30 = 15.57$ $t = (M_1 - M_2)/\sqrt{(s^2_{M1} + s^2_{M2})} = -14.37/\sqrt{31.14} = -2.57$ **The t-value is -2.57468. The p-value is .012608. The result is significant at $p < .05$.**

The following plot visualize the performance distribution of experiments with three different offspring population sizes.

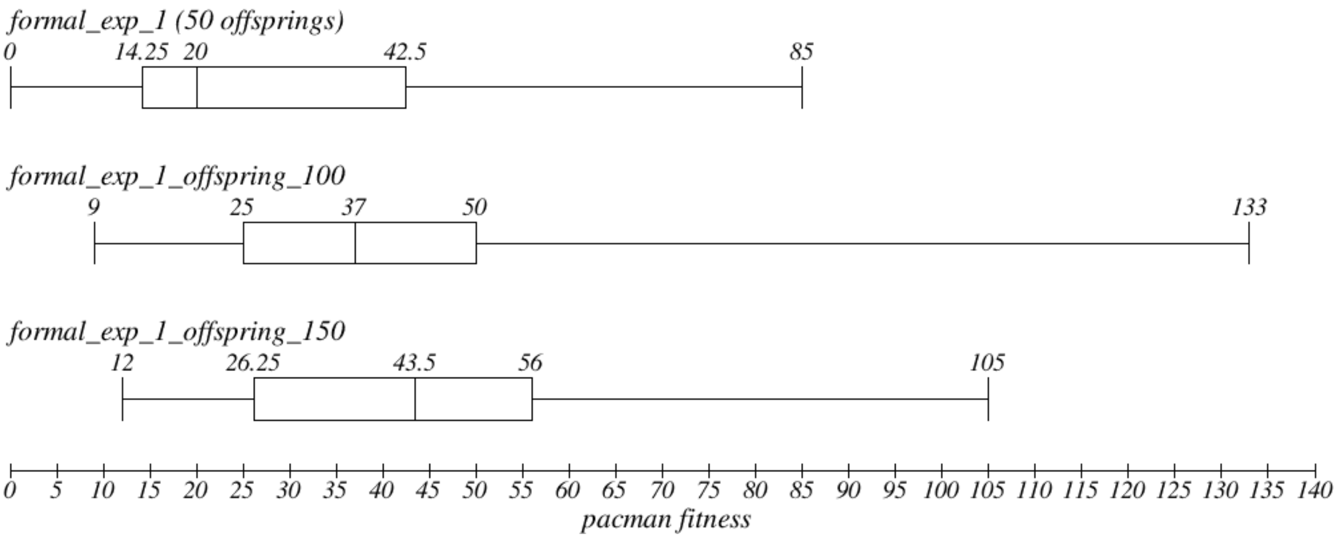

Illustration 2: Performance distrbution of three different offspring population sizes

The significant t-test results for the first and third comparison groups suggest that by controlling mating pool size and stable population size, higher offspring population size(higher survival pressure) results in better performance of the Pacman. Although the second comparison group did not result in a significant t-test result, it can be observed from the box plot that futher increasing the offspring population is likely to have a higher Pacman performance.

Discussion:

From the two sets of experiments results, it is found that higher survival pressure results in a higher Pacman performance. Additionally it is observed that in order for Pacman or Ghosts to demonstrate a acceptable intelligent behavior, survival pressure has to be relatively high in the evolving process, to use the earlier experiment as an example, 5 out of 100 survives is a 5 percent survival rate while 5 out of 150 survive is a 3.3 percent survival rate. Such situation shows that only a small percentage of the Pacman within a generation has a intellegent behavior, this can also be shown by the Average vs Best performance graph in the appendix. This may be a result of a high variation step by the tree crossover and mutation.

Conclusion:

In the future experiments, a better parameter need to be set for a lower mutation step. The author also suggest that a better variation method for the parse tree can be researched for a smoother exploration.

In addition, it is observed from various the game plays that a dead locking situation(where pacman and ghost moving back and force, repeating the same motion), after investigation by the author, it is very likely caused by having all game roles(Pacman and Ghosts) to make control decisions at the same turn with the same game state information, interestingly and more importantly, it results in none of the party over powers the other and both party has a fair amount of score, such situation appeared to be an equilibrium state within evolution process, similar phenomenon in the biological world is described as "Evolutionary Stable Strategy" by evolutionary biologist Dr.Richard Dawkins in his book The Selfish Gene. In plain words, it is as if the thief and the bank vault guard made a deal with each other such that the thief only steal a small amount every time and the guard will let the thief get away so that the thief get its money and the guard does not lose his job.

Bibliography:

Dawkins, Richard, 1941-. (1989). The selfish gene. Oxford ; New York :Oxford University Press

Due to the non ideal parameters chosen at the time of running all the bonus experiments, the performance of pacman and ghosts does not appear to be very intelligent.The following discussion will mainly focus on the game play behaviors of the Pacman and the Ghost.

Bonus 1:

Compared to the shared controllers scheme for the ghosts, it is observed that there is more diversity in behavior for the ghost. As mentioned in the normal experiments report, dead locking behavior is likely appear; however, it appears that dead locking situation is less likey to happen or maintained as a ghost may suddenly decide to pursuit the Pacman in where other ghosts maintain a interlocking movement with the Pacman, such evolutionary equilibrium is broken.

Bonus 2 a b c and d

It can be observed that behavior of the Pacmen and Ghost in individual controllers are drastically different from shared controller. Each Pacman and Ghost in “Individual Controller Setup” has a obvious different behavior than the others as each controller employs a different control scheme(parse tree). Compared to “Shared Controller” where the Pacmans tend to stick together in the particular run being conducted. Another interesting observation is that in bonus 2 experiments, some pacman are working harder(smarter) than the others while some individuals appeared to be lazy and unintelligent, similar situation on the ghost side. It raises the concern that some ghost and Pacman individuals are getting “free rides” and the selection process is not a group selection approach(such as genes in a species are being selected in a same group). Thus a better selection process and fitness function are needed for this setup.