

COMP 6660 Fall 2022 Assignment 2c

Joshua Boyd
jcb0216@auburn.edu

November 25, 2022

1 Competitive Co-Evolution Results (Green)

1.1 Methodology

In this experiment, a separate ParseTree class was created to represent the controller tree. Within this tree, the following functional nodes were used:

Functional Symbol	Description
/	Standard division between two children node values
*	Standard multiplication between two children node values
+	Standard addition between two children node values
-	Standard subtraction between two children node values
RAND	Random number between two children node values

Table 1: Co-Evolution Parse Tree Functional Nodes

Additionally, the following terminal nodes were used:

Terminal Symbol	Description
G	Manhattan distance to nearest ghost
P	Manhattan distance to nearest pill
W	Number of immediately adjacent walls
F	Manhattan distance to nearest fruit
M	Manhattan distance to nearest player
#.#	A constant value of one or more digits

Table 2: Co-Evolution Parse Tree Terminal Nodes

For adversarial evaluations, each pac player controller from the pac population was paired with a ghost controller from the ghost population. In the case that there were not enough pac controllers in the pac population, the last controller in the pac population would play the remaining games with however many ghost controllers were left in the ghost population and vice versa. At the end of the experiment, the Pac player from the game’s final generation with the highest fitness plays an exhibition game with the ghost from the same run.

1.2 Experimental Setup

The experiment was conducted for 30 runs, with each run performing 2000 evaluations each. The configuration file used for the experiment can be found in `./configs/green2c_config.txt`. For this experiment, I chose the number of nodes as the parsimony pressure metric due to the tree depth metric taking longer per evaluation. I chose to give the ghost population a lower μ than the pac population. This was to give the pac player a bit of an evolutionary advantage by lowering the search space for the ghosts, since it was a 1 vs 4 game. I kept the 35% mutation rate for each population from the 2b experiment where it seemed to perform well.

1.3 Results

After running the experiment, statistical data was collected for both the pac and ghost populations. Table 3 on page 4 reflects these statistical measurements for the pac population, while table 4 on page 5 reflects statistical measurements for the ghost population. Additionally table 5 on page 6 shows the maximum raw fitness of the final pac generation for each run, and table 6 on page 7 shows the maximum raw fitness of the final ghost generation for each run. A statistical box plot for the best fitness in each generation of the pac population during each run can be found in figure 1 on page 8. A statistical box plot for the best fitness in each generation of the ghost population during each run can be found in figure 2 on page 9.

A file containing the controller tree for the Pac player in the run with the highest fitness in the final generation can be found at `./solutions/greenFinalPacTree.txt`. Sequentially, a file containing the controller tree for the Ghost player in the same run with the highest fitness in the final generation can be found at `./solutions/greenFinalGhostTree.txt`. These two best players were put against one another at the end of the experiment to play a final exhibition game. The world file for this game can be found at `./worldFiles/greenFinalBest.txt`.

Run	Average Fitness	Standard Deviation
1	86.13	21.14
2	107.4	39.92
3	67.27	24.32
4	122.07	25.45
5	93.27	35.51
6	133.33	43.82
7	74.53	26.04
8	76.53	23.57
9	95.27	24.75
10	124.8	38.61
11	86.07	30.09
12	87.73	25.89
13	137.6	34.98
14	72.53	22.04
15	103	20.36
16	82.2	14.35
17	83.93	27.26
18	92	32.79
19	62.6	23.26
20	135.07	41.34
21	70.07	18.32
22	77.8	26.71
23	60.33	14.37
24	51.73	17.91
25	111.4	30.11
26	100.07	40.81
27	86.67	30.22
28	103.87	29.07
29	104.6	40.4
30	68.27	19.56

Table 3: Co-Evolution (Green) Pac Statistical Measures

Run	Average Fitness	Standard Deviation
1	1055.2	258.81
2	848.67	210.37
3	1145.87	95.26
4	887.53	196.95
5	1061.2	184.32
6	688.13	331.53
7	1073.67	141.41
8	1040.73	138.34
9	879.27	171.78
10	1113.13	167.99
11	1321.87	59.29
12	1008.27	175.74
13	625.67	321.54
14	1191	95.5
15	962.27	200.55
16	1044.73	281.1
17	1078.87	150.44
18	1244.4	57.6
19	1237.07	120.71
20	895.8	251.34
21	908.4	139.31
22	1282.67	79.35
23	343.67	297.55
24	991.07	225.19
25	1041	153.98
26	810.53	318.34
27	939	198.79
28	935.93	218.6
29	832.27	210.39
30	1061.07	201.56

Table 4: Co-Evolution (Green) Ghost Statistical Measures

Run	Final Generation Max Fitness
1	100
2	103
3	83
4	129
5	120
6	171
7	120
8	76
9	86
10	163
11	123
12	100
13	144
14	96
15	106
16	96
17	86
18	86
19	43
20	168
21	86
22	93
23	63
24	46
25	113
26	103
27	66
28	93
29	143
30	63

Table 5: Co-Evolution (Green) Pac Final Generation Max Raw Fitness

Run	Final Generation Max Fitness
1	1335
2	839
3	1302
4	773
5	991
6	404
7	1143
8	999
9	933
10	1327
11	1343
12	846
13	802
14	1236
15	1069
16	1213
17	971
18	1272
19	1339
20	1099
21	786
22	1322
23	987
24	901
25	1071
26	1032
27	851
28	610
29	660
30	1169

Table 6: Co-Evolution (Green) Ghost Final Generation Max Raw Fitness

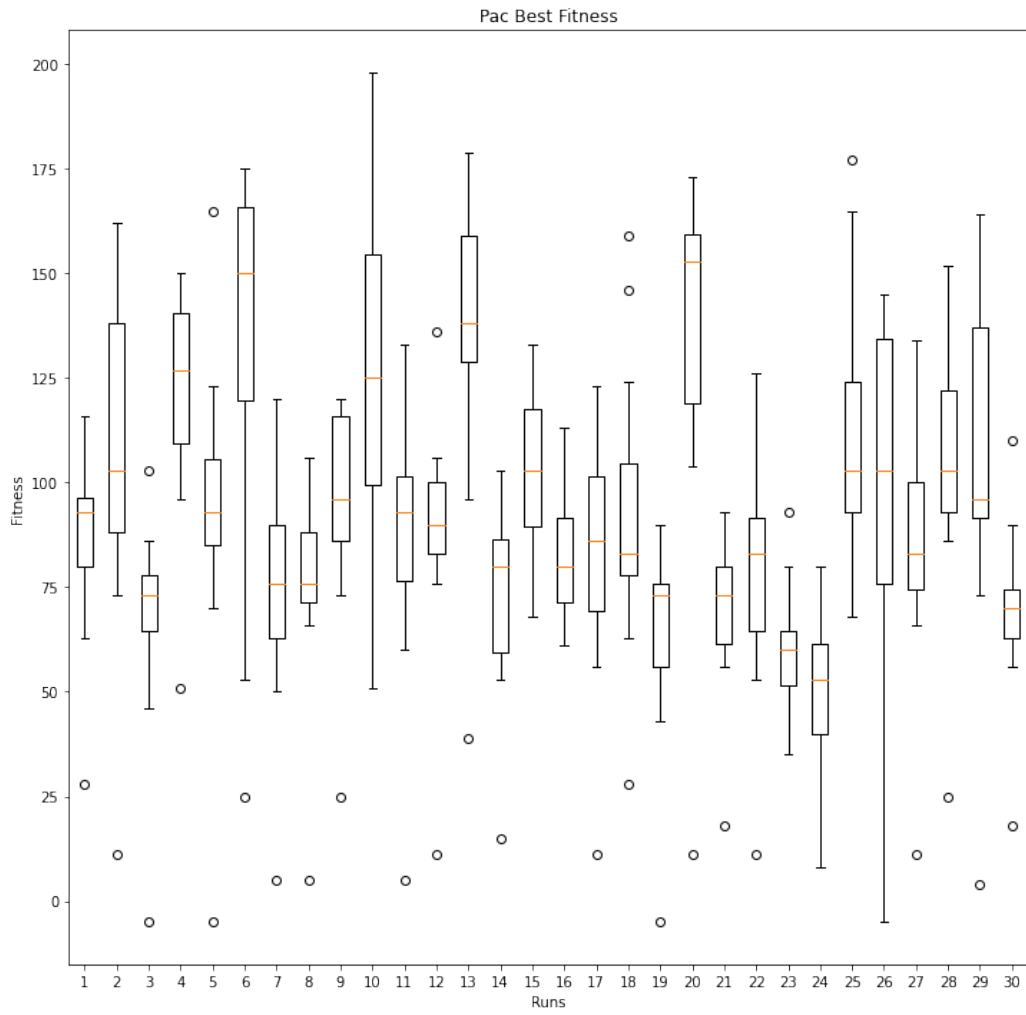


Figure 1: Green Experiment Best Fitness Per Generation Per Run for Pac Players

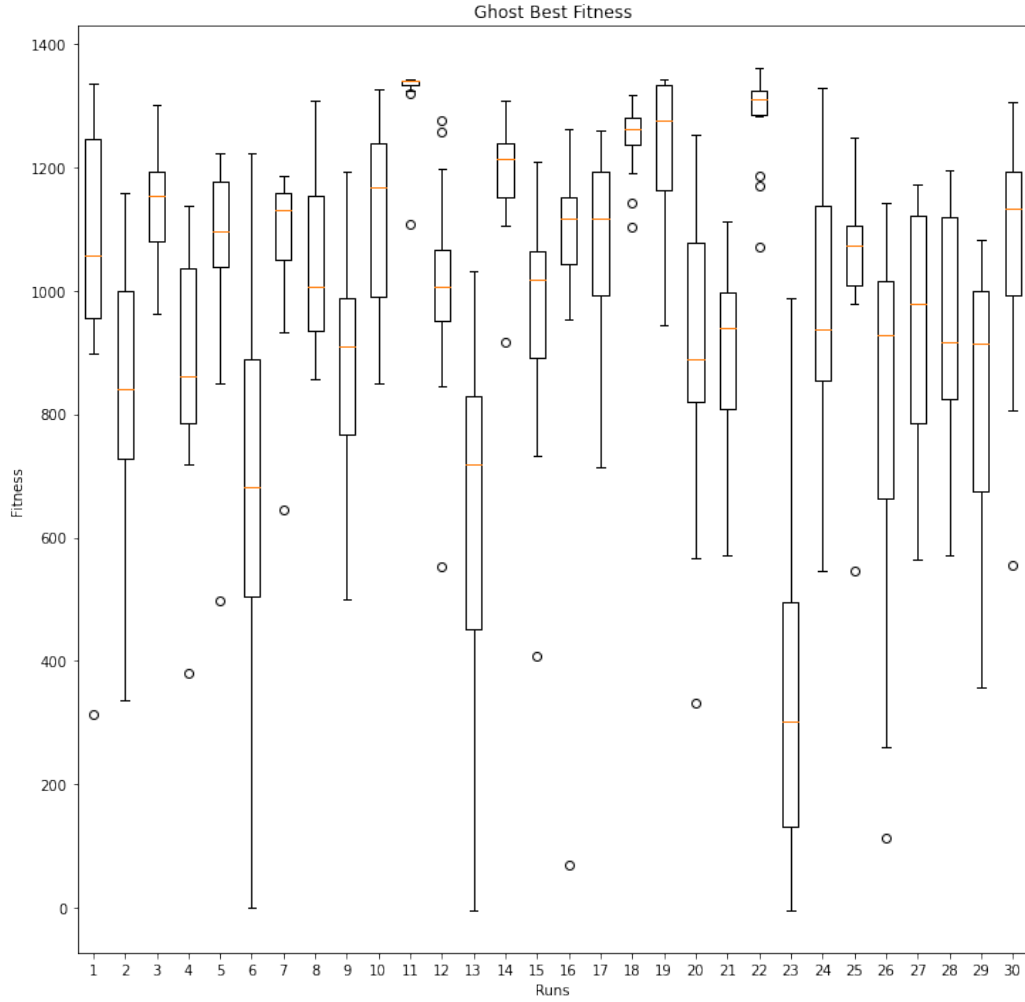


Figure 2: Green Experiment Best Fitness Per Generation Per Run for Ghost Players

1.4 Discussion

Although the Ghosts dominated this game against the Pac player, I can tell that the Pac player was reacting its movements to the movements of the Ghosts. In particular, towards the end right before one of the ghosts grabs the Pac player, they both do a sort of dance where it looks like the Pac player is trying to avoid the Ghost that is rapidly pursuing it.

1.5 Conclusion

I think that in this experiment, the Pac player was the under developed player in relation to the ghosts. The Pac player seems to make a number of seemingly random movements in the beginning. However, it is hard to say whether or not the Pac player was not just adjusting its position and recalculating it's strategies as the ghosts edged closer to it.

In observing the tree controller for the Pac player, the number of random moves could be due to the number of RAND terminal nodes in it's tree. I think that this could be further argued since the ghost controller tree is not as deep as the Pac player, but the ghosts move a bit more intelligently than the Pac player.

2 Cycling CIAO Plots (Yellow)

2.1 Methodology

The methodology for this experiment was the same as the green experiment, with the exception that was no exhibition game played at the end of the run.

2.2 Experimental Setup

Several configuration values were adjusted in an attempt to producing cycling CIAO plot. The mu value for both pac and ghost populations was lowered from green experiment values. Additionally, the depth limit was set at 3 with the reasoning that a lower depth limit would have a better chance of producing the same tree over and over again.

Admittedly, changing these values did not produce cycling. However, after reviewing lecture slides and seeing that the example plots were generated using random survival selection, survival selection for both pac and ghost population was changed to stochastic universal sampling from previous assignments. This produced cycling within my CIAO plots. The config file used for this experiment can be found at `./configs/yellow2c_config.txt`.

2.3 Results

As previously mentioned, cycling can occur when survival selection is happening randomly without reason. The CIAO plot for the pac population can

be seen in figure 3 on page 12. The CIAO plot for the ghost population can be seen in figure 4 on page 13.



Figure 3: Pac population CIAO Plot (Yellow)



Figure 4: Ghost population CIAO Plot (Yellow)

3 Multiple Identical Pac-Man Controllers vs Multiple Identical Ghost Controllers (Red 2)

3.1 Methodology

Same as green experiment.

3.2 Experimental Setup

Mostly the same as green experiment with the addition of the `num_pac` variable for increasing the number of pac players. Due to compute time increasing for the number of pac players, the max depth for trees was set to 5 instead of 10. Mu values and mutation rates were changed from green experiment to give some variety between experiments.

3.3 Results

After running the experiment, statistical data was collected for both the pac and ghost populations. Table 7 on page 15 reflects these statistical measurements for the pac population, while table 8 on page 16 reflects statistical measurements for the ghost population. Additionally table 9 on page 17 shows the maximum raw fitness of the final pac generation for each run, and table 10 on page 18 shows the maximum raw fitness of the final ghost generation for each run. A statistical box plot for the best fitness in each generation of the pac population during each run can be found in figure 5 on page 19. A statistical box plot for the best fitness in each generation of the ghost population during each run can be found in figure 6 on page 20.

A file containing the controller tree for the Pac player in the run with the highest fitness in the final generation can be found at `./solutions/red2FinalPacTree.txt`. Sequentially, a file containing the controller tree for the Ghost player in the same run with the highest fitness in the final generation can be found at `./solutions/red2FinalGhostTree.txt`. These two best players were put against one another at the end of the experiment to play a final exhibition game. The world file for this game can be found at `./worldFiles/red2FinalBest.txt`.

Run	Average Fitness	Standard Deviation
1	131.5	50.6
2	103.21	14.45
3	87.36	16.22
4	159.29	30.9
5	160.5	24.18
6	128.07	34.25
7	104.43	29.7
8	114.86	23.67
9	158.71	45.96
10	174.29	23.37
11	171.86	16.37
12	154.36	27.12
13	159.57	25.63
14	100.29	20.02
15	83.86	36.14
16	116.79	28.6
17	139.21	32.78
18	122.57	30.35
19	163.79	31.83
20	132.64	36.57
21	160.43	35.16
22	105.43	20.85
23	98.14	12.0
24	96.21	25.75
25	95.21	15.35
26	146.57	43.86
27	94.29	14.58
28	117	25.93
29	176.86	29.48
30	114.5	22.03

Table 7: Multiple Pac (Red 2) Pac Statistical Measures

Run	Average Fitness	Standard Deviation
1	815.5	301.24
2	555.93	348.26
3	829.86	265.25
4	854.57	166.18
5	1097.5	181.65
6	943.36	150.65
7	447.71	381.07
8	708.93	241.44
9	828.5	296.02
10	881.43	94.48
11	886.86	147.83
12	727.71	207.22
13	814.43	232.89
14	67.86	149.4
15	861.71	291.66
16	614.79	298.97
17	635.36	293.15
18	1141.36	108.91
19	893.64	71.69
20	610.86	231.43
21	782.57	266.42
22	749.07	251.33
23	628.57	320.29
24	919.07	264.12
25	104.57	185.4
26	628.86	376.5
27	411.36	339.11
28	796.64	310.32
29	1024.36	170.25
30	1083.93	165.81

Table 8: Multiple Pac (Red 2) Ghost Statistical Measures

Run	Final Generation Max Fitness
1	199
2	103
3	106
4	180
5	162
6	161
7	113
8	123
9	192
10	178
11	177
12	179
13	179
14	115
15	126
16	140
17	185
18	146
19	182
20	162
21	200
22	126
23	104
24	106
25	106
26	192
27	113
28	146
29	186
30	127

Table 9: Multiple Pac (Red 2) Pac Final Generation Max Raw Fitness

Run	Final Generation Max Fitness
1	909
2	544
3	1082
4	968
5	1333
6	961
7	49
8	939
9	956
10	916
11	930
12	865
13	795
14	95
15	1072
16	887
17	1013
18	1123
19	1024
20	660
21	930
22	719
23	933
24	1013
25	0
26	946
27	0
28	805
29	1039
30	1121

Table 10: Multiple Pac (Red 2) Ghost Final Generation Max Raw Fitness

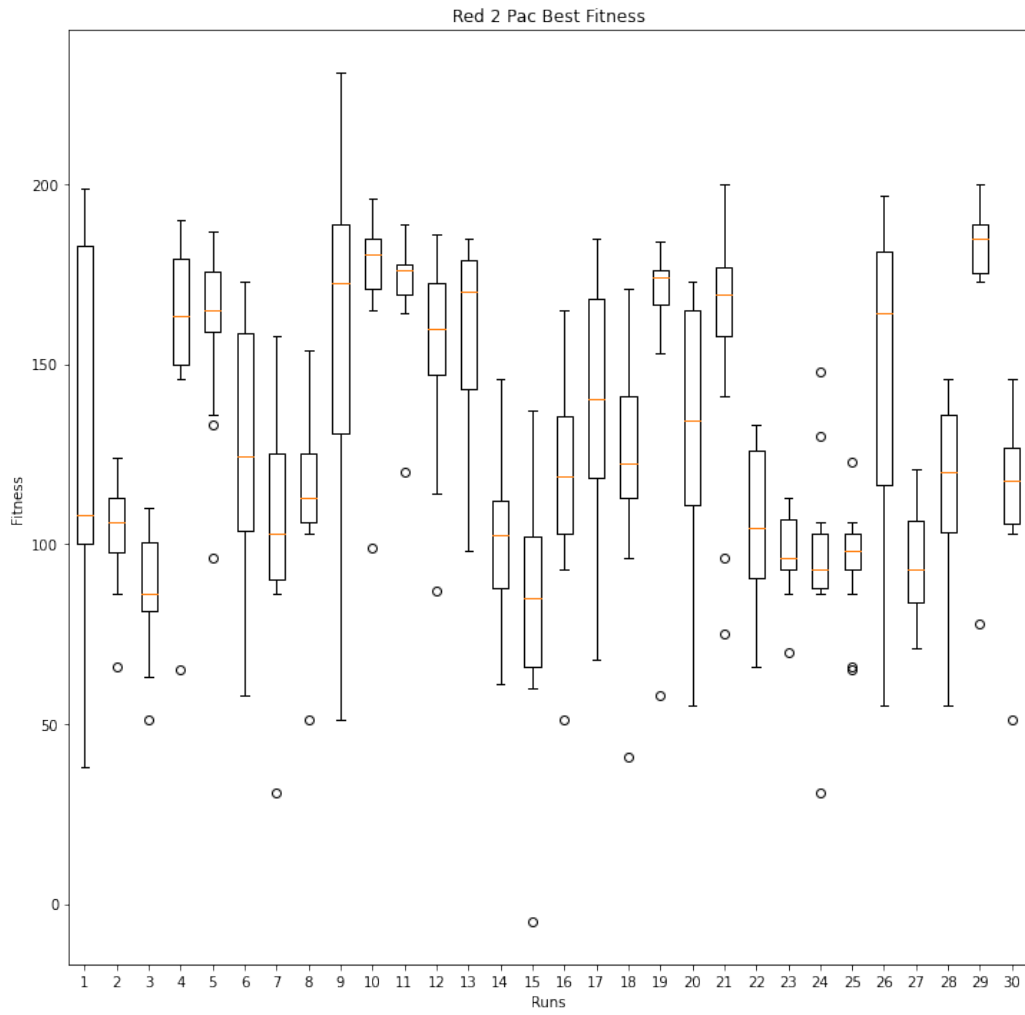


Figure 5: Red 2 Experiment Best Fitness Per Generation Per Run for Pac Players

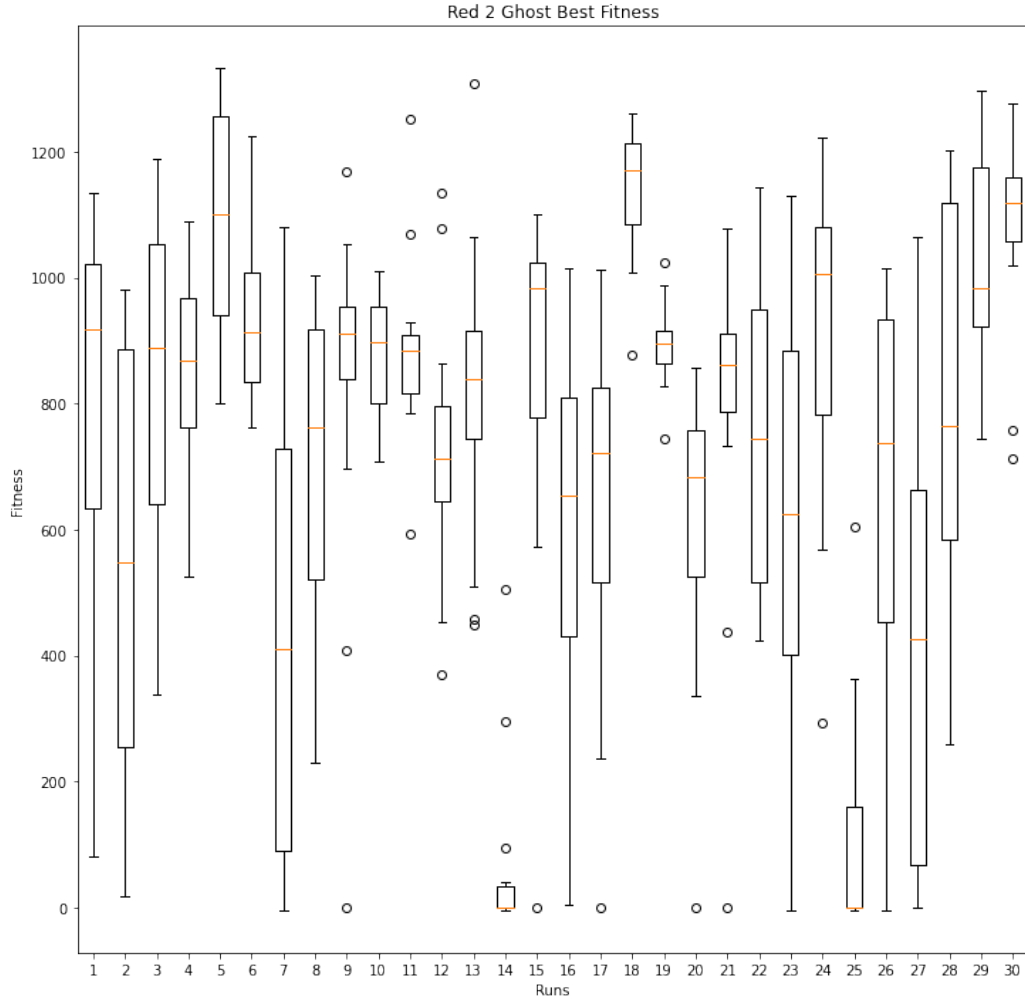


Figure 6: Red 2 Experiment Best Fitness Per Generation Per Run for Ghost Players

3.4 Discussion

The thing that I found most interesting about this experiment with multiple pac players is that once the ghost kills two of the four Pac players, it gets stuck in an infinite loop to going back and forth between the two Pac corpses. This allows the other two surviving Pac players to freely move about the board and ultimately get a high score. This could also be due to

the only terminal node (besides constant values) is the node for calculating the nearest Pac player.

3.5 Conclusion

In order to counteract the infinite loop that the ghost controller got into during the final game, future experiments could involve supplying the ghost with a terminal node deciding if the nearest Pac player is alive or dead.