This tracks convergences on random (pickled) graphs

==== RESTART: /Users/valeriemcculloch/Desktop/research/evolGamePickle.py ====

Gen. 0 (0.00%): Max/Min/Avg Fitness(Raw) [4.90(9.00)/3.57(1.00)/4.08(4.08)]

Gen. 10 (66.67%): Max/Min/Avg Fitness(Raw) [5.14(9.00)/3.50(0.00)/4.28(4.28)]

Gen. 15 (100.00%): Max/Min/Avg Fitness(Raw) [5.35(9.00)/3.98(2.00)/4.46(4.46)]

Total time elapsed: 24.882 seconds.

[['4', '4', '1', '0', '1'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4'], ['2', '3', '0', '4', '4']]

>>>

==== RESTART: /Users/valeriemcculloch/Desktop/research/evolGamePickle.py ====

Gen. 0 (0.00%): Max/Min/Avg Fitness(Raw) [4.98(9.00)/3.61(1.00)/4.15(4.15)]

Gen. 10 (66.67%): Max/Min/Avg Fitness(Raw) [4.79(9.00)/3.51(1.00)/3.99(3.99)]

Gen. 15 (100.00%): Max/Min/Avg Fitness(Raw) [4.90(9.00)/3.57(1.00)/4.08(4.08)]

Total time elapsed: 28.359 seconds.

[['1', '0', '3', '4', '0'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1'], ['1', '2', '1', '1', '1']]

>>>

==== RESTART: /Users/valeriemcculloch/Desktop/research/evolGamePickle.py ====

Gen. 0 (0.00%): Max/Min/Avg Fitness(Raw) [486.55(477.00)/360.73(366.00)/405.46(405.46)]

Gen. 10 (66.67%): Max/Min/Avg Fitness(Raw) [488.96(477.00)/368.24(374.00)/407.47(407.47)]

Gen. 15 (100.00%): Max/Min/Avg Fitness(Raw) [487.02(477.00)/371.80(376.00)/405.85(405.85)]

Total time elapsed: 249.690 seconds.

[['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3'], ['2', '3', '0', '2', '3']]

Above are three outputs of a random graph fitness n=15.

Notice the first two runs don’t produce very good chromosome at all, so I increased the loops in the fitness function so as the scoring would mean more

This chromosome looks better (from what I know about the space on certain fixed graphs) but it also doesn’t change at all through the iterations.

So I ran it again:

Gen. 0 (0.00%): Max/Min/Avg Fitness(Raw) [615.44(599.00)/451.10(461.00)/512.87(512.87)]

Gen. 10 (66.67%): Max/Min/Avg Fitness(Raw) [619.37(599.00)/462.40(473.00)/516.14(516.14)]

Gen. 15 (100.00%): Max/Min/Avg Fitness(Raw) [618.40(599.00)/481.66(488.00)/515.33(515.33)]

Total time elapsed: 246.672 seconds.

[['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4'], ['0', '2', '2', '0', '4']]

It seems that the genetic algorithm has trouble beating the first best fitness. However, maybe for a reason: this is also a good fitness lots of builds and adv-builds can be fruitful.