

COMP SCI 5401 FS2017 Assignment 1c

Stuart Miller
sm67c@mst.edu

October 7, 2017

1 Overview

Assignment 1c presented a great many parameters to test. As more techniques are introduced, it becomes more and more difficult to distinguish a genuinely "good" parameter value as one can never be entirely sure if it was solely one value that resulted in a change in fitness landscape, a combination of multiple values, or perhaps the order of calls to the random number generator came out more favorably. Therefore, in this assignment, it has been attempted to isolate parameters being tested and pick favorable and reasonable values to the parameters not immediately relevant.

As a precursor, the entirety of this assignment will make use of n-point crossovers and random reset mutations. Although other operators are implemented in the codebase, these two have been set to be self-adaptive in Bonus 2. For consistency with the runs in the Bonus section, even when self-adaptability is turned off, the analysis will still use only these two operators. For initial values and non adaptive states, the operators will be fixed at three crossover points and a 12% mutation chance. These are values that were determined to be beneficial in Assignment 1b. Furthermore, survival strategy will always be set to comma (μ, λ) and selection will be k-tournaments with moderate selective pressure.

2 Penalty Coefficient

To test the penalty coefficient, two control sets were run where invalid placements were solved by random replacements and then a repair. The penalty function was then implemented, with static weights of 1.0, 2.0, and 5.0, respectively.

As the graph in Figures 1, 2, and 3 all show, the penalty weight functions shows significantly lower fitness values than other methods. This is to be expected for intermediate solutions, but the best solution shouldn't have any penalty. Additionally, the fact that the different weights returned the same max fitness shows that this may not be an optimal strategy for this problem. It is worth noting that numerous evolutionary strategy were tried and this combination yielded the best and most consistent results.

In all three data sets, each penalty weight averaged to be more or less the same final value. This similarity between data sets is likely due to the fact that in all instances, invalid solutions are VERY easy to come by (and can consequently be resolved small alterations from the repair function). Regardless of the complexity of the shapes, penalizable values are just as likely.

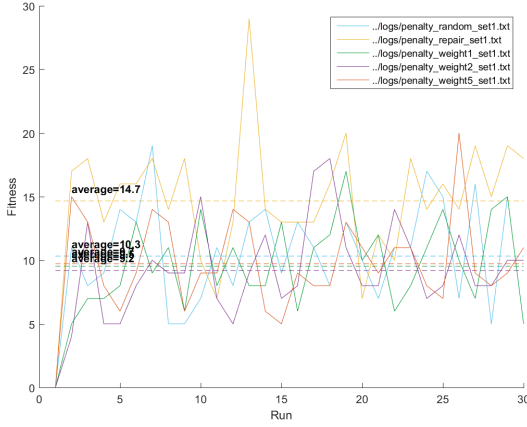


Figure 1: Penalty Weights, 50Shapes.txt

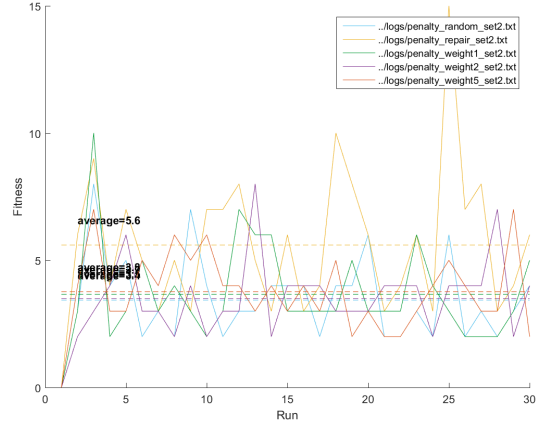


Figure 2: Penalty Weights, 100Shapes.txt

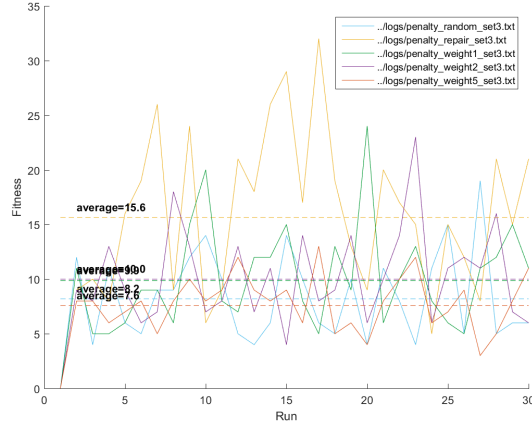


Figure 3: Penalty Weights, 100ShapesComplex.txt

3 Self-Adaptive Parameter

To test self-adaptability, the number of crossover points n was set to be adaptable. For offspring created, there was an equivalent random chance of inheriting each parent's n . Additionally, n had a small chance to mutate (equivalent to the genetic mutation rate) by ± 1 . For analysis, the mutation rate was averaged across all member of the population for each generation.

As it was determined in Section 2, the penalty function yields no great benefit for this problem, so the repair function method was used here instead. The other parameters were kept the same as in section 2.

As figures 4, 5, and 6 all show, allowing crossovers to self-adapt has a significant improvement on the average best fitness. It is also interesting to worth noting that while each instance started at three crossover points, some runs took this number higher, and some drove it lower. Examining the log (located in the ./logs/ directory) shows no consistent trend. It can only be assumed that each run's random initialization took it to a different area of the fitness landscape where more or less crossovers were desirable.

There is no distinct difference between the three data sets; self-adaptivity always proved

beneficial. In fact, the difference between the two was nearly the same each time. This is likely due to the fact that each instance of this problem has such a variable fitness landscape, that allowing the recombination to vary itself more (by increasing the number of crossovers) allowed each run to exploit its particular fitness landscape.

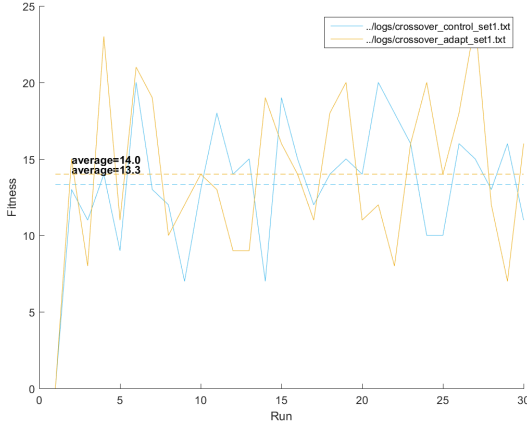


Figure 4: Adaptive Crossovers, 50Shapes.txt

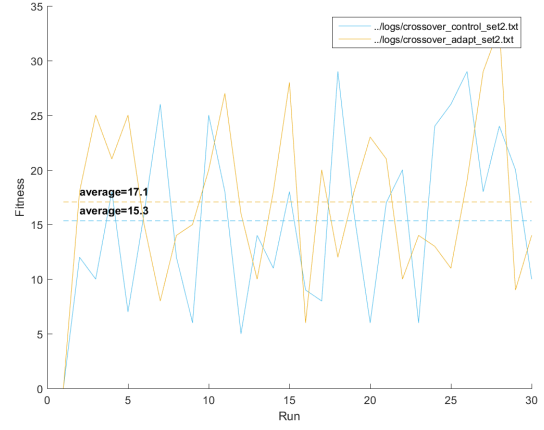


Figure 5: Adaptive Crossovers, 100Shapes.txt

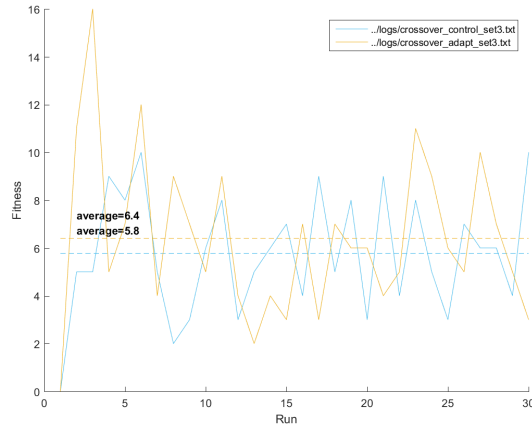


Figure 6: Adaptive Crossovers, 100ShapesComplex.txt

4 Box Plots and Statistical Analysis

Box plots were generated for each test run graphed above. However, since this totals to a rather large number of boxplots, they have not all been pictured in this report. A full listing of box plots can be viewed in `./doc/images/assn1c_boxplot_*.png`.

In Figures 7 and 8 box plots are shown for the first data set. It can be observed that the resulting populations are as to be expected, with an upper bound having the best fitness and the remaining population distributed below. It is worth noting that with the self-adaptive crossovers, the distribution tends to skew more towards the maximum. This is likely due to the

fact that, as shown in Section 3, self-adaptivity pushes the population closer to a better fitness solution.

A statistical analysis was also run (Figures 9 and 10) which further illustrates this finding.

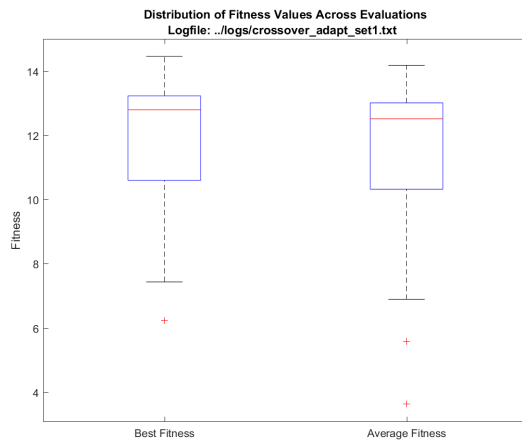


Figure 7: Adaptive Crossovers, 50Shapes.txt

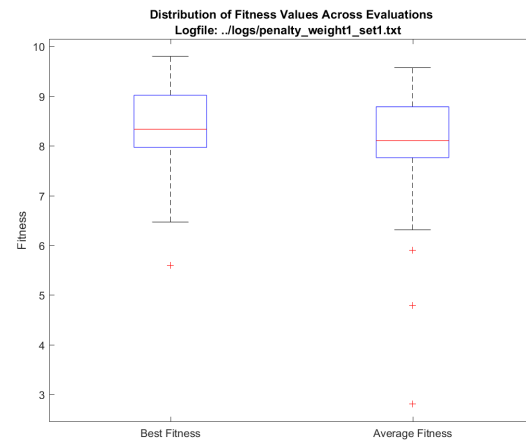


Figure 8: Penalty Weight 1.0, 50Shapes.txt

F-Test Two-Sample for Variances		
	Variable 1	Variable 2
Mean	13.84962406	15.08270677
Variance	18.53782183	46.39462292
Observations	133	133
df	132	132
F	0.399568326	
P(F<=f) one-tail	1.15805E-07	
F Critical one-tail	0.750236915	
t-Test: Two-Sample Assuming Unequal Variances		
	Variable 1	Variable 2
Mean	13.84962406	15.08270677
Variance	18.53782183	46.39462292
Observations	133	133
Hypothesized Mean Difference	0	
df	223	
t Stat	-1.76476609	
P(T<=t) one-tail	0.039485976	
t Critical one-tail	1.65171532	
P(T<=t) two-tail	0.078971953	
t Critical two-tail	1.970658961	

Figure 9: Adaptive Crossovers, 50Shapes.txt

F-Test Two-Sample for Variances		
	Variable 1	Variable 2
Mean	24.21052632	15.08270677
Variance	55.1523126	46.39462292
Observations	133	133
df	132	132
F	1.188765187	
P(F<=f) one-tail	0.160885075	
F Critical one-tail	1.332912284	
t-Test: Two-Sample Assuming Equal Variances		
	Variable 1	Variable 2
Mean	24.21052632	15.08270677
Variance	55.1523126	46.39462292
Observations	133	133
Pooled Variance	50.77346776	
Hypothesized Mean Difference	0	
df	264	
t Stat	10.44622691	
P(T<=t) one-tail	6.60617E-22	
t Critical one-tail	1.65064591	
P(T<=t) two-tail	1.32123E-21	
t Critical two-tail	1.968990497	

Figure 10: Penalty Weight 1.0, 50Shapes.txt

5 Bonus 1

6 Bonus 2