CS5401 Assignment 1c with bonus

Tanner Wendland

I. Main

The bonus that was implemented was a repair function that removed bulbs that shine on each other. There are a total of 4 plots, two for a randomly generated puzzle, and two for the provided puzzle.

II. Findings

Figure 1 corresponds to the best run through all thirty runs of the experiment in which we attempt to solved the puzzle provided and did not use the repair function. Figure 2 corresponds to the best run through all thirty runs of the experiment in which we attempt to solve the puzzle provided and did use the repair function. Figure 3 corresponds to the best run through all thirty runs of the experiment in which we attempt to solve a puzzle that was randomly generated and did not use the repair function. Figure 4 corresponds to the best run through all thirty runs of the experiment in which we attempt to solve a puzzle that was randomly generated and did use the repair function.

From the results, it can be seen that the experiments that utilized the repair function have a much higher average fitness score. It is also obvious that the best fitness scores for the randomly generated puzzle are also much higher on average. The author will go into detail on these averages in the statistical analysis section.

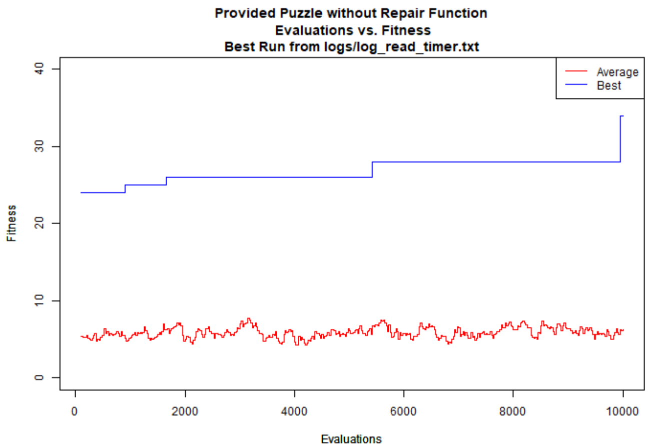

Fig. 1.

III. Statistical Analysis

We shall use a test statistic to compare performance, in particular the experiments that used a randomly generated puzzle shall be compared and the experiment that used the provided puzzles shall be compared. We shall use the f-test to determine if the variances can be assumed to be the same. We use the f-test feature in R to run f-test on the average fitness and the best fitness for the particular experiments, and the results are in Table I. We can see by Table I that for each comparison, the f-test concludes that the variances are not equal. Thus a two-tailed two-sample t test assuming unequal variances will be performed.

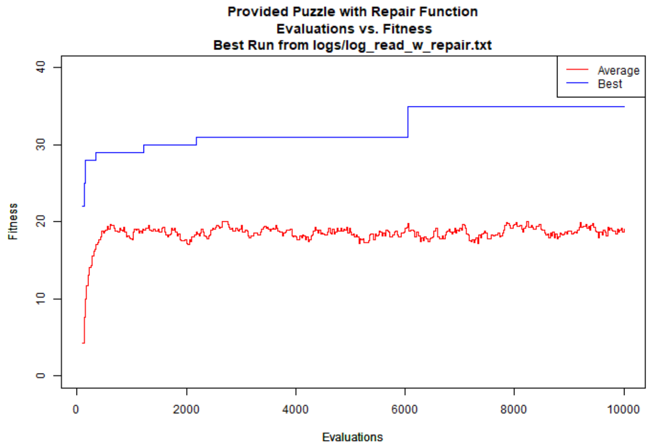

Fig. 2.

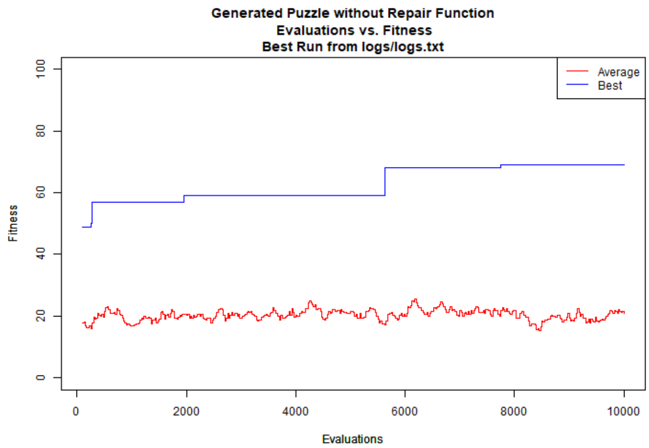

Fig. 3.

Our null hypothesis is: true different in mean is equal to zero. Our alternative hypothesis is: true difference in mean is not equal to zero.

Table II is the results from all four t-test performed, and for each t-test preformed the p-value is low enough where we can reject the null hypothesis. Thus the statistically better algorithm is the algorithm that has the largest mean fitness.

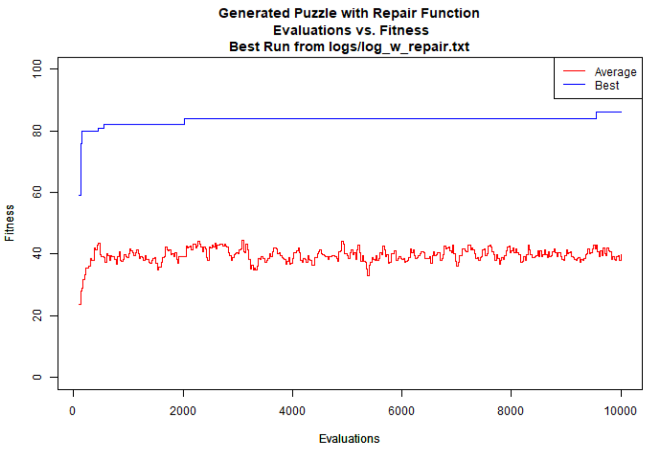

Fig. 4.

Table III provides the means and standard deviations for the average fitness score and the best fitness score for each experiment. From this table we can see that the experiments that used the repair function have higher means than their counterparts that do not use the repair function. As since the results from our t-test says that the experiments that have higher means is the statistically better algorithm, we can conclude that the experiments that use the repair function are better.

IV. Questions

- How much improvement does a constraint-satisfaction EA employing a penalty function obtain vs. an EA employing your additional constraint satisfaction technique vs. a plain-vanilla EA?
  - The improvement between an EA with a penalty function and a plain-vanilla EA is quite large. Before the penalty function was implemented, if the enforce wall constrain was enable it more often than not would never produce a solution with a fitness value greater than 0. The improvement from the EA with a penalty function to an EA with a penalty function and a repair function is also quite noticeable. For the experiments that were generated it is clear from our statistical analysis that the repair function experiments performed better than the EA that just used a penalty function. With the repair function the generated experiment produced much better results. The experiment with the provided puzzle also had an increase in performance, and in fact it actually produced the global optimum.
- How significant is the impact of Validity Forced plus Uniform Random initialization versus plain Uniform Random initialization for a plain-vanilla EA, a constraint-satisfaction EA employing a penalty function, and a constraint satisfaction technique employing your additional technique? Explain your findings!
  - The impact of the forced validity plus uniform random can vary depending on the EA being used. For the plain-vanilla EA it is not noticeable since invalid solutions are still created quite frequently for walls that do not have a singular unique way of placing bulbs around them. However for the EA that uses a penalty function it is more noticeable since the initial fitness values will be slightly higher since less walls will be penalized. For the EA with a repair function, it is similar to the results found with the EA that uses the penalty function.
- How is the penalty function coefficient correlated with the solution quality of your constraint- satisfaction EA employing a penalty function? Explain your findings!
  - If the penalty coefficient was to small, the solution quality would be low because it would allow invalid solutions exist their fitness values would be very similar to other solutions that have less constraint violations. If the penalty coefficient was too high, then invalid solutions were being killed off to quickly and one could lose genetic diversity very quickly. Manually tuning the coefficient allows to EA to keep these invalid solutions but still prefers solutions that do not violate constraints.

	F	Numerator df	Denominator df	p-value	95% Confidence Interval	Ratio of Variances
Generated Average Fitness	9.948	396	396	$<2.2*10^{-16}$	(8.167, 12.112)	9.947976
Generated Best Fitness	0.57999	396	396	$7.293*10^{-8}$	(0.4762, 0.7065)	0.5799894
Provided Average Fitness	0.37217	396	396	$<2.2*10^{-16}$	(0.3055, 0.4533)	0.3721734
Provided Average Fitness	0.27307	396	396	$<2.2*10^{-16}$	(0.2242, 0.3326)	0.2730656

TABLE I

	t	df	p-value	95% Confidence Interval
Generated Best Fitness	-73.584	474.82	$<2.2*10^{-16}$	(-21.413, -20.299)
Generated Average Fitness	-135.71	739.73	$<2.2*10^{-16}$	(-19.528, -18.971)
Provided Best Fitness	-38.719	654.9	$<2.2*10^{-16}$	(-5.775, -5.217)
Provided Average Fitness	-170.64	597.26	$<2.2*10^{-16}$	(-12.762, -12.472)

TABLE II

	Mean	Standard Deviation
Generated Best Fitness	62.7	5.4
Generated with Repair Best Fitness	83.6	1.7
Generated Average Fitness	20.3	1.7
Generated with Repair Average Fitness	39.6	2.2
Provided Best Fitness	26.7	1.5
Provided with Repair Best Fitness	32.2	2.4
Provided Average Fitness	5.8	0.7
Provided with Repair Average Fitness	18.5	1.3

TABLE III