Daniel Glauber

Project 2 Results

Table of results from experiments.

Crossover Operator	prob Apply Crossover	String Size	Min N	Max N	Threshold	Execution Time (sec)
Uniform	0.2	24	75	80	0.06667	0.98675
One-point	0.6	24	65	70	0.07692	0.57457
Two-point	0.6	24	120	130	0.08333	0.95921
One-point	0.6	48	320	340	0.06250	10.89071
Two-point	0.6	48	150	160	0.06667	3.84175
One-point	0.6	100	1440	1520	0.05556	109.20490
Two-point	0.6	100	1280	1360	0.06250	87.42322

I found that uniform crossover was only able to complete bisection on a string size of 24, but that was only possible by setting probApplyCrossover to 0.2 instead of 0.6. Both one-point and two-point crossover had no problem completing bisection for string sizes 24, 48, and 100. My results show that one-point crossover was better than two-point crossover when running bisection on a string size of 24. One-point crossover was able to finish faster and had a max n that was almost half the size of the max n for two-point crossover. Two-point crossover was the clear winner for string sizes of 48 and 100. For a string size of 48 two-point crossover had a max n that was over half the size of the max n for one-point crossover. For a string size 100 of two-point crossover had a max n that was 90% the size of the max n for one-point crossover. Based on my experimentation I can conclude that uniform crossover is so bad when using trap-4, it is not a viable option. Both one-point and two-point crossover have a range that of string sizes where each of them is better than the other. One-point crossover is better on shorter strings and two-point crossover is better on longer strings. In summary as the string size increases crossing over at more points is beneficial, but if the string size is not yet large enough there is potential it can hurt the performance.