Sam Trenter
Project 2
Task 4

What operators work well and what don't for trap4 and onemax?

trap4: for trap4 it seems that both the "point" crossover operators work well, which makes sense. In trap4 we wan't to avoid breaking up good segments and replacing bad segments. With uniform crossover, every time we crossover, we have a large chance to break up a good segment and replace it with a similarly high value segment with zeros. What the benefit is with either crossover is that entire segments can be lifted from one individual to another.

Onemax: For onemax we actually have a different issue, in that we can look at each bit individually. With the one/two point crossovers, we may replace our segment with a similarly good segment but that one will still have zeros. Uniform Crossover gives us the opportunity to just replace a 0 with a 1. While we will also sometimes replace 1 with 0 if our population slowly gets better that will happen less and thus our population will slowly get better.

Testing different variations:

Stringsize	Fitness	Generation limit	Crossover	Result (Lowest population model should work)
24	onemax	100	Crossover	3
24	onemax	100	1 point	3
24	onemax	100	2 point	3
24	Trap4	100	Crossover	Not found
24	Trap4	100	1 point	273
24	Trap4	100	2 point	325
48	onemax	200	Crossover	4
48	onemax	200	1 point	3
48	onemax	200	2 point	4
48	Trap4	200	Crossover	Not found
48	Trap4	200	1 point	825
48	Trap4	200	2 point	1091
100	onemax	300	Crossover	10
100	onemax	300	1 point	11
100	onemax	300	2 point	7

100	Trap4	300	Crossover	Not Found
100	Trap4	300	1 point	2320
100	Trap4	300	2 point	1485

Not found: unsurprisingly, crossover is terrible for trap4 and failed to converge for all of our tests. We would need to be extremely lucky to see it work.

What parameters should you vary?

If Doing a trap problem you should avoid uniform crossover. One and Two point seem to be similar but on the last run two point was much better compared to one point. This is probably not conclusive because of a small sample size.I theorize that as we add more points to crossover the effectiveness of it declines, due to it converging onto uniform crossover.

Even with “Many point crossover” converging to a uniform crossover like recombination, there could be a sweet spot for a certain string size. It looks like we might start to see this with the stringsize=100 problems.

Conclusion:

I think this project shows that A lot can be done to mess with the effectiveness of a GA but even more can be done to increase their effectiveness. Especially if we know about the things that are causing the problems. Even when a problem is specifically designed to give GAs issues can be worked to have minimal effect on our overall performance. I also have never implemented a bisection process and I thought it was a fun challenge.