Colin Conn

Homework 3

Evolutionary Computing

3/21/23

1. For the SGA I plan to tune crossover rate and mutation rate. I think that tuning the crossover rate for the SGA is interesting because it is not a factor in the ES, which could let me see which of the algorithms is more effective. I expect for a higher crossover rate to cause the algorithm to converge faster. This could potentially lead to being stuck in a local minimum. As for the mutation rate, I think that it would be an interesting parameter to tune because it accomplishes the opposite of the crossover rate. A higher mutation rate will help the algorithm avoid getting stuck in a local minimum. This will likely cause the algorithm to take longer to converge, but it will also be more likely to converge to the global minimum.

   For the ES I plan to tune the mutation rate and weights of recombination. I find tuning the mutation rate to be interesting because it will allow me to compare the parameter's impact between SGA and ES. Because my ES performs mutation before survivor selection rather than after like my SGA does, I expect a higher mutation rate to be more beneficial. If a genome is mutated negatively it will be much less likely to make it through selection with that negative mutation. As for recombination weights, I plan to have a weight value for each parent. I find this parameter interesting because as I am currently doing recombination the split is always 50/50. Parents are chosen randomly so I don't expect the weights to always make fitness higher, but I do think that it will cause the population to converge slower.

2. The utility measure I plan to use is average number of evaluations to a solution. I like this computational measure because it uses average fitness instead of best fitness. I am testing mutation rate for both algorithms and I want to use the average in case one member is mutated in a way that makes it a big outlier in the population. If I used MBF rather than AES, I could be judging on an outlier best member of the population rather than the population as a whole. AES can be used to compare the speed of different EAs. This is interesting because my different EAs have very different selection methods, and I would like to see how their time to reach a minimum average fitness is affected by their selection.

3.

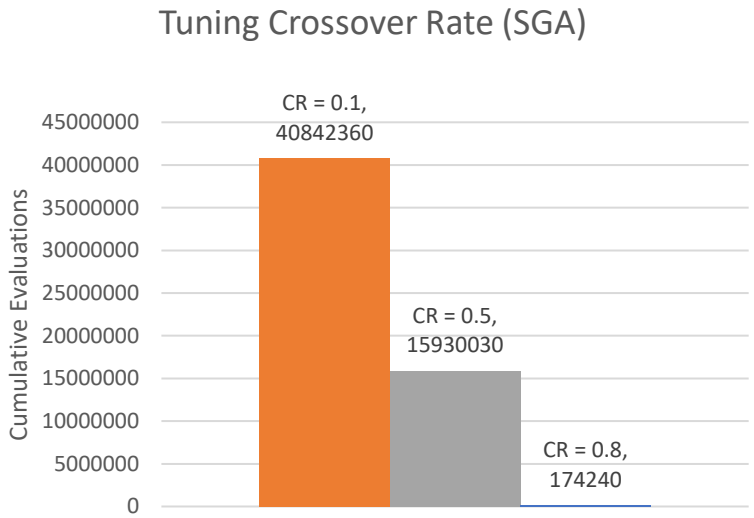
a. For each parameter in each EA I will test three different variations of the parameters as follows:

   SGA crossover rate: 0.1, 0.5, 0.8
   SGA mutation rate: 0.01, 0.05, 0.1
   ES mutation rate: 0.01, 0.05, 0.1
   ES recombination weights (parent 1/parent 2): 20/80, 50/50, 80/20

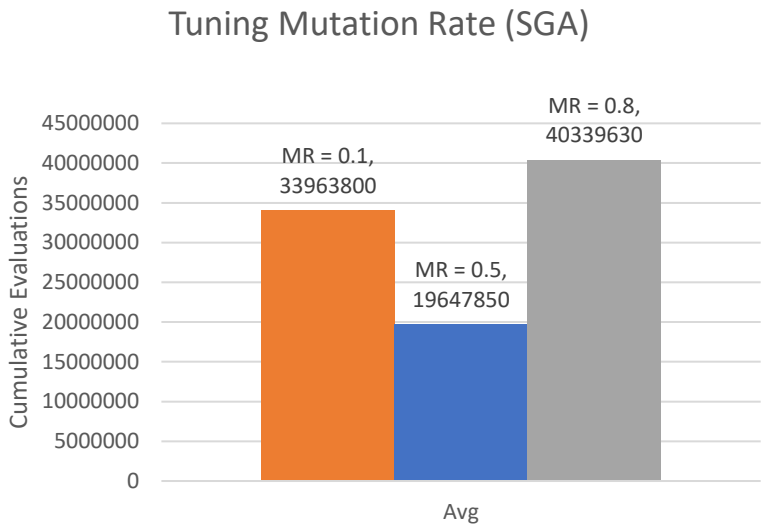
   I will run each EA 10 times for each parameter change with an average fitness threshold of 300 (which goes from around 0 to 1000). Additionally, for the sake of time I will limit the number of generation to a maximum of 1000000. The average fitness of 300 is attainable for both EAs, however my ES has a tendency to get stuck sometimes.

   The output for each run will be a text file that reports every generations statistics (as was the output for previous assignments). Once the average fitness reaches the threshold the program will terminate. This way I can just get the last generation and number of evaluations from each output file.

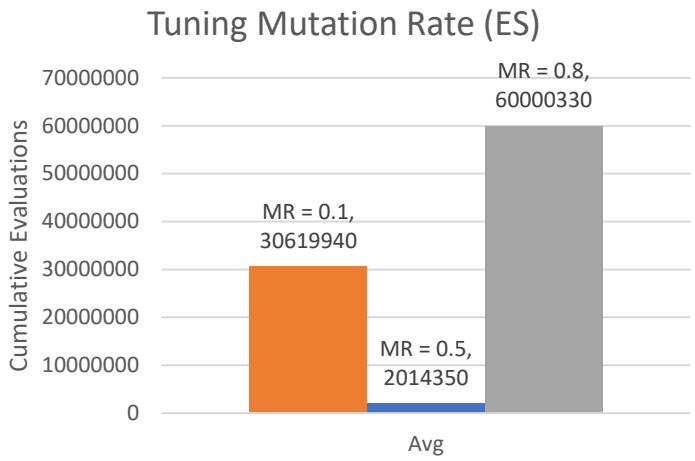
b. For the SGA, increasing the crossover rate of the algorithm lead to drastically fewer evaluations to reach the threshold. With an average evaluations of 174240, the crossover rate of 0.8 was over 90 times faster than a rate of 0.5 and over 200 times faster than 0.1 over the course of the 10 trials. Therefore, out of these options I would prefer to use 0.8 as the crossover rate for my SGA. There is probably some finer tuning that I can do to find the most efficient crossover rate, but that is for another time.



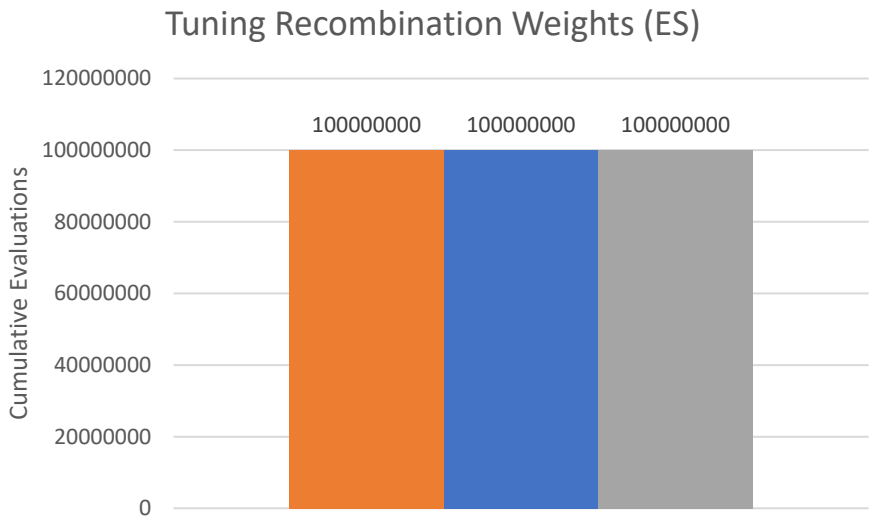
As for mutation rate, both mutation rates of 0.1 and 0.8 (33963800 and 40339630 average evaluations, respectively) were much slower than 0.5 (19647850). Despite being about twice as fast, the mutation rate of 0.5 achieved nowhere near the efficiency of most of the crossover rates tested earlier. I don't think that adjusting the mutation rate from a low value to a high value is something I would continue to do with my SGA because it leads to a lot of variability that I don't necessarily want in my SGA.



I think there is a problem with my ES that I spent too long trying to figure out that causes it to either get really good really fast, or it gets stuck and can't recover, so I'm not sure that AES is the best measure of utility for this algorithm. Because of this issue I did find a benefit to tuning the mutation rate higher. In situations where the ES gets stuck, a much higher mutation rate was able to sometimes get it unstuck. However, I still think that mutating every other genome is unnecessary and causes more problems than it solves. In several cases the algorithm got worse because it kept mutating in a detrimental way (with a mutation rate of 0.8 the algorithm more than once got stuck with a fitness of around 10^-15).



The recombination weights are where stuff really started getting interesting. For some reason when using any weights on the parents for recombination, the algorithm converged at exactly 250 every time. I can't for the life of me think of why that would be happening but as 250 is less than the threshold of 300, it looks very very slow (which I guess it technically is, but I don't think it's fair to just say its slow).



c. Some of my experiments matched my expectations, namely for crossover rate for SGA and mutation rate for ES. I was correct for both of these predictions, but for mutation I would consider it half-correct. I predicted that a high mutation rate would be more beneficial. This was correct to a point, but there was a point where the mutation rate was very much hurting the algorithm. For the mutation rate of the SGA I expected it to take longer to reach the threshold with a higher mutation rate, and

similarly to the ES having a mutation rate around 0.5 made the speed of the algorithm increase but it was much less effective for the SGA. Finally, we have the recombination weights, there was no way I could have predicted what happened. I think if I took some time to really examine the recombination I might be able to find out why it keeps converging at the same fitness but for now I'm lost on that.

If I were to do this again I would dig deeper into the recombination weights with a more robust set of tests. This convergence was by far the most interesting component of this parameter tuning and I would love to find out why that happened. I would repeat the test with many other distributions of weights between the parents. Additionally, I think I used ranges that were too high for mutation rate tuning, so if I were to do the assignment again I would choose some smaller numbers to use in addition to the bigger ones to compare the effects.