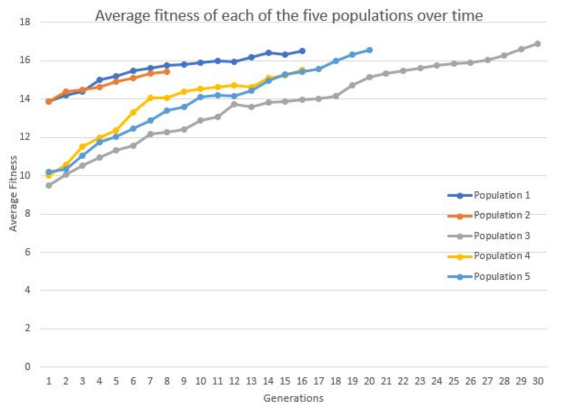
Part 1

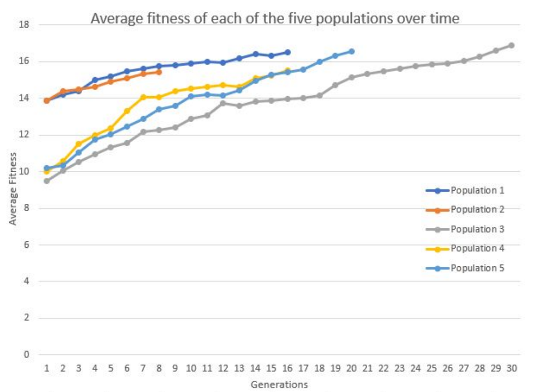
1. Avg = 27.08
   Min = 7
   Max = 50(caps out here due to condition in writeup)

2. 

   I notice that there is a linear increase in average fitness over time in each of these runs regardless of how many generations it requires to achieve its string of all ones. I also noticed that the average fitness caps out at roughly 17 even if there are a greater number of generations. I believe that there is a correlation between generation number and a higher average fitness however the fact that the average fitness caps out at 17 means that our average fitness will only grow more slowly since there are more generations than before. Our initial average fitness(generation 0) generally lies somewhere between 9.5 and 10.5. The differences in some of the changes in slope of the graph are likely caused by crossover and mutation functions working concurrently to push individuals in the population to be more fit.
3. With crossover being turned off we have an average of 50, a min of 50 and a max of 50. Essentially, what this is saying is that at least within the first 50 generations we cannot find the string of all ones or the most fit individual possible(fitness value 20). When we look at our runs we see that the average fitness caps out at about 16 and the best fitness hovers around 14-16 not varying by very much. This is because we do not have any crossover happening. When we perform a single point crossover of two individuals in a generation we introduce a variant that combines the genomes of the two fittest individuals in different order (p1(head)p2(tail)) or(p2(head)p1(tail)). This allows us to have a greater chance of having an even fitter individual than our previous generation. We initially had a crossover rate of 0.7 meaning that the probability of us performing a cross over on the tow fittest individuals in a population for that generation is 70%. Whether this gives us a fitter individual or not is up to chance however most of the time we end up with fitter individual, especially with a larger population size. Now in this scenario where the crossover rate is 0 there isn't much variance in the population of each generation because crossover to attempt to increase our fitness value and bring about our most fit individual(20) is non-existent. Instead, our variance is introduced

through mutation. In our runGA function, we have a parameter mutation rate that determines the rate at which a single point mutation will occur. Essentially, it will change one of our values in our genome from a 0 to a 1 or vice versa. We gave the runGA function a mutation rate of 0.001. This value is extremely low and makes the probability of single point mutation next to nothing. Even so, a single point mutation only changes one value in an entire genome whereas crossover introduces 2 entirely new genomes with half the values being varied since we chose our crossover point to be right at the center of our genome.

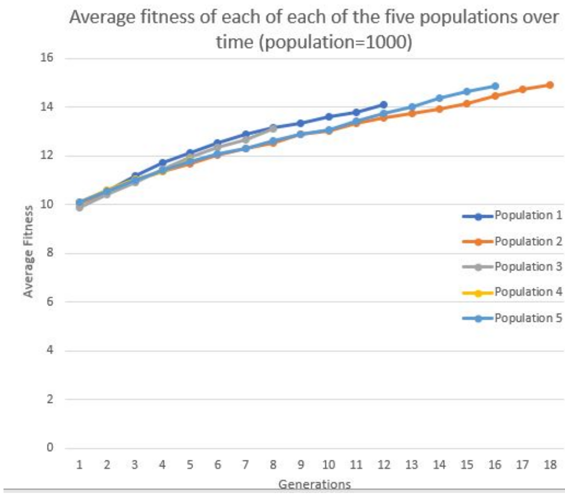
4. Initial Parameters Population = 100, Mutation Rate 0.001 and Crossover 0.7
   Using initial parameters Avg = 27.08, Min =7, and Max = 50



Population from 100 to 1000:

Let us start with varying the population from 100 to 1000. After running this parameter in our runGA function, I noticed that the number of generations required to converge to our fittest individual(20) is a lot less than with the population being 100. The average came out to 11.68 with a max of 18 and a min of 4. Intuitively, this makes sense because when we increase the population with the assumption of randomization, we have a larger gene pool and hence a higher chance to have an already fit individual in the population. From there it isn't hard to extrapolate with both our crossover and mutation to get to our fittest individual of 20. In the run, our initial generation #0 already starts at a best fitness of 16/17 which is much higher than when population = 100 where we started with a best fitness somewhere between 9.5 and 10.5.

See graph below of average fitness of each of the five populations over time. Can see a quicker convergence when pop = 1000 in comparison to initial graph where pop=1000.

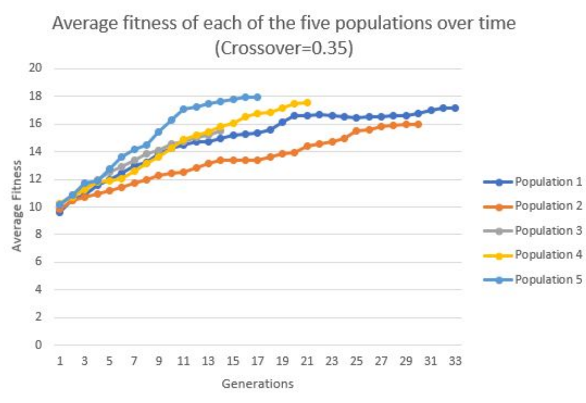


When population = 1000

Crossover 0.7 to 0.35 to 1:

Let us start with varying the crossover from 0.7 to 0.35. In this run, we see an increase in the time it takes to converge to our fittest individual(20). The average came out to 38.62 with a max of 50 and a min of 12. This makes sense intuitively because the probability of a crossover happening is half what is was previously. This means that within our population less of the individuals will be varied towards greater fitness because the rate of crossover is half the probability of what it was before.

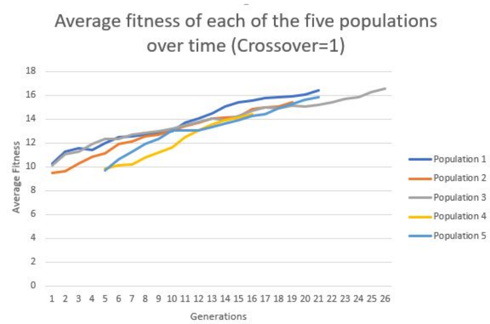
Can see in the diagram below that convergence time is longer for crossover rate = 0.35 when average fitness is plotted for 5 random populations over time compared to initial graph where crossover rate = 0.7



When crossover rate = 0.35

Let us start with varying the crossover from 0.7 to 1. In this run, we see a decrease in the time it takes to converge to our fittest individual(20). The average came out to 22.3 with a max of 50 and a min of 6. This makes sense intuitively because now we have crossover every single time. Every single generation will have a crossover between the two fittest individuals. This increases the variation in the gene pool and allows us to get to the fittest individual more quickly.

Can see in the diagram below that convergence time is quicker for crossover rate = 1 compared to initial graph where crossover rate = 0.7

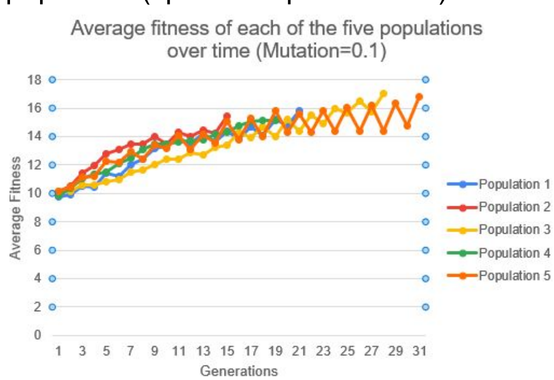


When crossover rate =1

Mutation from 0.001 to 0.1 to 1:

Let us start with varying the mutation from 0.001 to 0.1. In this run we see a slightly longer time it takes to converge to our fittest individual(20). The average 32.2 with a max of 50 and a min of 10. This makes sense intuitively because the probability in either case is fairly small with respect to the population and considering the fact that it is only a single point mutation(changing) one variable there still can't be all that much variance in the population. Since the mutation is such a small change(in a single 0 or 1) it would actually not do much to the population and in fact might even hurt the population by changing a bit in a super fit individual from 1 to 0 making it even less fit and working counterproductively. The changes are so minute that it caused the runtime to increase very slightly due to making fitter individuals less fit

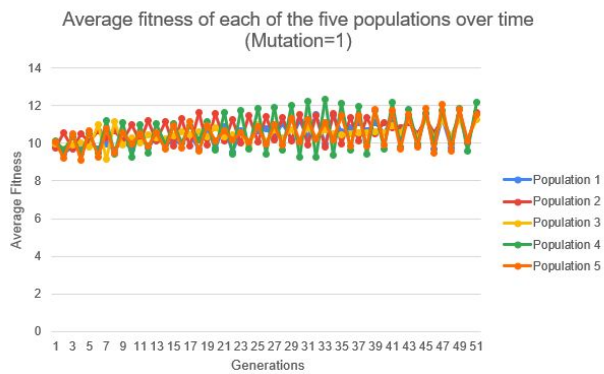
Can see in the diagram below that mutation = 0.1 has a similar convergence time to initial graph where mutation = 0.001. Can also see big variations in fitness of population(sporadic up and down) which is caused by single point mutations



When mutation rate = 0.1

Let us start with varying the mutation from 0.001 to 1. In this run we see an inability to converge to our fittest individual(20). The average comes out to 50 with a min of 50 and a max of 50. Essentially what this is saying is that, at least at 50 generations, we are unable to converge to the fittest individual with a mutation rate of 1. This makes sense intuitively because we are performing literal point mutations for each individual in each generation which is consequently causing the fit individuals to become more unfit. One small point mutation can't cause a whole individual to become extremely fit and have all ones in its bits. If you think of this as analogous to cancer caused by mutated cells, we

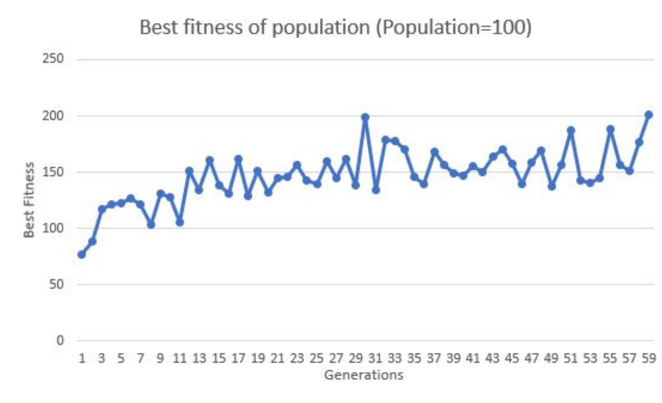
understand that mutations are harmful to the fitness of a population which explains the performance. Mutations are never good and hence do not enhance fitness.
Can see in the diagram below that when mutation rate = 1 we do not converge to 20 after 50 cycles in comparison to initial graph where mutation rate =0.001. The average fitness is far too low for convergence. Also very sporadic changes in fitness.



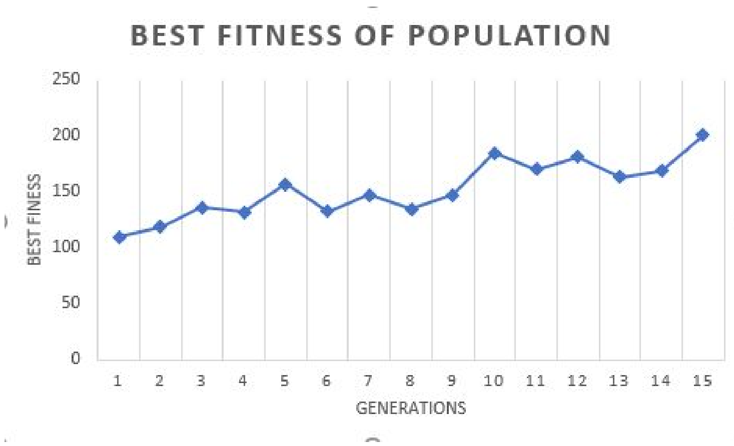
When mutation rate = 1

Part 2

Initial Parameters Population = 100, Mutation Rate 0.001 and Crossover 0.7
Let us call an arbitrary such that when our fitness value > 200 we have achieved a good strategy to Robby picking up the cans. We will measure how good our strategy is by looking at the number of generations it takes to find a fitness value of greater than 200. I decided to do a proof of concept by graphing best fitness over generations until we hit a best fitness of 200(a decently good strategy)
Here is that graph given the initial states given above:

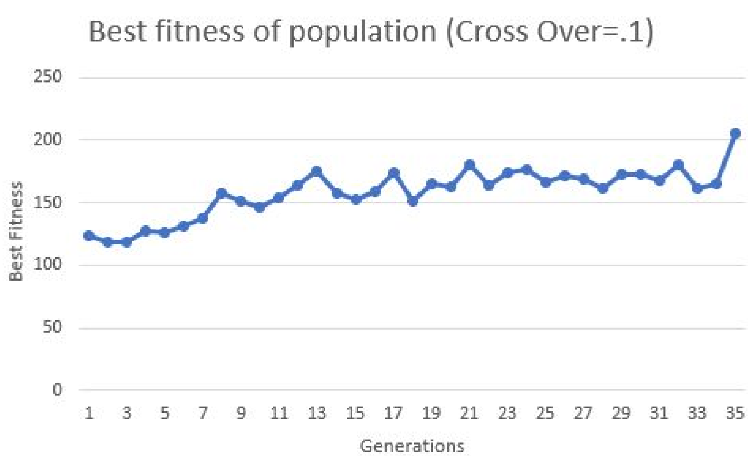


Population size 100 to 1000 :When we changed our population from 100 to 1000 we achieved our best strategy with a fitness of 126, far better than strategyM and the best strategy of population = 100 which was only a mere 63. This intuitively makes sense because we as we increase the number of strategies Robby tries(or individuals in the population) we have a more likely chance to get a really good strategy. As mentioned earlier, I also looked at how long it would take for each of these iterations to reach a fitness value of > 200. We can obviously see in the diagram below that the number of generations it took to get to a fitness value of 200 is much less on the 1000 population than the 100.



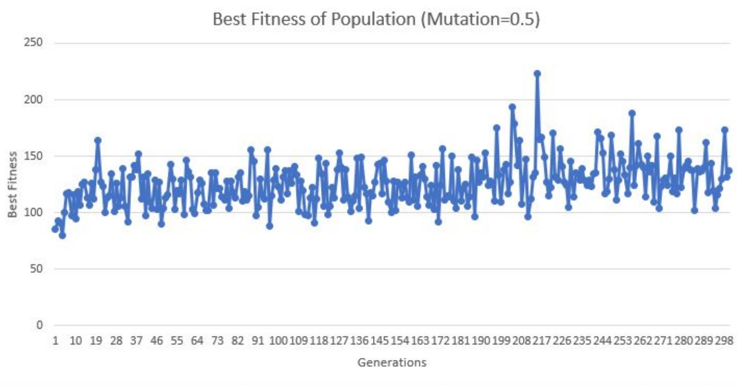
Population = 1000

Crossover rate 1 to 0.1: When we changed our crossover rate from 1 to 0.1 it took much longer for us to reach a fitness of 200 which was at generation 35. This makes sense intuitively because if we decrease the rate at which crossover occurs there is going to be less variation in the population. So, unless we got lucky and have some super fit individual in the beginning it is going to take longer for us to reach a fitness of 200. We need to crossover individuals to maximize their fitness. We can obviously see in the diagram below that it takes more generations to reach a fitness of 200 in comparison to when the crossover rate was 1.



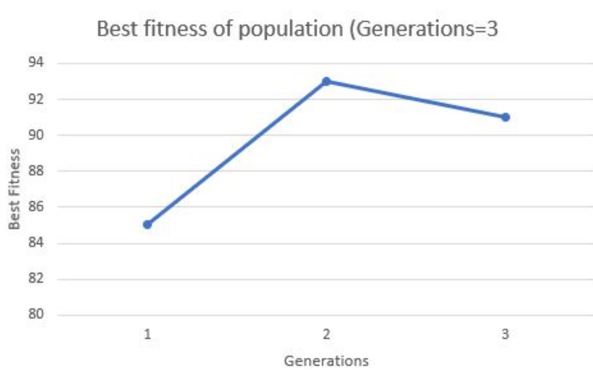
crossover rate = 0.1

Mutation Rate 0.005 to 0.5: When we changed our mutation rate from 0.005 to 0.5, we do not even achieve a fitness of 200 and our fitness for the strategy comes out to 21. This is essentially saying that the rate of a mutation happening at each generation is 0.5 This makes sense intuitively because when we perform a single point mutation, we are varying the individual strategy by 1 step which is not enough to create any significant variation in the population. In fact, it may even hinder the strategy because even one wrong step can throw our whole path off and have us take longer to converge to a fitness of 200. In this case our rate of mutation is so high that it is simply impossible to converge to a fitness of 200 due to the sporadic nature of a single point mutation. We can see in the diagram below a lack of convergence to 200 and the sporadic(up and down) nature of a single point mutation with respect to the best fitness.



mutation rate = 0.5

Generations 300 to 3: When we changed our generations from 300 to 3 we do not achieve a fitness of 200. This makes sense intuitively because it makes zero logical sense that Robby would be able to have a strategy that can already have a fitness of 200 unless there is already some super fit individuals in the population. The diagram below shows that we hardly achieve a fitness of 91 when we only allow 3 generations of strategies to run.



generations = 3

Overall the GA is fairly efficient and it seems like increasing population, keeping crossover rate high and mutation rate low gives us our most fit individuals or allows us to converge to fitter individuals more quickly.