AntColonyTSP White Paper

The ant colony algorithm implements two modes: online delayed, and offline. At the start, there is a population of ants which is spread evenly among 29 cities, and for the graphing section, a default of 200 ants will be used unless specified otherwise. Then, for each generation—in which there are 250 in total—the ants get randomly picked to traverse every city exactly once, and finish at their starting city.

If in online delayed mode, each time an ant does a traversal, pheromone is evaporated and then placed on its path based on the inverse of the length (1 / len). However, if it is in offline mode, each iteration, pheromone is evaporated, and then pheromone is placed on the path of the best traversal using the same inverse length equation (1 / len). The difference here is that in online delayed mode, once each ant does a traversal (which occurs each iteration), the pheromone will evaporate, and then pheromone will be placed on the path which the ant traversed. However, in offline mode, this does not occur to every ant each iteration, but rather only to the ant which had the fittest (least costly) traversal.

Every time that evaporation takes place, the amount of pheromone which disappears is equal to the evaporation factor, meaning that the amount which stays is one minus the evaporation factor; thus, if the evaporation factor is 0.1, then each time evaporation takes place, 90% of the pheromone stays, and 10% evaporates. The default pheromone placed on each path when the simulation starts is 1, and the default evaporation factor for the graphs below unless specified otherwise is 0.2.

Each time an ant is to move, it is based on the transition control, so that if a randomized number in the range of [0, 1) is picked and is less than the transition control, the best move is always taken, but otherwise the path is picked based on roulette. When the best path is chosen, each path is calculated according to the pheromone divided by the length to the beta power, and the calculation which is the highest gets picked for the ant to go to that city. For roulette, the calculation is based on the pheromone to the alpha power divided by the length to the beta power, and each city has a chance to get chosen based on its number relative to the sum of the calculations of all cities. Alpha is 0.5 and beta is 1, and the default transition control parameter is 0.5 for the calculations below, unless specified otherwise.

For the evaporation factor, an evaporation constant of 0.05, 0.1, 0.15, …, 0.95 was tested, with each value being tested 25 times, and the median being reported. This means that each point on the graph represents the median of 25 trials with the same parameters. These tests use the default parameters as specified previously, and the only parameter being changed is the evaporation factor. The reason the median is being used and not the arithmetic mean is because when someone would use this algorithm, they would call it once and use that result, so the median shows the value which the user would be expected to receive. The graph for evaporation is present in Figure 6.

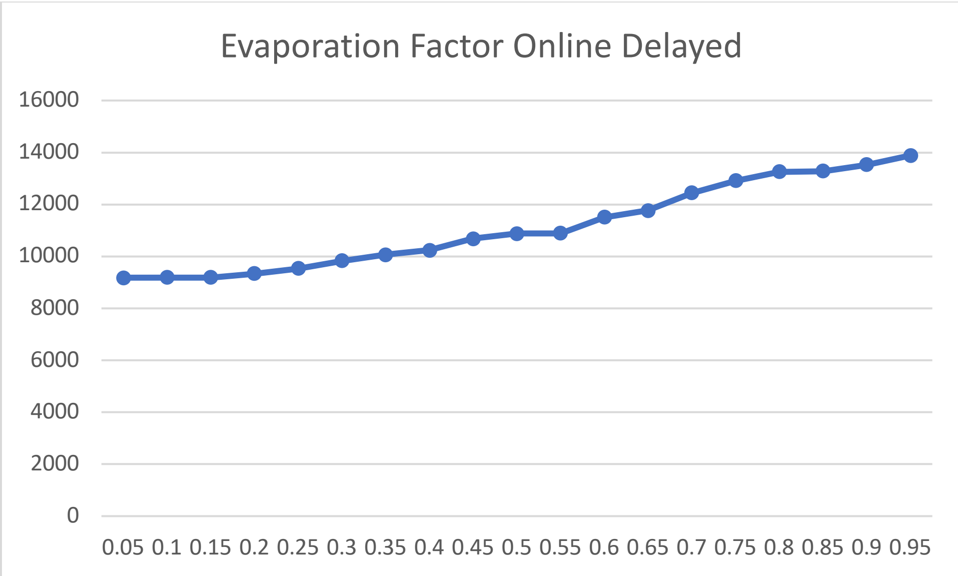

Figure 6. Evaporation factor online delayed.

As the evaporation factor gets greater than 0.15, the fitness gets worse. The difference between the best fitness and worst fitness is quite large (nearly 5000), and there is a clear trend. To see exactly what evaporation factor is best, another set of trials has been conducted from 0, 0.01, …, 0.15 but this time the result is the median of 75 trials for each data point. This is present in Figure 7.

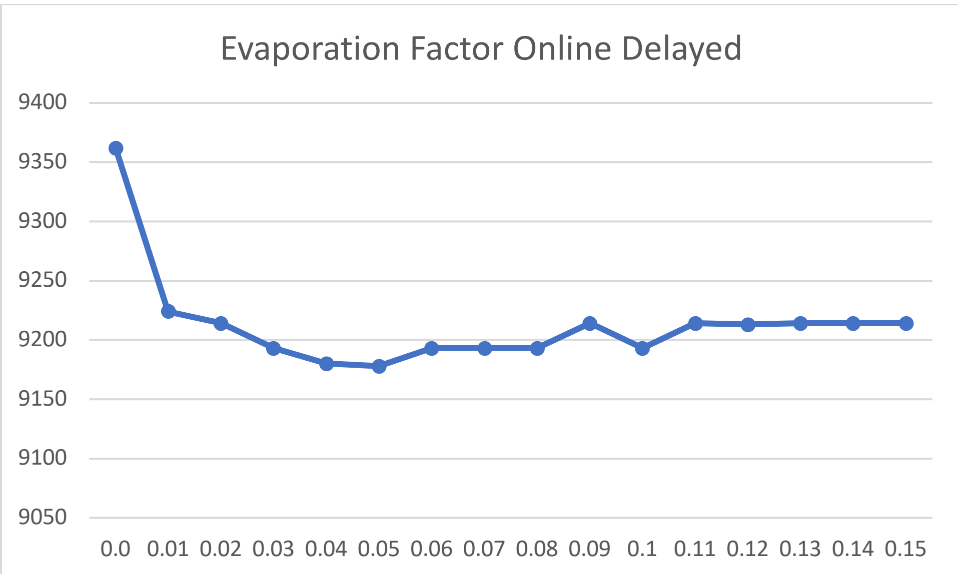

Figure 7. Evaporation factor online delayed small values.

Therefore, we can see from this graph that the best evaporation factor is 0.05 since it has the lowest cost (best fitness). The reason that higher evaporation factors are not as good is because the pheromone will leave too quickly and will not be that useful. The reason low evaporation factors are bad is because if it does not evaporate at all, the pheromone for bad solutions will stay for too long. However, from this graph, in total, the changes in the fitness from changing evaporation factor is very small. As long as evaporation factor is in the range of [0.03, 0.15] then it is approximately ideal.

The next variable to modify is the transition control, and it is from 0.05, 0.1, …, 0.95 with each value on the plot being the median of running the same algorithm with the same parameters 25 times. This graph is present in Figure 8.

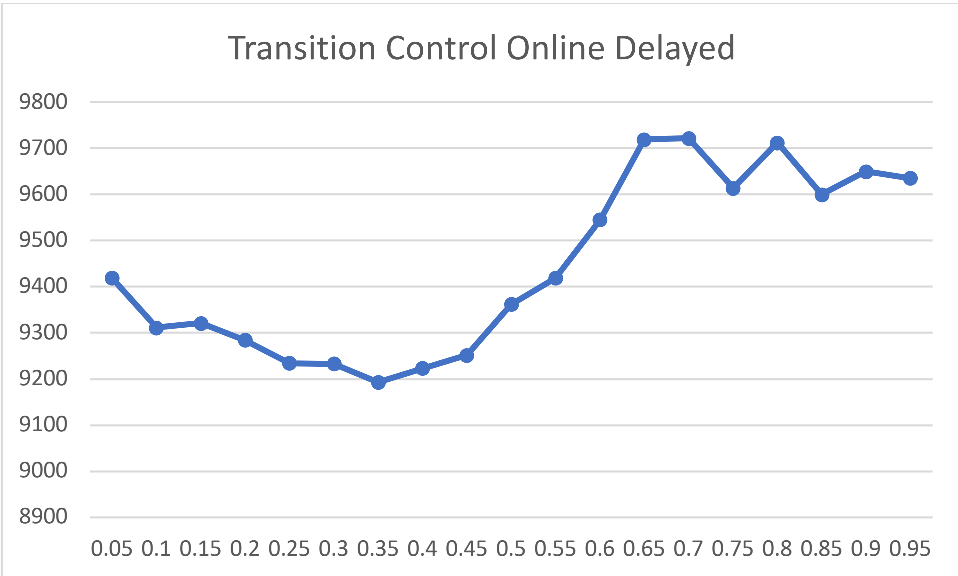

Figure 8. Transition control online delayed.

Thus, it can be seen that the best transition control is 0.35, and as it is decreased or increased, the fitness gets worse. At 0.35, each time an ant moves, it has a 35% chance of taking the best move, and a 65% chance of exploring the possibilities based on a roulette decision. The reason low transition control is bad is because all the moves are based on roulette, and it does not guarantee enough known good solutions. However, a too high transition control is bad too because then everything is based only on the best move, and there is no exploration. Additionally, since the difference between the worst and best fitness is rather large (over 500), there is a clear trend.

The final parameter to look at for online delayed is the population size. With the same logic as the last two parameters, each parameter is run 25 times and the median is taken, with the values being 25, 50, 75, …, 500. This is present in Figure 9.

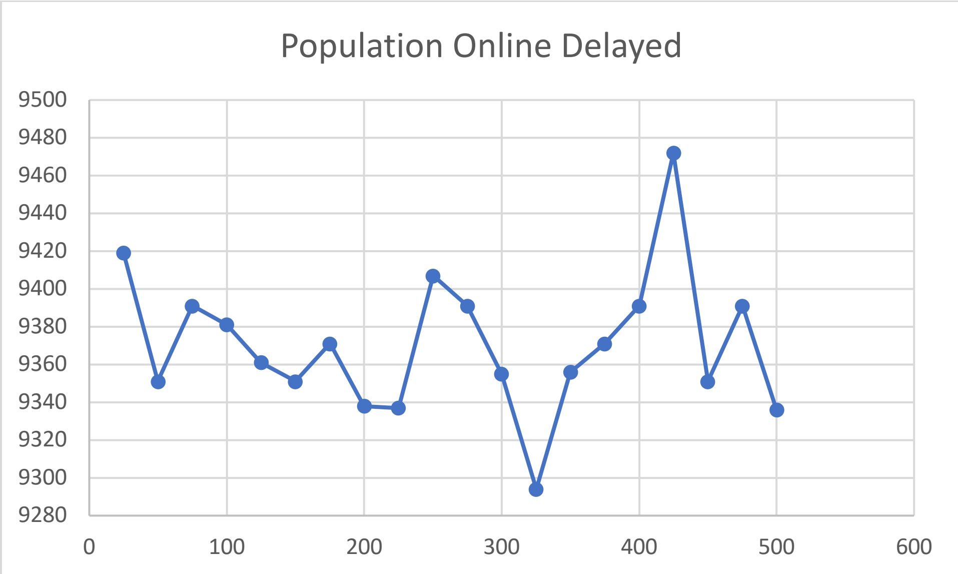

Figure 9. Population online delayed.

This figure does not show a clear trend, indicating that population might not have an effect on fitness for online delayed. In order to verify if this is randomness, or if there is a trend, it will be run again, but with the following data points: 29, 58, …, 493, meaning that this time there is a step of 29 instead of 25. This can be seen in Figure 10.

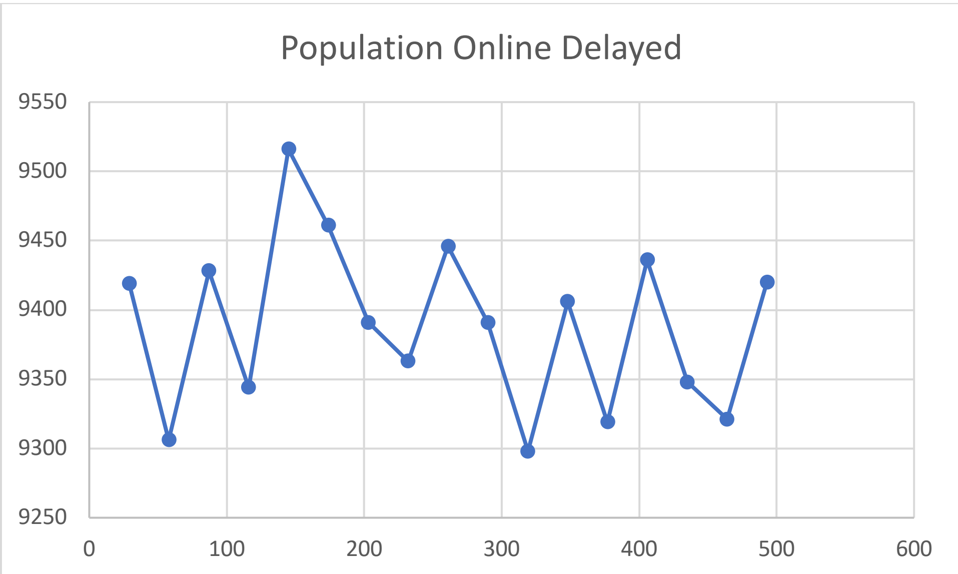

Figure 10. Population online delayed, trial 2.

Thus, there is no correlation between the population size and the fitness for online delayed mode.

Next, looking at offline mode, the first parameter to be tested is the evaporation factor, as shown in Figure 11. Each data point is the median of 25 trials.

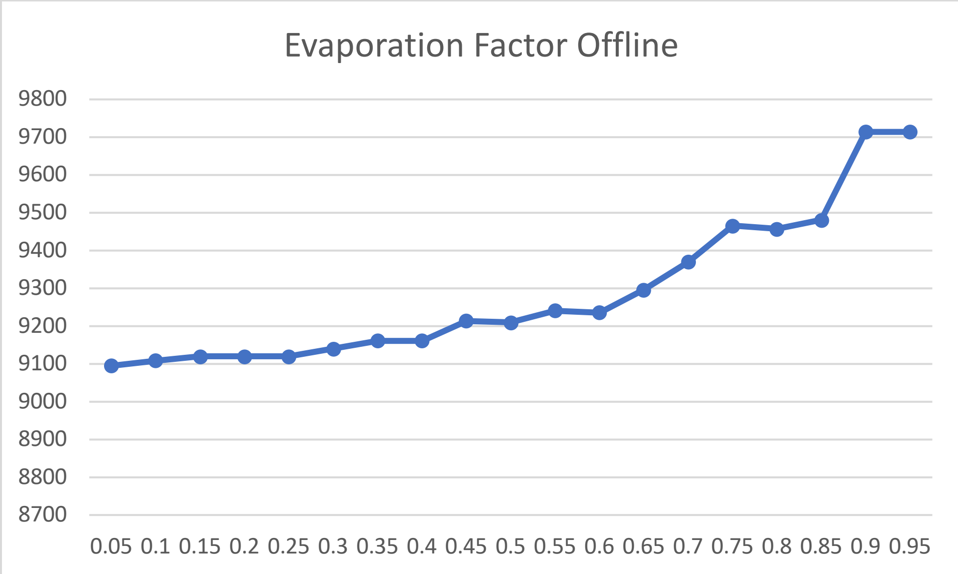

Figure 11. Evaporation factor offline.

This shows that having too high of an evaporation factor leads to sub-optimal solutions. This result is significant because the difference between the best fitness and the worst fitness is over 600. To see exactly what evaporation factor is best, another set of trials has been conducted from 0, 0.01, …, 0.15 but this time the result is the median of 75 trials for each data point. This is present in Figure 12.

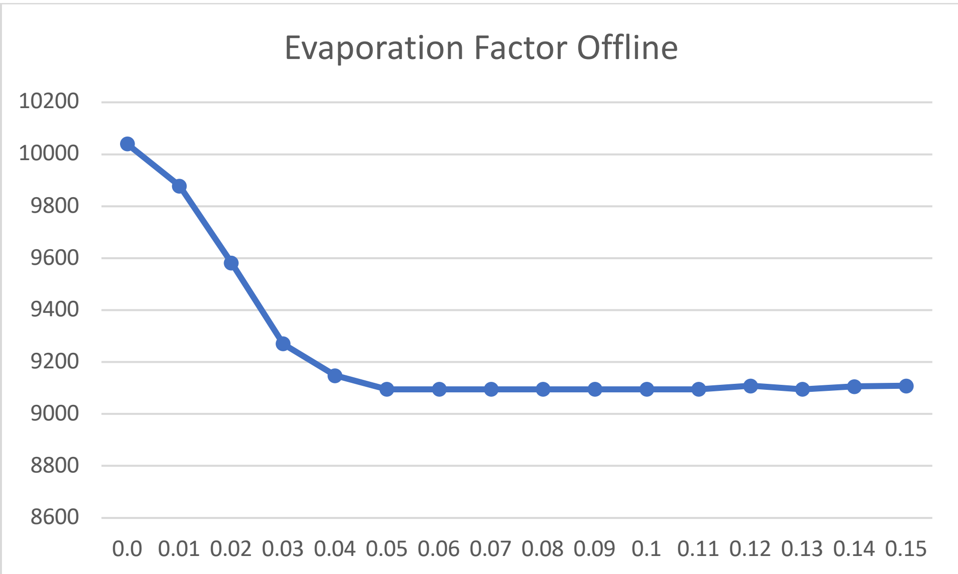

Figure 12. Evaporation factor offline small values.

Having a too small or large evaporation factor is not optimal. From this graph and the last one, as long as the evaporation factor is in the range of [0.05, 0.15] it is a good value. The single value to pick would be 0.1 since it is halfway in the range. It is slightly different that the online delayed mode, because for that 0.05 was picked, but the reasoning remains the same.

The next parameter to test is the transition control in offline mode, again each data point is the median of 25 trials. This is present in Figure 13.

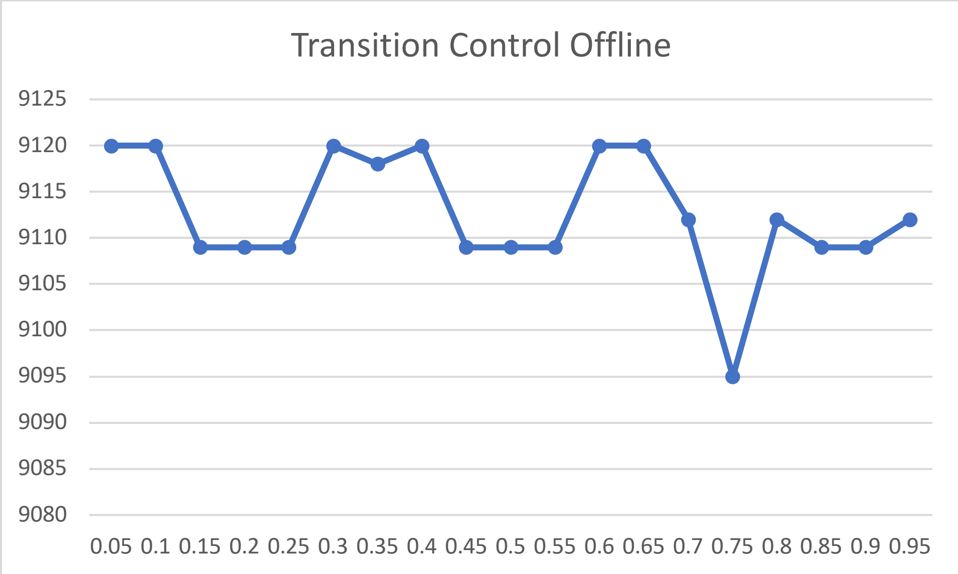

Figure 13. Transition control offline.

This shows that there is no link between fitness and transition control for offline mode, because the difference between the best fitness, which is 9095, and the worse fitness, which is 9120, is only of 25. However, to be sure that there is no link, it has been run again and is present in Figure 14.

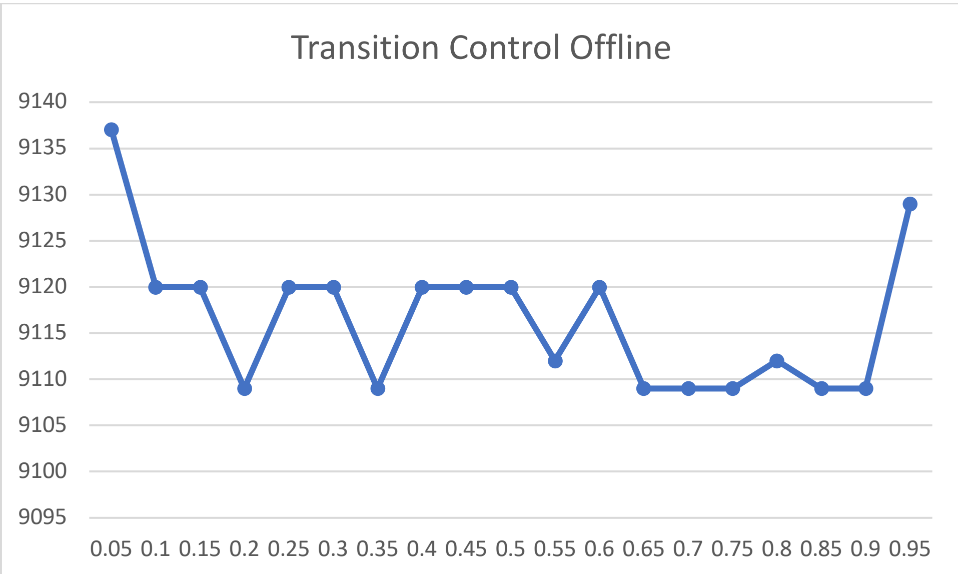

Figure 14. Transition control, trial 2.

This shows, again, that the difference between the best and worst fitness is quite low, and also shows that there is no trend. This shows that the transition control does not affect the fitness for offline. This makes sense, since for offline, pheromones are placed on the path for the best solution in each generation, which means that the transition control does not matter.

The final parameter to test is the population size for offline mode. Again, this is the same conditions as online delayed mode. This is present in Figure 15.

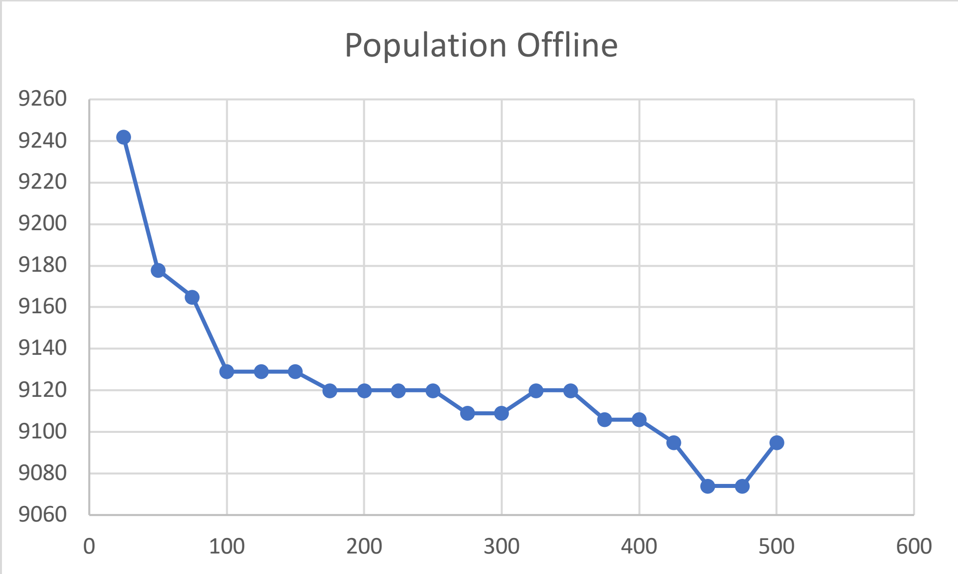

Figure 15. Population offline.

This shows exactly what is expected to occur with population, which is that the fitness should get better as the population increases. Population 500 has a strange uptick, which is likely simply due to randomness. However, since the difference between the worst and best fitness is only slightly higher than 140, it will be run again to verify that it is not just due to incredibly unlikely statistical chance (because keep in mind, almost every point, which is the median of 25 trials, shows a pattern). Figure 16 shows another trial, but this time 29, 58, …, 493.

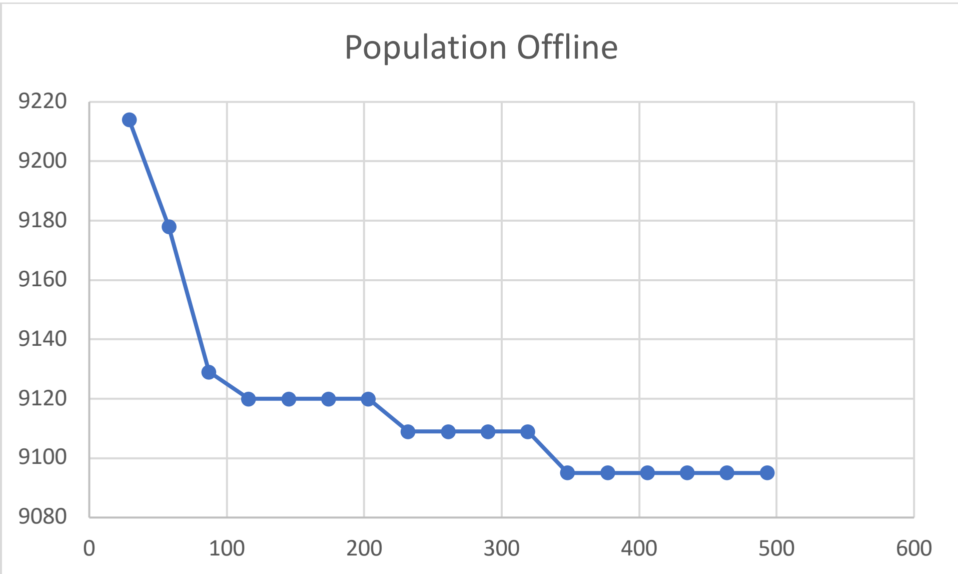

Figure 16. Population offline, trial 2.

As shown in the graph, there is a trend with population, meaning the greater the population is, the better the fitness is. The difference between the worst and best fitness is not massive, but it does exist, and it is reproducible.