Homework 4
Report
Jesse Harper

Introduction 1.1:

The problem solved in this section is on finding the median number of people required in a group to have two people with the same birthday. The definition of same birthday, for this purpose, means that the two people have birthdays separated by less than seven days. The problem was solved using a method of artificial intelligence called monte carlo simulation. After running the simulation it was found that a median of 21 people was required to have a group with two people that had the same birthday.

Models and Methods 1.2:

The method used to find a solution to this problem was *Monte Carlo Simulation*. This method employs random search to find a solution to problems, such as the one found. First birthdays are enumerated as integer values from 1 to 365. Then a group is started as a collection of integers. The group is expanded by one, with a randomly generated integer from 1 to 365. Each member of the group has a birthday. The group is expanded until a new member is generated that has the same birthday as another previously added member. The number of members is recorded in an array and a new group is formed using the previous process. For this simulation, 10,000 iterations were performed, however any number of iterations on this simulation can be performed.

A helper function *search_group()* was created to help the iterative process. This function would be called after adding a new member to a group and return true if two members of the group had the same birthday. The current *group* would passed into the function and this group was searched and the newest addition to the group would be tested against the previous ones.

Calculations and Results 1.3:

Here is sample output for a simulation with a 10000 iterations.

```
Median Number of People:  21
```

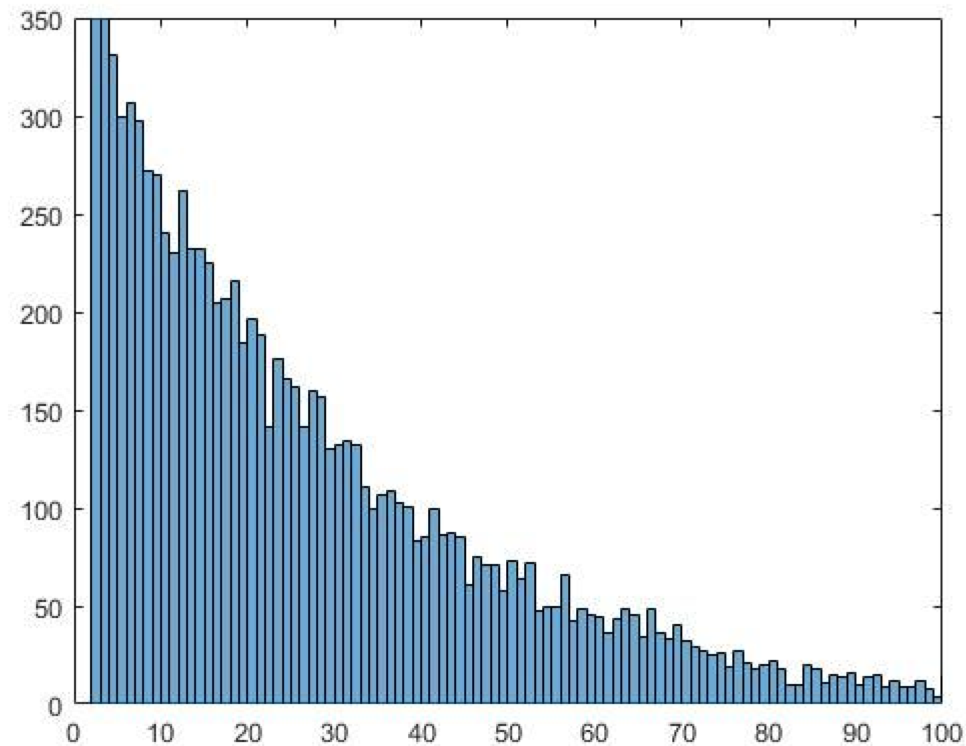

Table 1.3.1: Histogram of number of groups versus number of people required

Discussion 1.4:

The median number of people required, as found through the use of this simulation, was 21. This number seems very high, especially when you look at the average value of the same data set. The mean number is only 3 when you round to the nearest integer. The mean is reflected in the histogram, table 1.3.1, which shows very large numbers of groups with less than 10 people. Given that there are certain periods of the year when more people are likely to be born, I believe it is more likely that the median should go down if you take into account this fact.

Introduction 2.1:

This problem examens the traveling salesman problem. The traveling salesman problem is that given a set of points needed to be visited in a closed path, what is the order of points you visit in order to minimize total path distance. The problem was solved using a variation of monte carlo and genetic algorithms called *simulated annealing*. Though a few different starting numbers of points were considered, this problem was solved for a square unit grid and 10 points which need to be visited. The shortest path distance found for this was approximately 2 units, but that the way random numbers are generated have a large impact on the solution found.

Models and Methods 2.2:

Simulated annealing is a variation of monte carlo simulation and genetic algorithms which helps to reduce the likelihood of becoming stuck in a local minimum which is not the global minimum. The way this works is that while only randomly generated solutions that are less than the previous one are kept for genetic algorithms, they are not always kept for simulated annealing. If a better solution is found then then a probability function generates a number which allows the possibility of discarding good solutions to allow for more variation. Equation 2.2.1 shows the probability function.

$$P = exp(\frac{-c\Delta L}{T})$$ **Eqn. 2.2.1**

If P is greater than a randomly generated number, then the solution is discarded.

The Temperature, T, is reduced with each iteration until the likelihood of P < than any random number is effectively zero. T is reduced with each iteration using a equation 2.2.2.

$$T_{new} = T(1 - \lambda)$$ **Eqn. 2.2.2**

Calculations and Results 2.3:

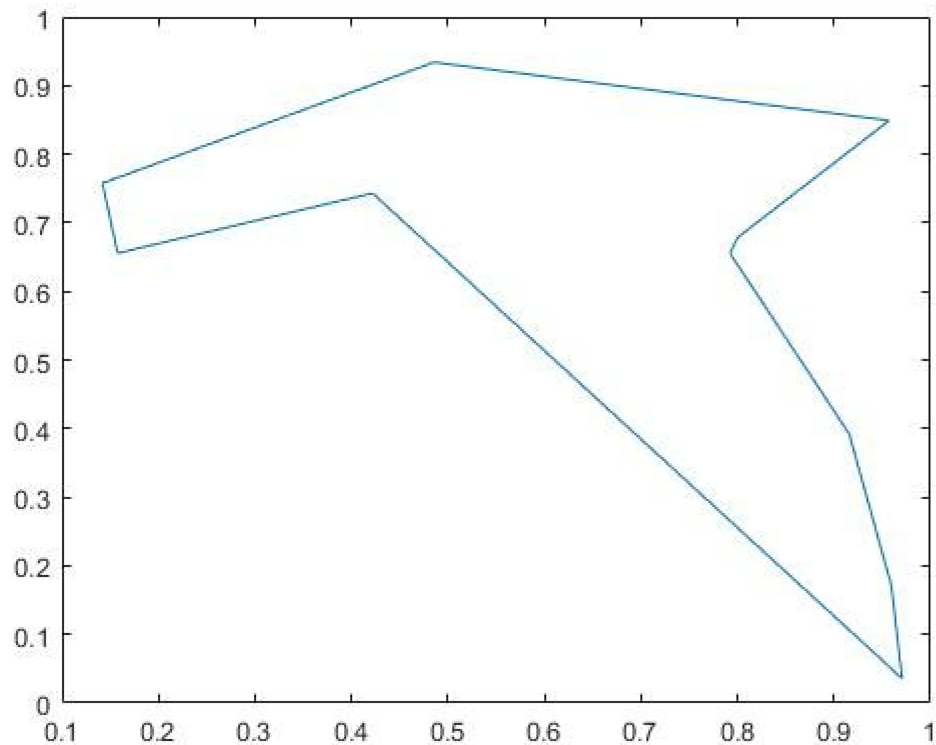

Figure 2.3.1: final path created using N =10 and default rng

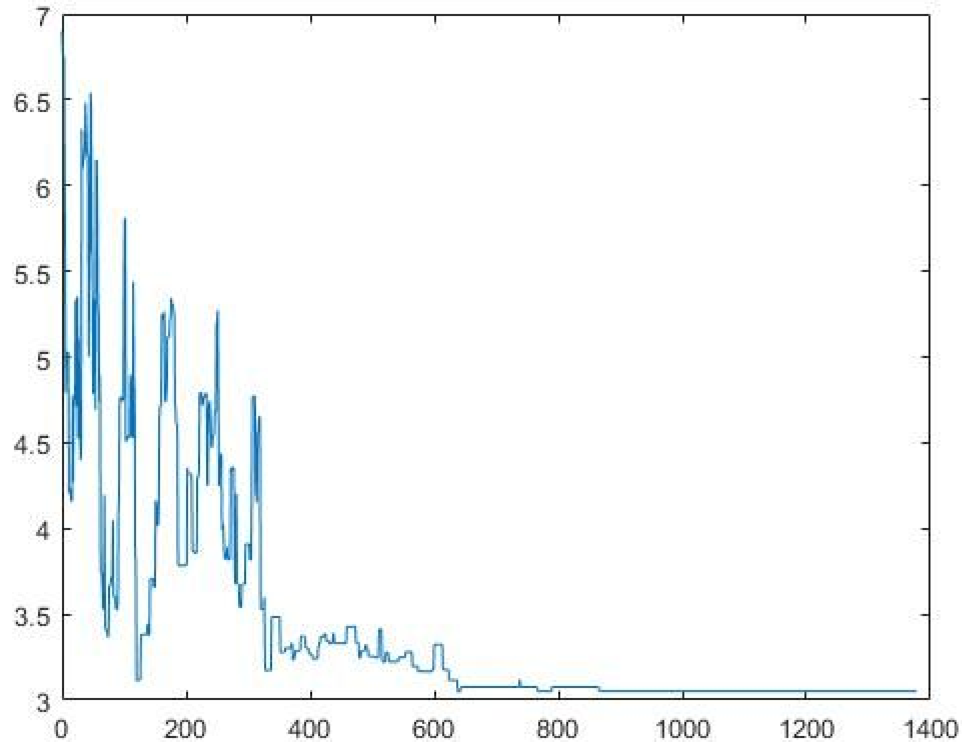

Figure 2.3.2: Total path distance by iteration N = 10 default rng

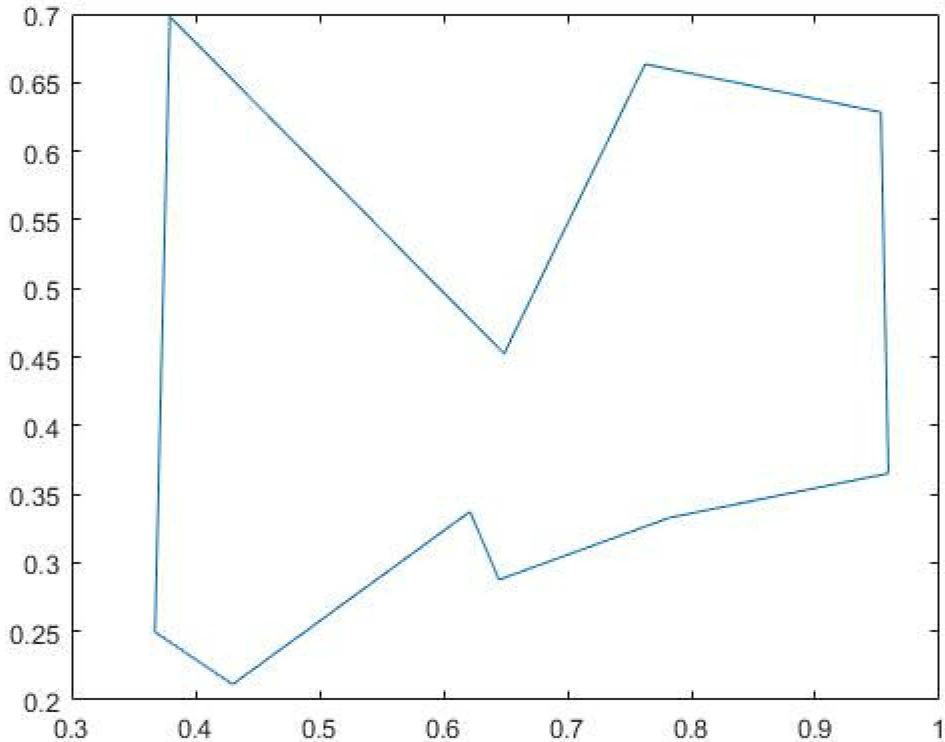

Figure 2.3.3: Path generated using N = 10 and shuffle rng

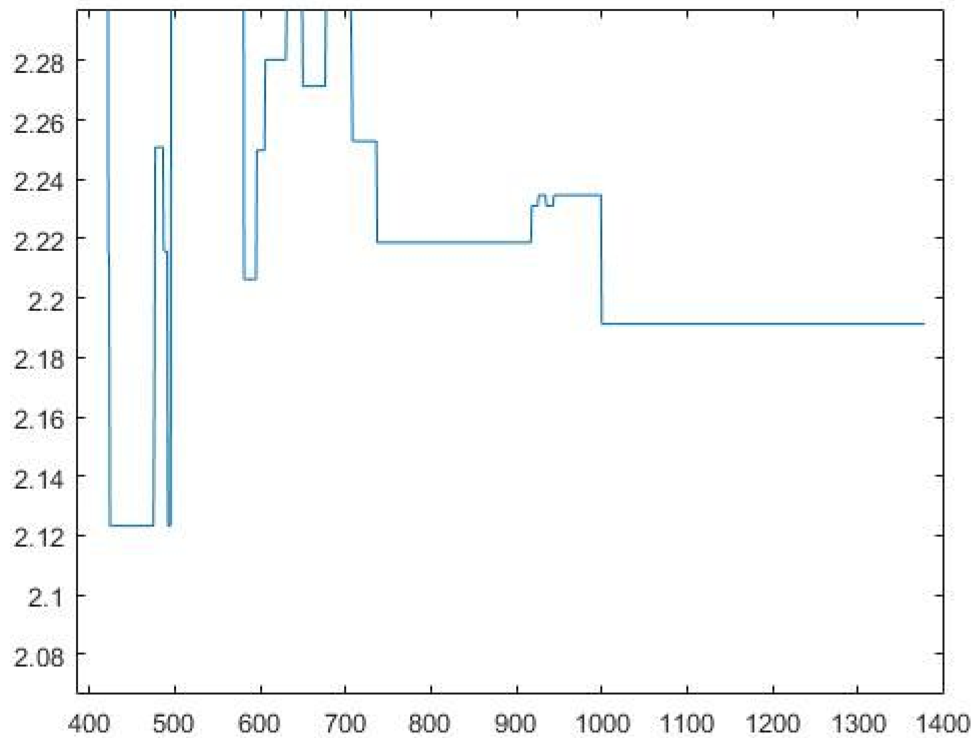

Figure 2.3.4: Total path distance by iteration N = 10 shuffle rng

Discussion 2.4:

The simulated annealing approach to the traveling salesman problem seems to give at least good solutions is a relatively short period of time. The answers are not always the best solution, as can be seen from figure 2.3.4. This figure shows a plot of the total distances and that lower path was actually found, but discarded by the simulation. The program is still guessing and checking randomly for a solution, even though the algorithm tends to keep better solutions, a better solution can be discarded randomly. The random discarding of solutions is meant to prevent getting stuck in local minimus, but in this case it hurts the performance of the program.

Varying the parameters lamda and c, from equation 2.2.1, affect the behavior of the simulation. Lambda controls how many times a random discarding of a good mutation occurs. Raising it means the Temperature lowers slower, which increases the number of opportunities for random jumps from a local minimum. However increasing c decreases the probability that a good solution will be discarded when a good solution is found.