**Homework 2**

Ali Abbas

Ahsan Sanaullah

Fernando Trevino Ramirez

**Abstract**

In this paper, we explore how different representations on a Genetic Algorithm (GA) can affect its performance.

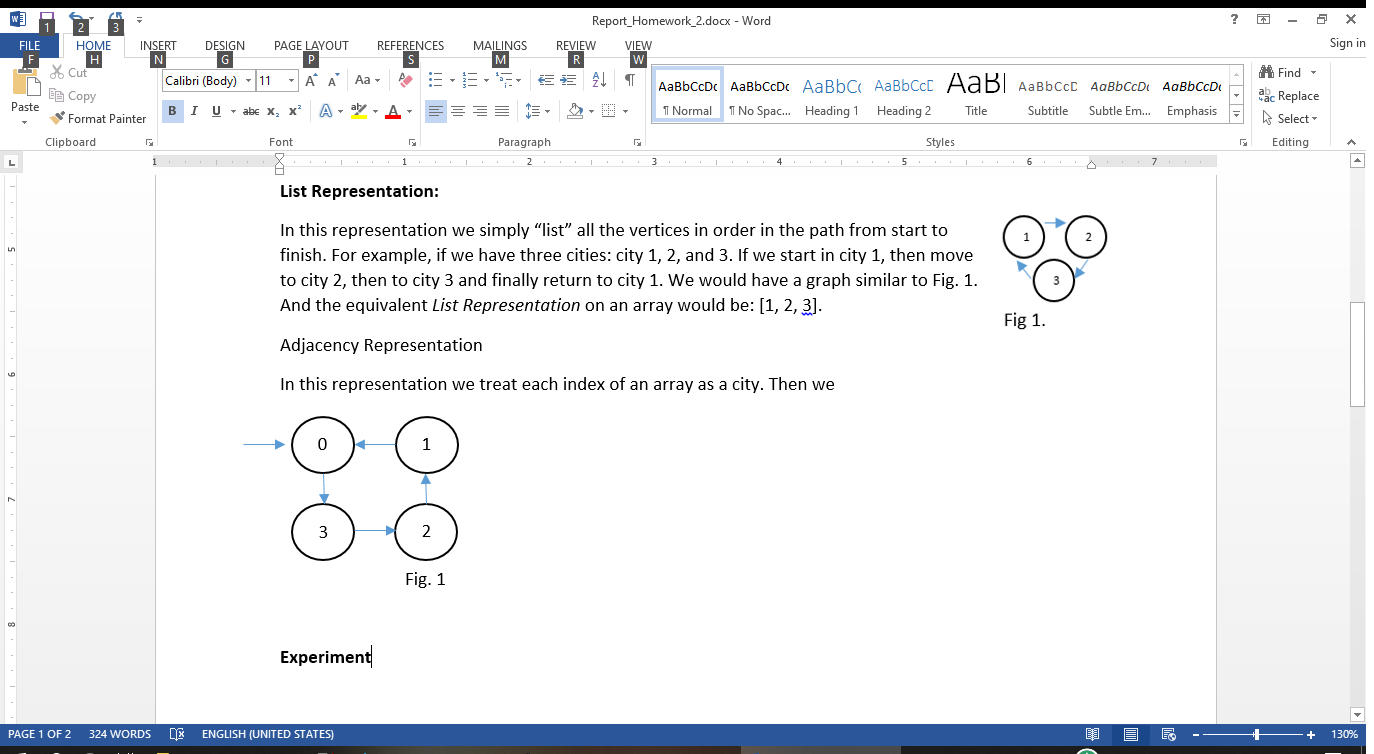
**Introduction**

To test the importance of choosing the correct representation to a problem intended to be solved with a GA, we first start by finding a suitable problem. The problem assigned by our professor Annie Wu [1] was the famous Traveling Salesman Problem or TSP for short. Then we were allowed to choose two different representations: List and Adjacency were the representations we decided to use for this project. Since we were given the coordinate point for each of the cities, we assumed that all cities are connected and thus we initialized a 2 dimensional array with the distance between each vertex at the beginning of each experiment. Where distance d is defined as:

**Traveling Salesman Problem (TSP) review:**

Given a number of cities n, find a path that goes through all the cities exactly one time and ends in the first city. In other words, all cities must be visited and the graph must contain exactly 1 cycle. If any of this conditions isn’t met, then the solution is considered to be invalid. For more information we recommend a book we found called “The Traveling Salesman Problem” [2].

**List Representation:**

In this representation we simply “list” all the vertices in order in the path from start to finish. For example, if we have four cities: city 0, 1, 2, and 3. If we start in city 0, then move to city 3, then to city 2, then 1, and finally return to city 0. We would have a graph similar to Fig. 1. And the equivalent *List Representation* on an array would be: [0, 3, 2, 1, 0]. But since we always return to the city we started, we can omit the last 0, leaving us with this array: [0, 3, 2, 1].

**Adjacency Representation:**

In this representation we treat each index of an array as a city. Then the integer stored in each given index is the city after the city . The assumption in this representation is that We always begin at city 0. To see which city comes after city 0, we must check the value of the array at position 0. Suppose it is city , the city after city is stored in position of the array and so on. For the same example as before, our array would look like this: [3, 0, 1, 2].

**Experiments**

In this project we compare two approaches of solving the Traveling Salesman Problem with a GA, using two representations of a possible path solution. For the list representation, we can use very straight forward crossover and mutation operators. But for the adjacency representation, it’s harder to apply crossover or even mutation without applying a repair function to get a valid solution.

**Results**

By looking at the graph on image 1…

**Conclusion**

In conclusion, we found out that using different representations…

**Extensions**

By only testing the GA on the traveling salesman problem and only using two representations, we limit the generality of the results. To solidify our results we would need to: Experiment in a bigger range of problems. And expand the number of representations per problem.

**References**

[1] Hal Stringer & Annie Wu (2004). “A Teaching GA” [Computer software]. Florida, Orlando: UCF.

[2] Applegate, D. L.; Bixby, R. M.; Chvatal, V.; Cook, W. J. (2006), “The Traveling Salesman Problem”, ISBN 978-0-691-12993-8.