

Thesis Update

Over the past two weeks, I have revised the Python code used to determine matrix consistency. These changes included:

- Recording the position(s) of inconsistent elements in the matrix.
- Recording the position of the largest distance between two matrices.
- Altered distance calculation to handle cases where the denominator between arbitrary items is not the same.

After making these changes, I started to run some experiments. Using a 3x3 matrix I ran a series of simulations that determined the number of detected inconsistent locations. When changing one item in the matrix, five inconsistencies were detected regardless of the position of the inconsistent element. The simulation did not include elements on the diagonal because that would invalidate one of the primary properties of these matrices. When I made the matrix 4x4, eight inconsistencies were detected. When the matrix was 5x5, the algorithm detected 11 inconsistencies.

So far, the number of inconsistencies to the size of the matrix is as follows:

Matrix Size	Inconsistencies Detected
3x3	5
4x4	8
5x5	11
6x6	14

Below is the data from the simulation run. Each inconsistent location is a tuple, where the format of the data corresponding to the algorithm is [i, j, k]

```
Distance from allOnes to allOnes: 0.0
Number of differences: 0
['1', '2', '1']
['1', '1', '1']
['1', '1', '1']
Matrix Diagonal Is 'Good': True
Matrix Is Consistent: False
Number on Inconsistencies: 5
Inconsistent Locations:
[[0, 0, 1], [0, 1, 2], [0, 2, 1], [1, 1, 0], [2, 1, 0]]
```

```
['1', '1', '2']
['1', '1', '1']
['1', '1', '1']
Matrix Diagonal Is 'Good': True
Matrix Is Consistent: False
Number on Inconsistencies: 5
Inconsistent Locations:
[[0, 0, 2], [0, 1, 2], [0, 2, 1], [1, 2, 0], [2, 2, 0]]
```

```
['1', '1', '1']
['2', '1', '1']
['1', '1', '1']
Matrix Diagonal Is 'Good': True
Matrix Is Consistent: False
Number on Inconsistencies: 5
Inconsistent Locations:
[[0, 0, 1], [1, 0, 2], [1, 1, 0], [1, 2, 0], [2, 0, 1]]
```

```
['1', '1', '1']
['1', '1', '2']
['1', '1', '1']
Matrix Diagonal Is 'Good': True
Matrix Is Consistent: False
Number on Inconsistencies: 5
Inconsistent Locations:
[[0, 2, 1], [1, 0, 2], [1, 1, 2], [1, 2, 0], [2, 2, 1]]
```

```
['1', '1', '1']
['1', '1', '1']
['2', '1', '1']
Matrix Diagonal Is 'Good': True
```

```
Matrix Is Consistent: False
Number on Inconsistencies: 5
Inconsistent Locations:
[[0, 0, 2], [1, 0, 2], [2, 0, 1], [2, 1, 0], [2, 2, 0]]

['1', '1', '1']
['1', '1', '1']
['1', '2', '1']
Matrix Diagonal Is 'Good': True
Matrix Is Consistent: False
Number on Inconsistencies: 5
Inconsistent Locations:
[[0, 1, 2], [1, 1, 2], [2, 0, 1], [2, 1, 0], [2, 2, 1]]

Greatest distance value = 1.0
Greatest distance location = [0, 1]
```

Now that I finally have a good simulation framework to draw meaningful conclusions from, the next step is to start experimenting with strategies to take an inconsistent matrix, and alter the inconsistent location(s) until the matrix is consistent. For this, I will start with the 'All Ones' matrix and make a single element be '2' instead of '1'. I will then attempt to write an algorithm that can make the matrix consistent. After the algorithm has found a consistent matrix, the distance between the inconsistent matrix and the newly consistent matrix will be calculated. The goal is to find a consistent matrix with the smallest distance between the original matrix and the new consistent matrix.