CS414 HW2: Percolation Threshold Analysis

Emir Kantul 27041

May 26, 2023

1 Introduction

In this report, I present my approach to completing the given assignment. To achieve this, I followed several steps, which are detailed below:

1. I implemented the `create_bonds` function to generate bonds based on the provided dimensions and threshold probability `pthr`. To do this, I created a 2D numpy array of the given dimensions and iterated through each cell. If a random number was less than `pthr`, I created a bond between the current cell and its neighbors (right and down) if they were within the grid dimensions.

2. I developed the `find_clusters` function to identify clusters of occupied cells in the grid. I employed the Union-Find algorithm or a similar method to group connected bonds together. Afterward, I returned the list of clusters sorted by their size.

3. I implemented the `check_percolation` function to determine if a given cluster percolates. I checked for bonds connecting the left side to the right side (horizontal percolation) or the top to the bottom (vertical percolation) of the grid.

4. I ran simulations for different `p` values using the provided code. This code executed the simulation for a range of `p` values and stored the results in a file.

5. I analyzed the results by performing the following tasks:

    (a) Visualizing clusters on grid systems for different `p` values with small increments and around `pc` using the `visualize_grid` function provided.

    (b) Plotting percolation probabilities for different `p` values ranging between (0,1) by calculating the ratio of percolated simulations to the total number of simulations for each `p` value.

    (c) Plotting average finite cluster sizes for different `p` values by calculating the mean cluster size for each `p` value.

(d) Plotting the mean and standard deviation of `P\inf` (the percolation probability for the largest cluster) for different `p` values.

(e) Estimating `pc` (mean and CI) for different system sizes and plotting `N` vs. `pc` by fitting the data to a function that relates the system size to the percolation threshold.

Upon completing these steps, I gained a comprehensive understanding of the percolation threshold and the behavior of the system for various `p` values and dimensions.

2 Final Version and Results

In the final version of the project, I expanded the range of dimensions and p_values to analyze the percolation threshold more thoroughly. I modified the code to cover different system sizes and used finer increments for p values around the critical percolation probability, p_c. The following changes were made to the initial code:

1. I modified the dimensions to cover a larger range: DIMS = [(50, 50), (100, 100), (150, 150), (200, 200)].
2. I used two different sets of p values for the analysis:
   (a) `pExplore = np.linspace(0, 0.8, 17)[1:]` for exploring the range between 0.05 and 0.75.
   (b) `pExpand = np.linspace(0.45, 0.55, 21)[1:]` for a more accurate estimation of p_c in the range between 0.45 and 0.55.
3. I updated the simulation loop to iterate through different system sizes and p values.

After implementing these changes, I re-ran the simulations and analyzed the results. I visualized the clusters on grid systems for different p values and dimensions and plotted the percolation probabilities, average finite cluster sizes, and mean and standard deviation of P_∞ for the new range of p values. Finally, I estimated the percolation threshold p_c for different system sizes and plotted the relationship between system size (N) and p_c. The updated analysis provided a more detailed insight into the behavior of the system and the percolation threshold.

3 Visualizations of Grids with Different p Values

In this section, we present the visualizations of grids with different p values ranging from 0.48 to 0.56. These visualizations provide a clear understanding of the behavior of the system and the percolation threshold as the p value changes. For each p value, the corresponding PDF file containing the grid visualization is included below.

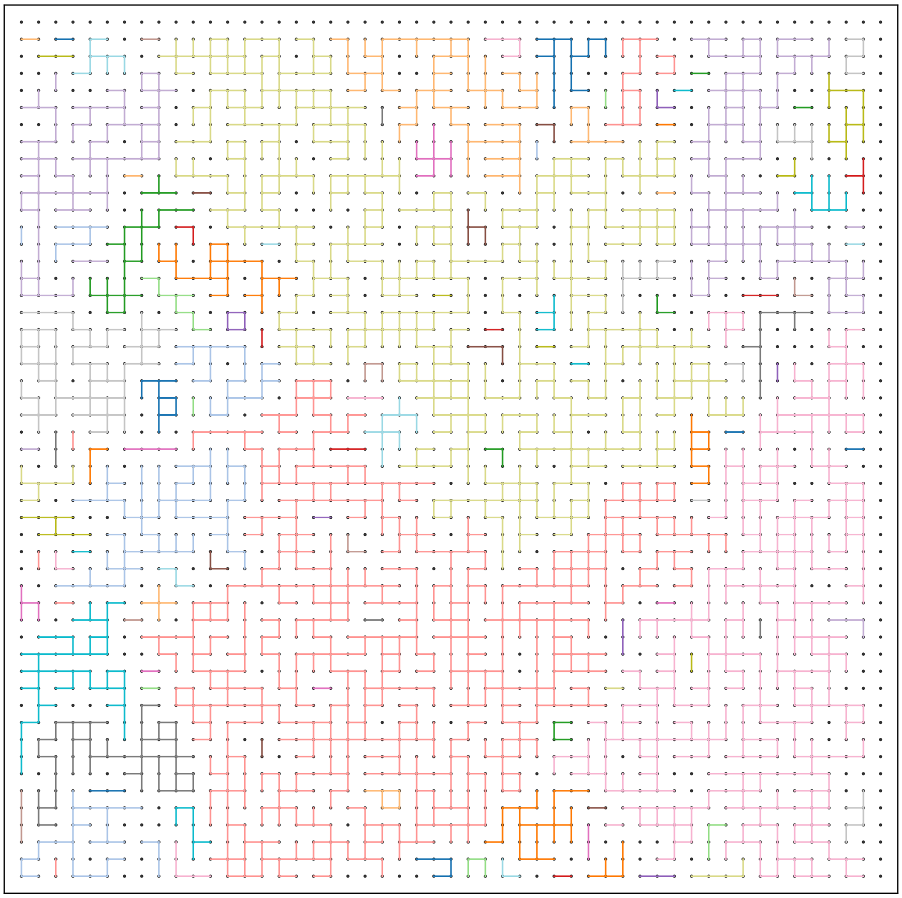

Figure 1: Grid visualization for p = 0.48

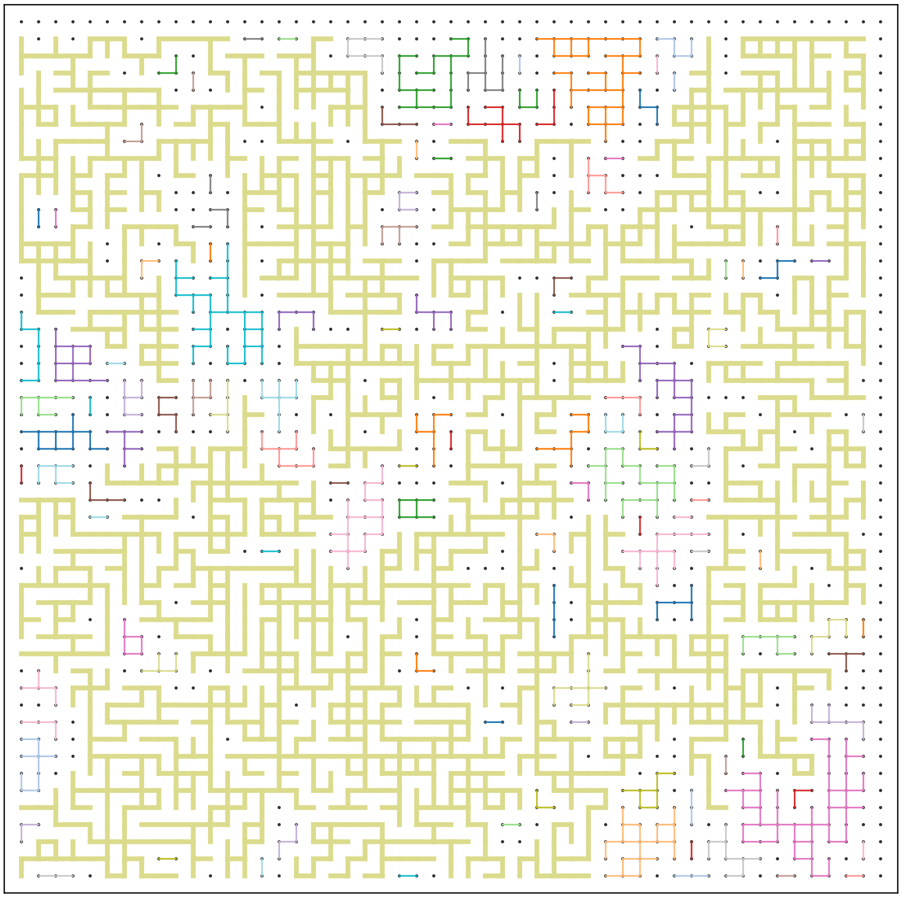

Figure 2: Grid visualization for p = 0.50

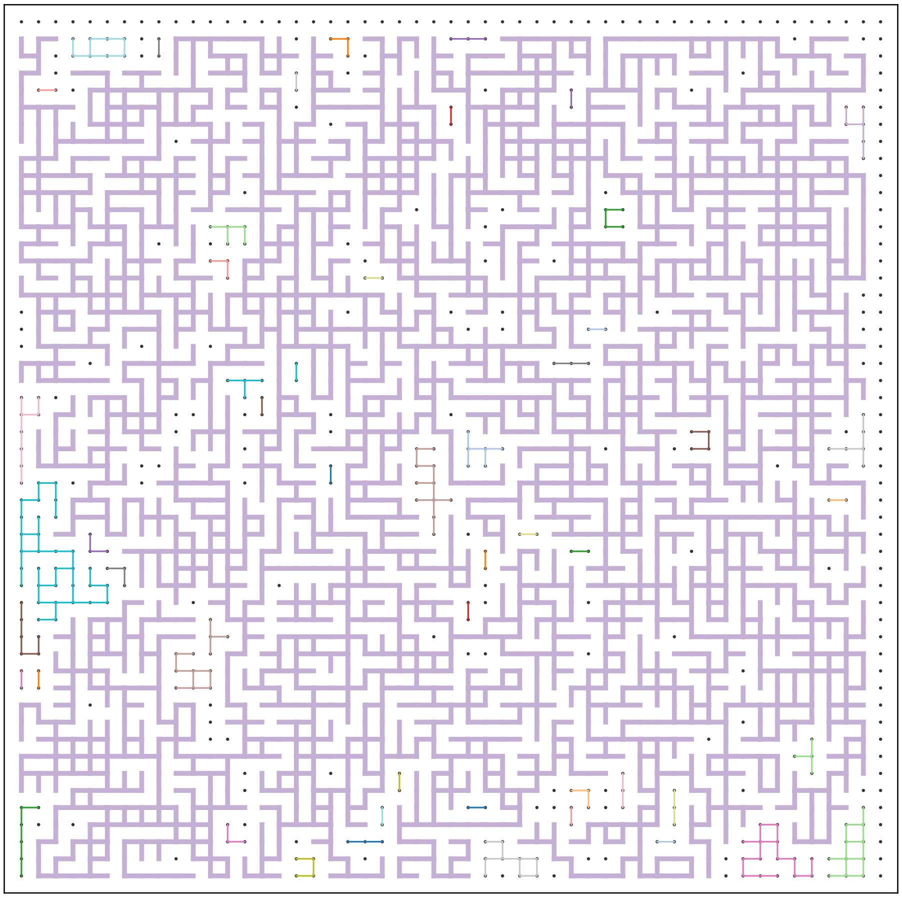

Figure 3: Grid visualization for p = 0.56

4 Additional Simulations, Analysis, and Results

For a more comprehensive understanding of the percolation threshold and the behavior of the system, deeper simulations and analysis have been conducted. These additional simulations cover a wider range of parameters and provide valuable insights into the system's characteristics.

The complete results, visualizations, and detailed analysis of these extended simulations can be found in the accompanying `.html` file. This file contains interactive elements and extensive information on the percolation phenomena studied in this project. We encourage the reader to explore the `.html` file for a more in-depth examination of the percolation threshold and related topics.

5 Conclusion

In this project, I successfully implemented the functions required to generate bonds, find clusters, and check for percolation. By running simulations for various dimensions and p values, I gained a deeper understanding of the percolation threshold and its behavior in different scenarios. The updated range of dimensions and p values allowed me to analyze the system more comprehensively, resulting in a more accurate estimation of the critical percolation probability, p_c. This work contributes to the study of percolation phenomena in complex systems and provides a solid foundation for further exploration of related topics.