AI Assignment 2 (ii)

Ganesh Patchipala 20420682

Introduction

Task 2 focuses on the scalability of the decentralised graph colouring algorithm as the graph size varies. This report aims to understand how increasing the number of nodes affects the algorithm's ability to efficiently find a solution.

Approach

Graph Model and Topology

The Erdős–Rényi model was employed to generate random graphs of different sizes. This model was selected for its widespread use and to maintain consistency with the initial exploration in Task 1.

Key Parameters

Node Sizes: Varied from 50 to 1000 in increments of 50.

Probability of Edge Creation: Maintained at 0.1 to keep the graph sparsity consistent across different sizes.

Colour Selection: A set of 10 colours was chosen to ensure there were enough colours available to provide a feasible solution for larger graphs.

Algorithm Description

The decentralised algorithm from Task 1 was adapted to handle larger graphs. Nodes still autonomously selected their colours to minimise conflicts, iterating until no conflicts remained or a max iterations(100) is reached.

Experiments and Results

Experiment Setup

- The experiment involved generating a series of Erdős–Rényi random graphs for each specified node size.
- The experiments were designed to observe the algorithm's performance across a range of graph sizes, ranging incrementally from 50 to 1000 nodes.
- At each node count, an Erdős–Rényi graph was generated with a fixed probability of 0.1 for edge creation between nodes.
- The number of colours provided for the graph was determined by the square root of the number of nodes, rounded up to the nearest whole number. This approach offers a baseline for the minimum number of colours likely required to achieve a proper colouring, assuming a random distribution of edges. Without this feature, denser graphs would fail to reach a state with no conflicts.

Execution

Each graph underwent the colouring process through the decentralized algorithm, which involved nodes independently selecting colours and iteratively resolving conflicts. For every graph size, the experiment was repeated 10 times to avoid any anomalies caused by the random nature of graph generation and initial colour assignments. The iteration count required for the algorithm to converge to a conflict-free colouring was recorded for each run. The time taken for each graph to reach no conflicts was also recorded.

Results Analysis

The results were plotted to show the relationship between the number of nodes and the average number of iterations and average time taken to reach no conflicts.

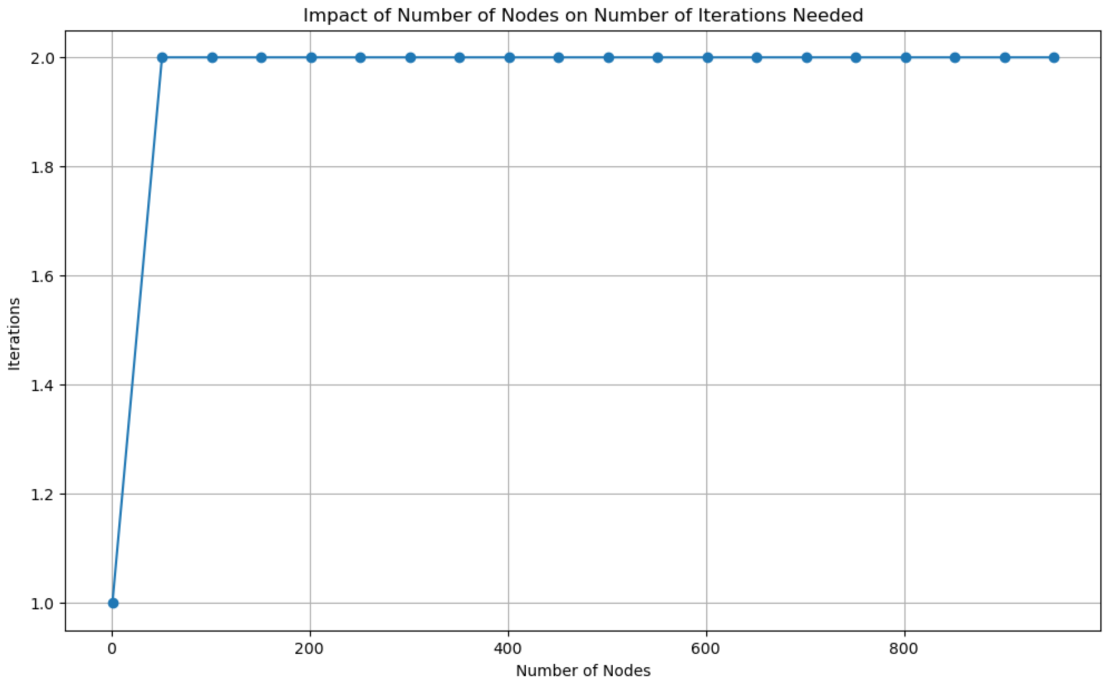

Figure 1: Graph showing how many iterations were needed for each graph to reach no conflicts.

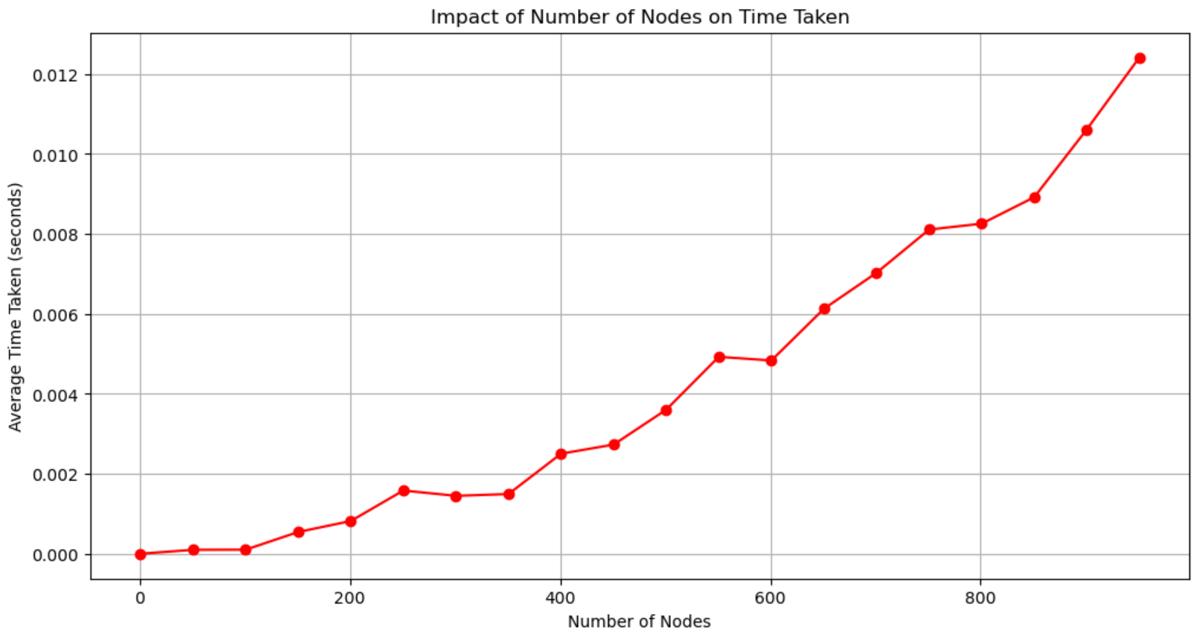

Figure 2: Graph showing how long was needed for each graph to reach no conflicts.

Iterations:

- Despite variations in node count, the iteration counts remained completely constant after the first initial graph, showcasing the algorithm's stable performance.
- This stability suggests that the algorithm's ability to resolve conflicts does not significantly degrade as the graph becomes larger.

Time Taken:

- The time taken increases as the number of nodes in the graph increases, showing a general upward trend, which tells us that with a higher node count, it will generally take more time to process due to increased complexity.
- The rate of increase in time taken is not linear. it shows a gradual climb with some fluctuations.
- There are points where the time taken notably increases or decreases compared to the preceding size, indicating variability in the graph's structure or complexity at different sizes.