Comparison of Sudoku Solving Algorithms

Problem

- There are different types of algorithm for solving Sudoku puzzles all with different characteristics and levels of complexity.
- A comparison of these algorithms is needed to determine the best and most efficient ones.

Solution

A GUI application that allows a user to select a number of Sudoku puzzles to be generated, then select with algorithms are to be run over that set of puzzles. The results of the algorithms being run can be compared in raw data form or through a graph with a line

Technology

The development of the application is being done in Python due to its easier implementation of the algorithms and its extensive data analysis libraries that allow for more effective comparisons.

For the GUI Tkinter is used to allow for an interface that is easy for the user to understand. For data analysis NumPy and Panda packages are used to process the data collected and get summary statistics with Matplotlib being used for data visualisation.

Features

- Graphical User Interface for users
- Implementations of Backtracking and a selection of stochastic algorithms

Aim:

To develop a test bed that allows for the comparison of a selection of main sudoku solving algorithms at a range of difficulty of puzzle.

Objectives:

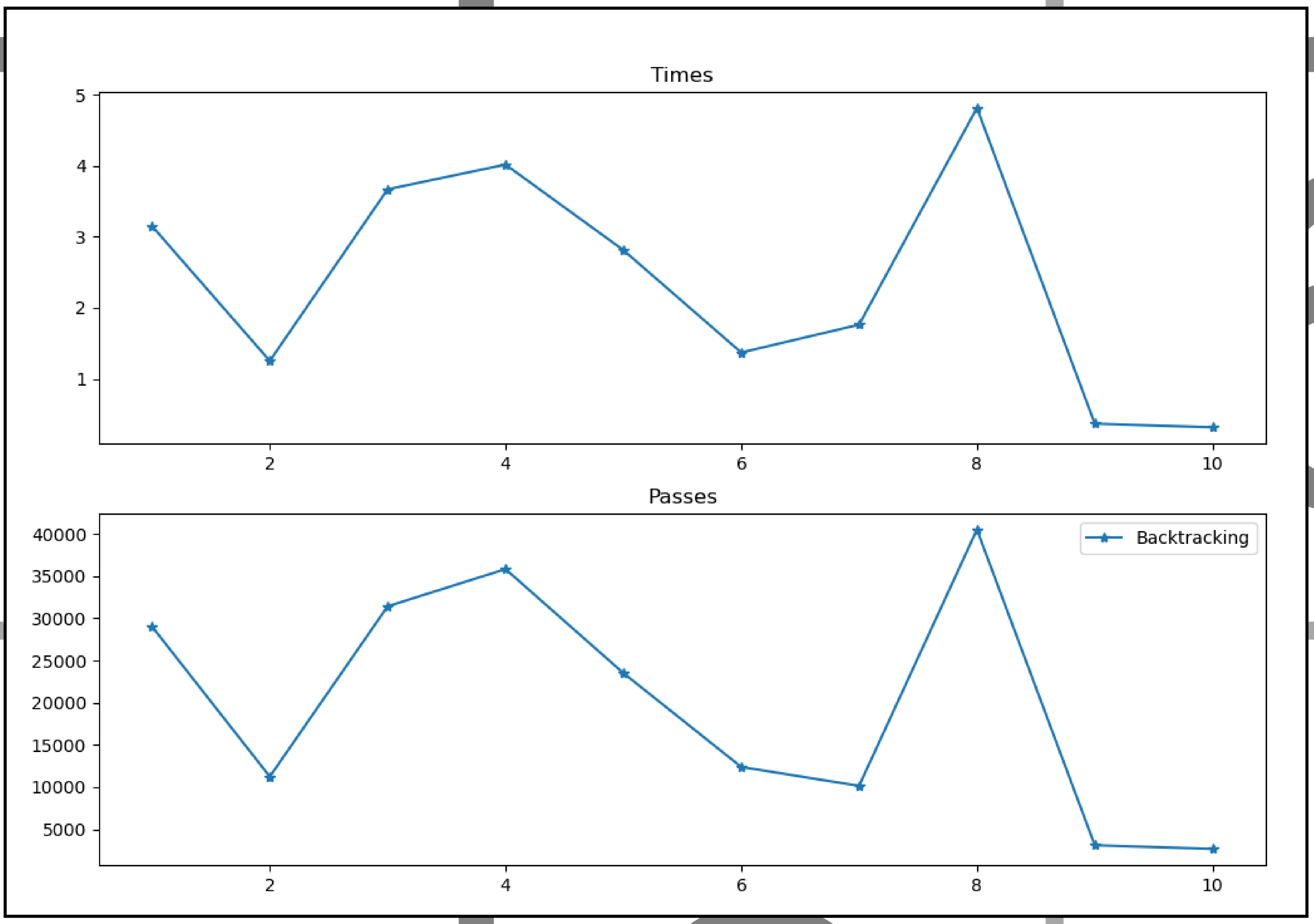
Explore current methods of sudoku solving and select three

Develop test bed to allow comparison of algorithms

Establish test data for comparison of algorithms

Implement algorithms into the test bed

Evaluate implemented sudoku algorithms at multiple complexities



Future Development

- Add further stochastic algorithms to allow for better comparisons
- Add implementations of Constraint programming to compare against stochastic algorithms.
- Add web client to allow users to access the software from the internet
- Add visualisation graphic of algorithms working in real time to help user understand how they work.