SCIF30005: Core Programming, Visualisation and Data Analysis for Scientists

Forest Fire Project - Report

Fuad Jamari (2163440)

1) Model Rules

There are three main input arguments that will be used in running the forest fire simulation in the main function which are Initial probability, p. Number of random grids, M. Size of random grids, N. In this project, we will be using the Von Neumann neighbourhood for the burning process. Below is the example of a random grid of size 20x20 generated by initial probability, p = 0.5 representing an unburnt forest.

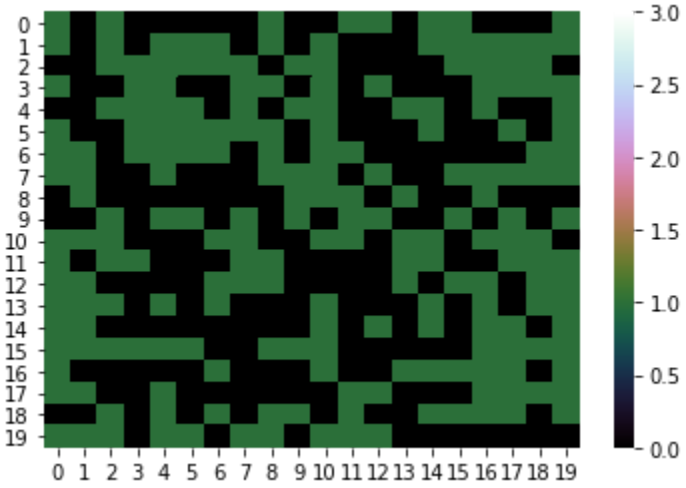

Diagram 1 : Forest generated using p = 0.5, N = 20. Trees (green). No trees (black).

Below is the final result of the burning process after the forest has completely burnt (where there are no other trees that can be burnt using Von Neumann neighbourhood rule). Throughout this project, we will be using this model to study the model convergence, adding of wind factor and analysing the performance of parallelising the tasks in simulating the models.

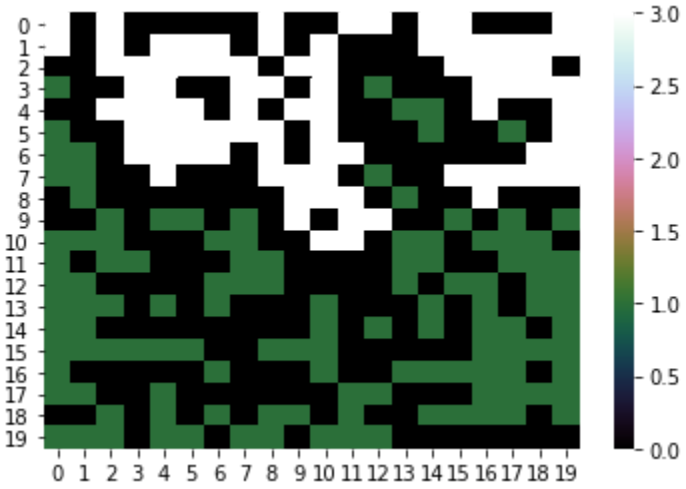

Diagram 2 : Forest after completely burnt. Burnt trees (white).

Functions to work out the Forest Fire Model

For the purpose of running the Forest Fire model and recording data, there are some functions that are built to make the process much more simpler. Below are the list of the functions used and their purposes:

I. **generate_forest** : To generate random trees at a given seed with initial probability p.

II. **print_forest :** To print the current state of forest.

III. **starts_to_burn :** To start the burning process of forest fire.

IV. **check_false_wind :** To check whether the input string is a false representation of wind directions.

V. **parallel_burning_wind :** To run the forest fire iteratively with consideration of wind direction.

VI. **check_forest :** To check whether two burning forest states are different.

VII. **all_burn_wind :** To run forest fire in one random 'forest' grid and record the running time and total steps of forest fire.

VIII. **return_seed :** To create random seeds for M random grids simulation.

IX. **fire_bottom :** To estimate the probability of fire to reach the bottom of the forest grid.

X. **M_model_wind :** To run forest fire simulation on M randomly generated forest grids.

XI. **convergence_data_wind :** To record the runtime and average steps of forest fire simulation on M randomly generated forest grids of size N x N at all initial probabilities, p = {0, 0.05 …, 1}.

XII. **write_file :** To write the collected data from simulation into a text file.

XIII. **write_forest :** To write the current state of forest grid into a text file.

Parallelisation : In this project, parallelisation is done within the main for loops in the working functions (written in blue colour) that run the forest fire simulations. This is done in order to separate printing tasks from the simulations so all statements can be printed out properly outside the parallelised functions. Besides, the parallelisation is implemented using OpenMP with static scheduling. The reason for using static scheduling is due to model convergence at large enough random forest grids, M. Throughout this project, only omp_set_num_threads are used in the main function for the purpose of setting the number of threads to work on the parallelise functions. The reason for not parallelising the main loop of convergence_data_wind is to allow the collection of average steps and runtime to be according to the chronological order of probabilities, (i.e p = {0, 0.05, 0.1, …, 1}). However, convergence_data_wind still uses other parallelised functions such as M_model_wind to work on the data collection.

2) Model Convergence

I. Number of random grids, M - In order to study the convergence of values in the average steps taken to finish forest fire depending on the number of random grids, M we create five different sets of simulations with different numbers of random grids. In this case the grid size are fixed to 100 x 100 or N = 100.

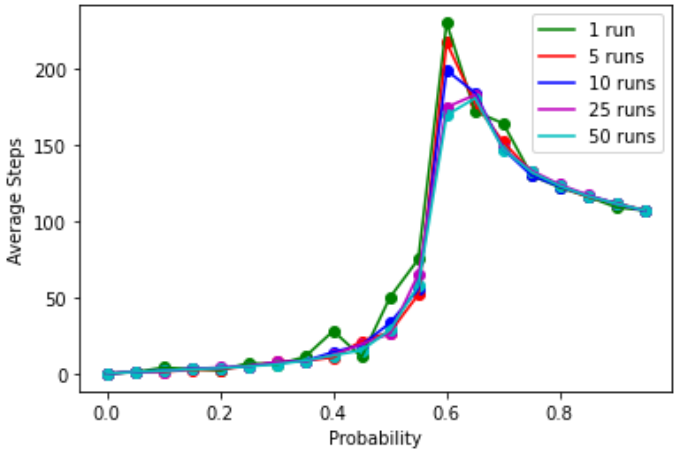

Diagram 3 : Average steps taken to completely burnt the forest of grid size N = 100 generated by random seed with different number of random grids, M

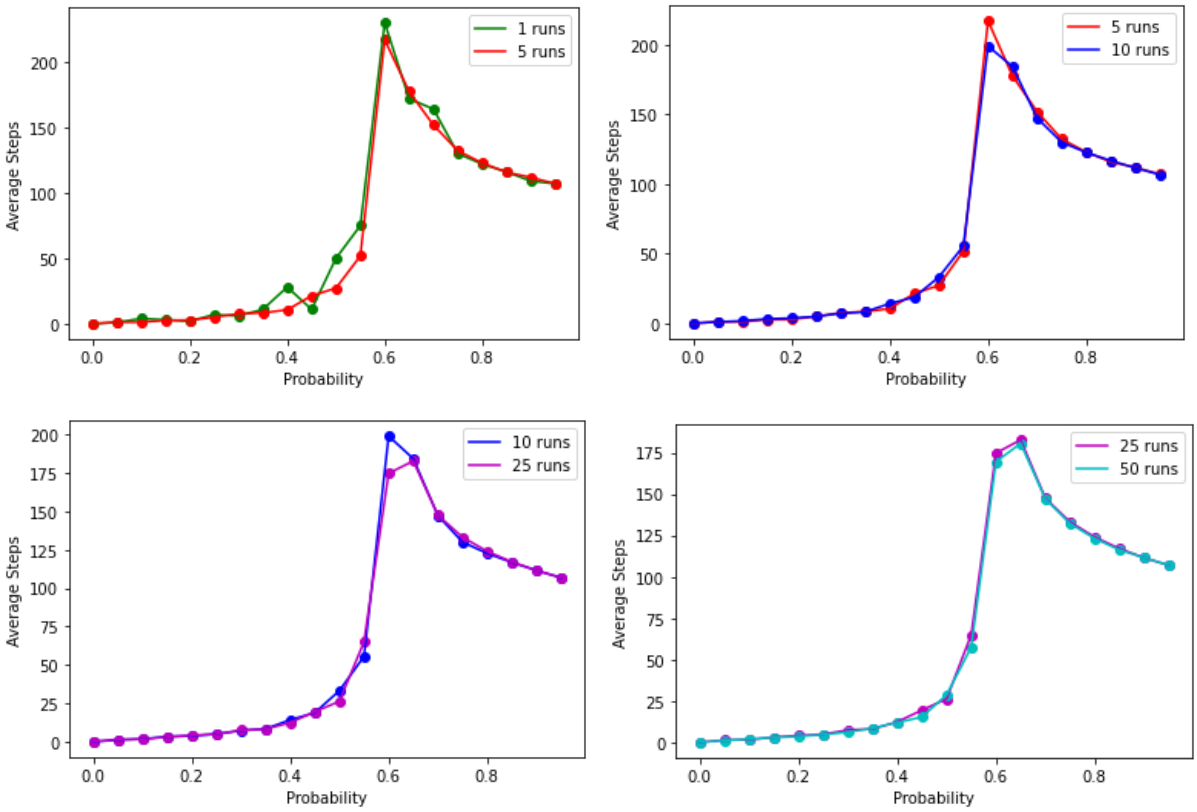

Diagram 4 : Average steps taken to completely burn the forests using 1 random grids (green), 5 random grids (red), 10 random grids (blue), 25 random grids (magenta) and 50 random grids (cyan).

II. Size of random grids, N - In order to study the convergence of values in the average steps taken to finish forest fire depending on the size of random grids, N we create five different sets of simulations with different sizes of random grids. In this case the number of random grids are fixed to M = 50.

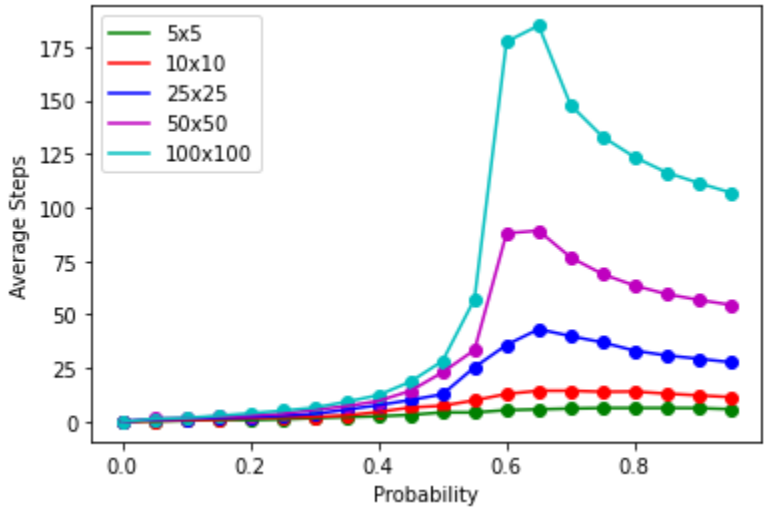

Diagram 4 : Average steps taken to completely burn the forests using 5x5 random grids (green), 10x10 random grids (red), 25x25 random grids (blue), 50x50 random grids (magenta) and 100x100 random grids (cyan).

Discussions - Based on the results obtained through simulations, we can observe that the average steps of forest fire converges when the number of random grids, M increases. In which case the trends for 25 random grids seem similar to the trends for 50 random grids. However, it is not the case with the size of random grids, N since the average steps increases as the size of random grids increases. This is quite intuitive since at every given probability, it will take much longer to finish the burning processes. Even so, the overall trends of average steps seems quite similar between different grid sizes which is due to the convergence effect of the number of random grids which is kept constant to M = 50. Since we know that the model converges in number of steps as the number of random grids M increases, it supports the decision to use static scheduling in the parallelisation of the functions. Hence, we can assume that for a large enough M, each thread will complete an approximately equal amount of tasks to run M number of Forest Fire simulations.

3) Probability of reaching bottom grid

Besides studying the effects of the preset arguments (size of random grids and the number of random grids) on the average steps of forest, it is also interesting to know whether for which values of probabilities, p can we expect the forest fire to reach the bottom most part of the grid. Using the simulations from the model convergence, we record the probability of fire to reach the bottom grid by averaging the observations in the simulations that implement different numbers of 100x100 random grids, M. Also, it is interesting to know whether the size of random grids, N influenced the possibility of forest fire reaching the bottom grid by studying the trends of the simulations that implement different sizes of 50 random grids.

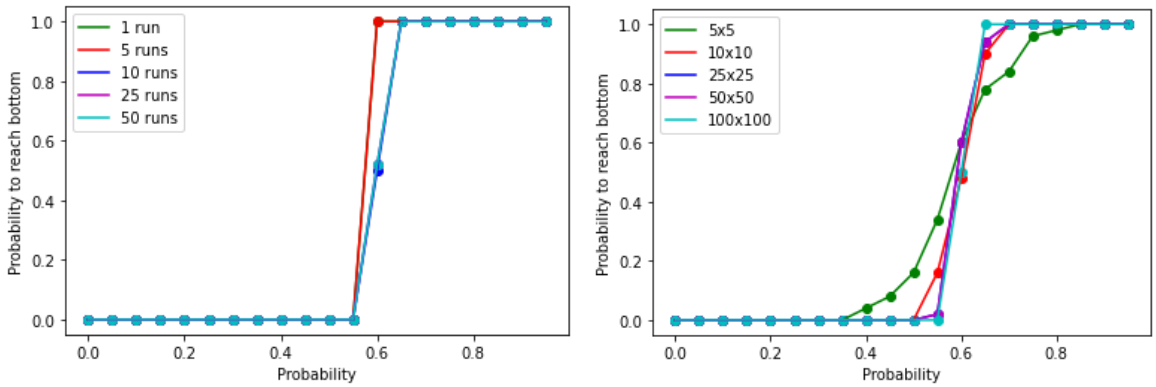

Diagram 5 : Graphs of probability of fire to reach bottom grid against the values of preset probabilities, p. Convergence effect due number of random grids, M (left). Difference in trends due size of random grids (right).

Discussions - In the left case, although results may differ in each run of forest fire, it is expected for the forest fire to reach the bottom grid with certain (or with the probability of 1.0) whenever the input probabilities, p are set to values of at least 0.65 considering that convergence of model taking place. On the other hand, the right diagram suggests that for an increasing size of random grid we will be more certain of the possibility of forest fire to reach the bottom grid. This can be observed through the less fluctuating trend on the larger random grid size. In short, for a large enough random size grid, N and considering the model converges at the chosen number of random grid, M we expect the forest fire to reach the bottom most part of the grid when the preset probabilities are $p \geq 0.65$.

4) <u>Adding Wind</u>

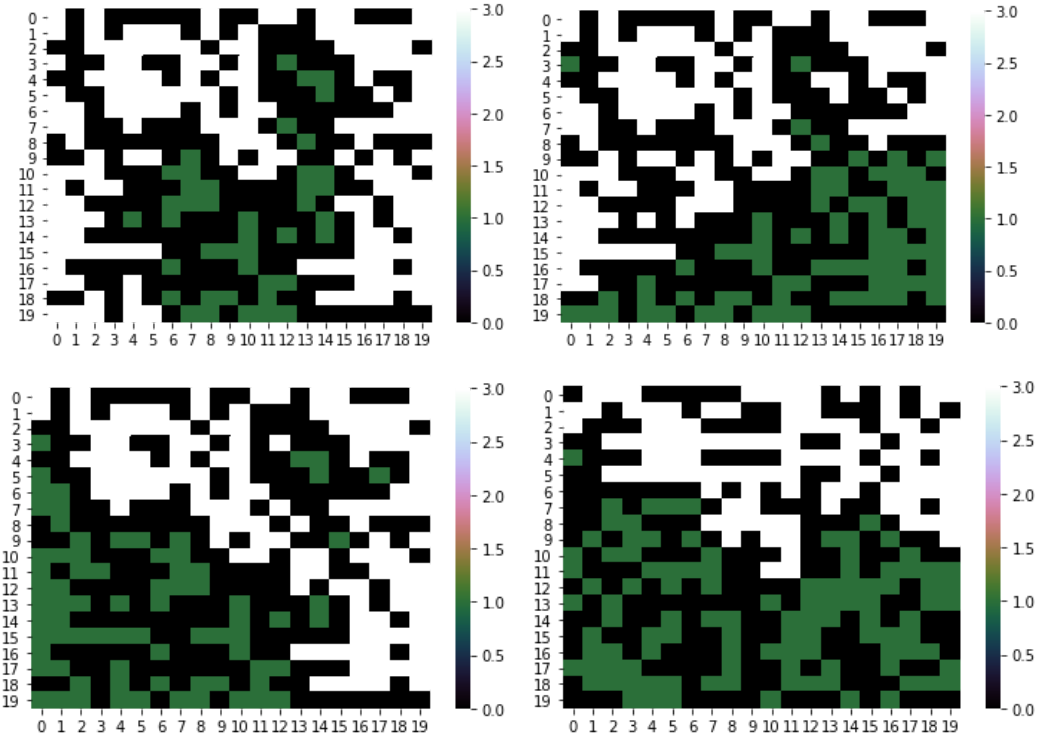

Diagram 6 : Forest fire with downward wind (top left). Forest fire with leftward wind (top right). Forest fire with rightward wind (bottom left). Forest fire with upward wind (bottom right).

By comparing Diagram 2 with Diagram 6, we can deduce that adding wind in different directions will affect the number of trees burnt by the forest fire. Hence, we need to study the effects of the wind towards the average steps taken to complete the forest fire and probability to reach bottom grids.

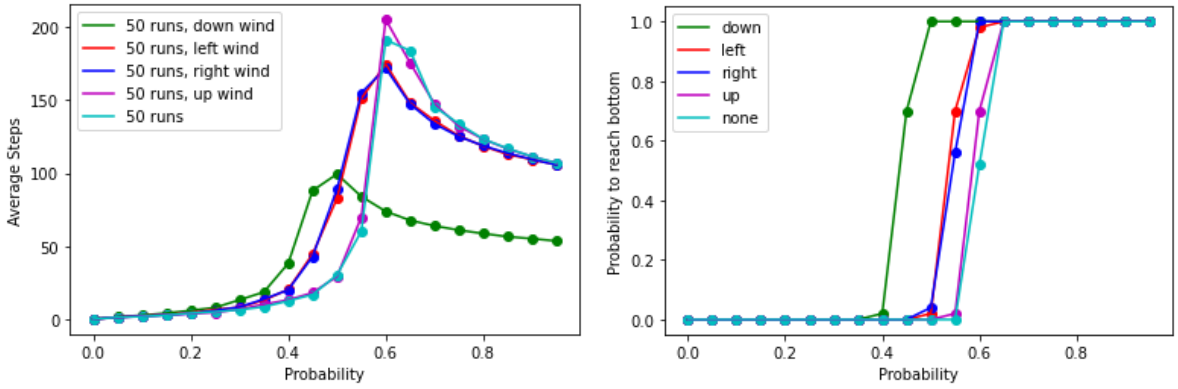

Diagram 7 : Wind direction effects towards the average steps (left) and probability to reach bottom grid (right). Downward wind (green). Leftward wind (red). Rightward wind (blue). Upward wind (magenta). No wind (cyan).

Discussions - In relation to the normal trend of no wind (which is similar to the one in Diagram 4), only the downward wind can reduce the average steps of forest fire at every initial probability, p and reaches the bottom grid at the lowest probability. Also, in the presence of downward wind, the maximum reading of average steps is at a lower initial probability than no wind. Besides, both of leftward and rightward winds result in quite similar trends as the no wind simulation apart from both of them has the maximum average steps at smaller probability compared to the latter and having the same initial probability to reach the bottom grid with certainty which is lower than the initial probability of no wind. Upward wind however, has the highest maximum average steps of forest fire in comparison to other wind directions. But, it still has the same initial probability as no wind to reach the bottom grid with certainty.

5) <u>Performance Analysis</u>

The purpose of this analysis is to evaluate the efficiency of multithreading in decreasing the running time of the forest fire model. To analyse the performance of the Openmp implemented in the model, we look at the walltime graphs and the speedup graph constructed by the data of the simulated forest fire with initial probability p = 0.6 and number of random grid M = 50 of three different sizes which are N = 50, 100, 500.

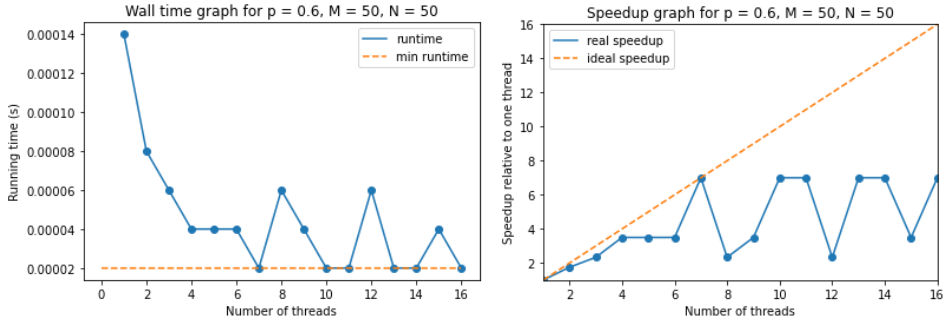

Diagram 8 : Wall time graph of test using 50x50 random grid size (left). Speedup graph of test using 50x50 random grid size (right).

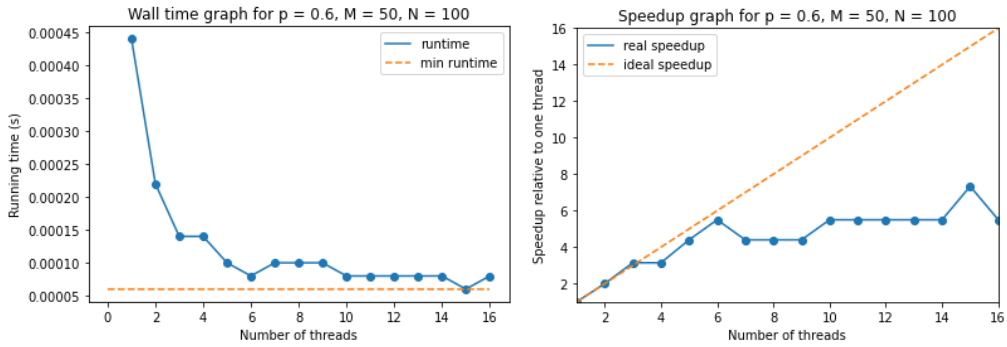

Diagram 9 : Wall time graph of test using 100x100 random grid size (left).
Speedup graph of test using 100x100 random grid size (right).

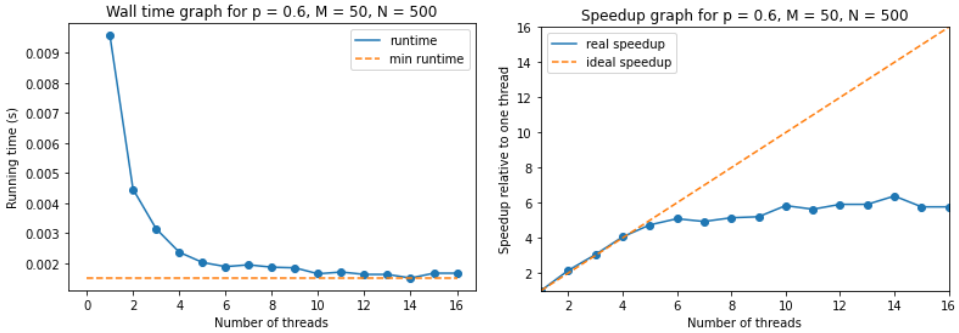

Diagram 10 : Wall time graph of test using 500x500 random grid size (left).
Speedup graph of test using 500x500 random grid size (right).

Discussions - i) Based on the Diagram 8, we observe that both the walltime graph and the speedup graph are not smooth. Besides, the minimum running time can be obtained by running the simulation with a different number of threads which are 7, 10, 11, 13, and 14. Also, referring to the speedup we can see that we can possibly achieve the ideal speedup when we use 7 threads for simulation of forest fire with N = 50. **ii)** Based on Diagram 9, we observe a much smoother walltime graph and speedup graph with 15 threads having the lowest running time. However, differ from the speedup graph in Diagram 8, the speedup gain in using 14 threads is not ideal since the ideal choice for the number of threads are 2 and 3 while 15 threads still has the highest speedup. Hence, considering both walltime and speedup we can choose 15 threads to obtain maximum performance for N = 100. **iii)** Based on Diagram 10, we observe the most smoothest walltime graph and speedup graph. In the walltime graph, we can observe how the running time converges to the minimum value as the number of threads increases. Therefore we can refer to the speedup graph for which we can nearly obtain the ideal speedup when using 1 to 5 threads while 14 threads recorded the highest speedup. Hence, by that we can decide to multithread using 5 threads to get the ideal speedup or use 14 threads to obtain maximum performance for N = 500.

Summary : Based on the wall time and speedup due to multithread, it is observable that the performance can significantly be improved by using more than one thread. Also, the fluctuation in performance trend can be reduced by the increasing size of random forest grids which are associated with an increasing amount of tasks for each thread. Hence, due to the unpredictable fluctuations for the subsequence performances when using more than six threads for N < 500, it seems pretty reasonable to choose 6 threads as the maximum number of threads to be used in small random forest grids. Besides, the implementation of static scheduling aligns with the model convergence due to the stability of the Forest Fire average steps for M more than 25.