Ben Steves

CS 374

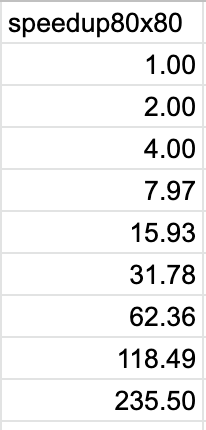
Prof Adams

9-22-21

Analysis - Monte Carlo Trees

As one would expect, as the fixed probabilities increase, burn percentages increase alongside - though this increase is not linear. When the program is given a smaller probability, it is a little harder to actually burn many trees since each new tree set on fire by its neighbor is given the same probability of igniting the next tree. The probability that the tree’s begin to accelerate on a 45 degree tangent differs with the size of the forest, if only slightly. The 20x20 forest is at its 45 degree tangent at about 0.43, and both 40x40 and 80x80 have steeper accelerations around 0.46 and 0.47, respectively. Decelerations all seem to be the same for all size forests, close to about 0.61 or 0.62. These are just ballpark estimates obtained by the graphs I had made. Inflection points are all at about 0.52.

The number of iterations peaks near burn probabilities of 0.55-0.57 (20x20), 0.53-0.55 (40x40), and 0.52-0.53 (80x80). As the burn time increases more rapidly, more trees in the forest burn, which increases the number of iterations. As the burns start nearing 1, they approach 1 very slowly, and with that the number of iterations decreases from its peaks.

As the number of processes increases, the runtimes halve - so on the log\_2 scale, we see a linear decrease between runtime and the number of processes (of course, this isn’t linear on the normal scale, but appears so because of the log transformation). The speedups for each number of processes are shown below, going down from 1-256 processes. 

The speedups here are only useful for the parallel part of the program, however, and the entire program is not embarrassingly parallel - it is part parallel part sequential. Though I don’t know the exact percentage of the program that is parallel and sequential, Amdahl’s law applies to this program and there would be a limit in speedup, because the sequential part of the program would hinder any speedup at a large number of processes. Gustafson’s law proves true to this program as well, as speedup does increase as more trees are added to the forest. This is visible in the above tables, though again, this is only for the parallel part of the program.

The fire’s burn times increase as the number of trees in the forest increases, which makes sense because the larger the forest gets, the more trees that could be burned with some probability. The number of processes does not have any effect on iterations. For example, as seen in the graphs, there is hardly any difference for the number of iterations for np=1 and np=256, though these iterations do differ by size.

Burn probabilities are pretty much the same no matter the size of the forest, as visible from the graphs. There is an acceleration and deceleration in burn percentages near the same probabilities for all three sizes, giving very similar looking graphs.