Most of the work is done by iterating through the Sudoku and use constraining information to eliminate possibilities. Backtracking only comes into play in hard problems. At first, I only implement minimum remaining values heuristic for backtracking algorithm because it is enough. Almost for all cases, the result appears without delay.

Then I implemented degree and least-constraining value heuristics. They do not really seem to improve the performance because there is so little space for improvement and calculating the heuristic itself requires time.

There is basically no difference in performance difference. For all cases, the solution appears within 1 second. I can only see a slight delay in sudoku\_hardest.txt.

My iterative method is very powerful, so backtracking is used to the minimum extents. For each digit, I check all rows, columns and squares and find out all the possible positions for this digit. There are many times those positions are within one row, column or square. In such case, other positions within that region cannot contain this digit. With this method, computer can use simple logic to eliminate search space to minimum or empty before backtracking search.

Fred Zhang