

Sudoku Report Template

1. Background for Analysis

Recall that Monster Sudoku puzzles are described by four parameters:

- N = the length of one side of the $N \times N$ grid, also the number of distinct tokens
- P = the number of rows in each block, also the number of block columns.
- Q = the number of columns in each block, also the number of block rows.
- M = the number of values filled in from the start.

Note: Norvig uses "box" where we use "block" -- the terms are equivalent.

[Use the Tournament time-out value of 600 seconds (10 minutes) per board below.]

[Do not count boards on which you time out.]

1.a. Estimate the Critical Value of "hardest M "

Obviously, if $M = 0$ then you can find a solution very quickly, while if $M = N^2$ then you can fail very quickly. Somewhere in between those extremes there is a "hardest" value of M , i.e., a value of M for which your program takes, on average, the longest time to succeed or fail. For standard Sudoku, $P = Q = 3$. For Monster Sudoku, choose P and Q to be the two factors of N closest to \sqrt{N} . (If you chose $P = 1$ and $Q = N$, you would have a very easy problem.)

Note that you need to consider only a few values of M that are close to zero or close to N^2 . You only need to bracket loosely the "hardest M ," then do local sampling within the bracketing interval to refine your estimate. For standard Sudoku, you can get an initial estimate of M by counting the number of filled cells in any published 9×9 puzzle. For Monster Sudoku, you can estimate a new value of M by

$$M_{\text{new}} = M_{\text{old}} * ([N_{\text{new}} / N_{\text{old}}]^2).$$

Thereafter, sample various values of M in the vicinity of your initial estimate. Generate and solve 10 or more random problems for each such sampled value of M , until you are confident you have bracketed closely the "hardest M ." Sample within the bracketing interval to estimate which value of M produces the longest average time. Then do additional time trials in the vicinity of that value of M in order to get a more accurate value. Use your timing data to produce the graphs required in your report below.

1.b. Estimate the Value of "Half-Solvable M ," where $P(\text{solvable}) = 0.5$

Obviously, if $M = 0$ then the puzzle is solvable, so $M = 0$ implies $P(\text{solvable}) = 1.0$. As M increases, at some point $P(\text{solvable})$ will begin to decrease. When $M = N^2$, $P(\text{solvable})$ will be very small. At some value of M between $M = 0$ and $M = N^2$, exactly half the boards will be solvable and half will not be. Use the methods in the subsection above to estimate the value of M for which $P(\text{solvable}) = 0.5$ ("half-solvable M "). Is that value approximately the same as the value you get for "hardest M ?"

2. Report Template (What to Turn In)

[Anything in boldface below is required; anything below in angle brackets <...> is a parameter; and anything below in square brackets [...] is an instruction. The safest file format for document transmission is always PDF (at least, currently). Usually, students will write the report in Word or some equivalent text processor, and then convert the final version to PDF for submission.]

[Turn in only ONE REPORT PER TEAM --- not per person.]
[It is OK to use more than one page, if needed]

My name, ID#, UCInetID: <Mary Roe, 999999999, mroe@uci.edu>

Partner name, ID#, UCInetID (or “none”): <John Doe, 888888888, jdoe@uci.edu> or “none”

The programming language(s) you used in your project: <C++/Java/Python>

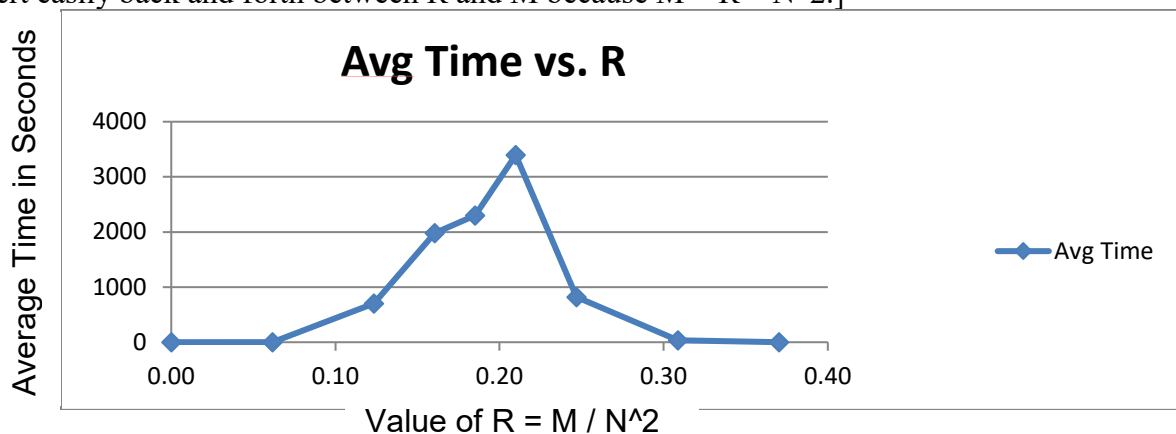
What, if any, additional heuristics did you use in your Tournament Final AI?

By turning in this assignment, I/We do affirm that we did not copy any code, text, or data except course material provided to us by the textbook, class website, class gitlab, or teaching staff.

A. The Critical Value of “Hardest M” for $N = 9$

A.1. Find the critical value of “hardest M” for $N = 9$ and BT.

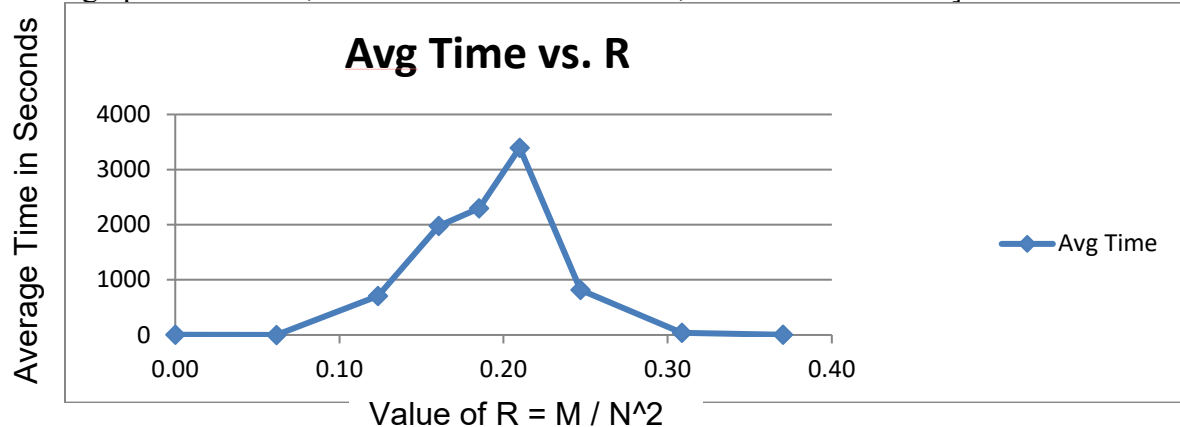
[Produce a graph for BT, similar to that shown below, where $R = M / N^2$. Note that you can convert easily back and forth between R and M because $M = R * N^2$.]



[For you, the x-axis will be $R = M / N^2$, and the y-axis will be the average time that BT took to solve each randomly generated standard Sudoku puzzle or to prove that it was unsolvable.]

A.2. Find the critical value of “hardest M” for $N = 9$ and BT with FC (BT+FC).

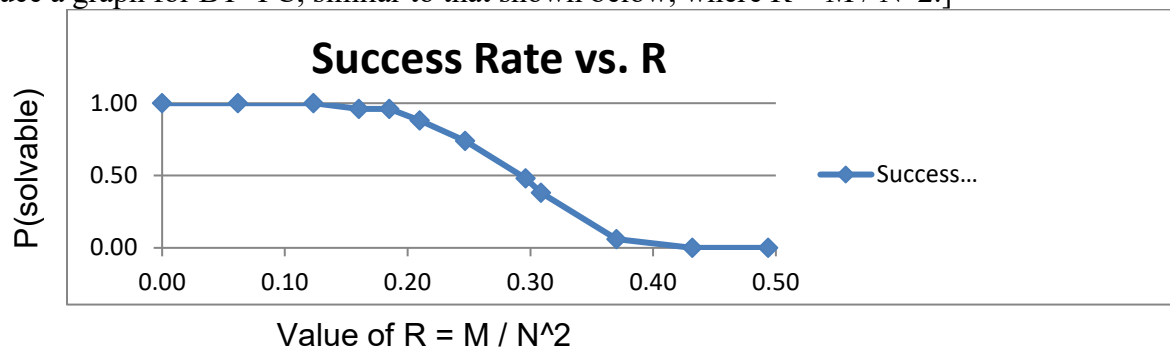
[Produce a graph for BT+FC, similar to that shown below, where $R = M / N^2$.]



[For you, the x-axis will be $R = M / N^2$, and the y-axis will be the average time that BT+FC took to solve each randomly generated standard Sudoku puzzle or prove that it was unsolvable.]

A.3. How does puzzle solvability for BT+FC vary with $R = M / N^2$?

[Produce a graph for BT+FC, similar to that shown below, where $R = M / N^2$.]



[For you, the x-axis will be $R = M / N^2$, and the y-axis will be the probability that a randomly generated standard Sudoku puzzle is solvable using BT+FC.]

A.4. Answer these questions (for $N=9$):

A.4.a. What critical value of “hardest M” do you get for BT (in A.1)? for BT+FC (in A.2)? Are these values approximately the same?

A.4.b. For “hardest M,” what is the mean and standard deviation of the total time for BT? for BT+FC? Are these values approximately the same?

A.4.c. For what value of M does $P(\text{solvable}) \approx 0.5$ (“half-solvable M”) occur for BT+FC? Is this value approximately the same as your value of “hardest M” for BT+FC (in A.4.a)?

B. Combinations of Methods for N = 9

B.1. Find mean (sdev) total time (in seconds), “hardest M,” and “half-solvable M” for N = 9 and the method combinations specified below.

[Produce a table similar to that shown below. Mean (Sdev) Total Time must be computed at the “hardest M” for that particular combination. Show the standard deviation (sdev) for each mean.]

Mean (Sdev) Total Time	Hardest M	Half-Solvable M	Combination of Methods
			BT [Basic backtracking search alone.]
			BT+FC [Add Forward Checking (FC) to BT]
			BT+FC+MRV [Add Minimum Remaining Values (MRV) to BT+FC]
			BT+FC+MRV+LCV [Add Least Constraining Value (LCV) to BT+FC+MRV]
			BT+FC+MAD+LCV [Add Degree Heuristic (DH) to BT+FC+MRV+LCV]
			BT+FC+MAD+LCV+NOR [Add Norvig Heuristic (NOR) to BT+FC+MAD+LCV]
			Your Tournament Final AI [In all its glory!]

B.2. Answer these questions (for N=9):

B.2.a. Which more sophisticated methods let you solve N = 9 for “hardest M” puzzles faster than BT+FC?

B.2.b. Which more sophisticated methods have overhead costs that outweigh their potential benefit? (I.e., they make your solver run slower than BT+FC.)

B.2.c. Do you get (approximately, i.e., within sampling error) the same values of “hardest M” and “half-solvable M” for the different combinations that you tested?

C. “Monster” Sudoku for $N > 9$

[For “Largest Solvable N” cite only Ns where “hardest/half-solvable M” are computable.]

C.1. Find mean times, “hardest M,” and “half-solvable M” for $N > 9$.

[Produce a table similar to that below. For each method combination, Largest Solvable N means the largest N for which “hardest M” and “half-solvable M” are computable within the time limit. Mean and standard deviation (sdev) are computed for the “hardest M” and “largest solvable N.”]

Largest Solvable N	Mean (Sdev) Time	Hardest M	Half-Solvable M	Combination of Methods
				BT [Basic backtracking search alone.]
				BT+FC [Add Forward Checking (FC) to BT]
				BT+FC+MRV [Add Minimum Remaining Values (MRV) to BT+FC]
				BT+FC+MRV+LCV [Add Least Constraining Value (LCV) to BT+FC+MRV]
				BT+FC+MAD+LCV [Add Degree Heuristic (DH) to BT+FC+MRV+LCV]
				BT+FC+MAD+LCV+NOR [Add Norvig Heuristic (NOR) to BT+FC+MAD+LCV]
				Your Tournament Final AI [In all its glory!]

C.2. Answer these questions:

C.2.a. Which more sophisticated methods let you to reach larger values of N for “hardest M” puzzles? Does the answer to this question change as N increases?

C.2.b. Which more sophisticated methods have overhead costs that outweigh their benefit? (I.e., they make it slower.) Does the answer to this question change as N increases?

C.2.c. Do you get (approximately) the same value of “hardest M” for different N and different method combinations? for “half-solvable M?”