Sudoku Final AI Report

Team number: ***34***
Member #1 (name/id) ***Colin Yee / 58312560***
Member #2 (name/id) ***Ervin Grimaldi / 51767760***

I. Minimal AI

I.A. Briefly describe your Minimal AI algorithm. What did you do that was fun, clever, or creative?

*Our **Minimal AI** algorithm implements **forward checking** into the solver to reduce the number of invalid board solutions from the current board. When we first submitted our Minimal AI, the algorithm would initially grab the last assigned variable from the trail directly and propagate constraints from there. We wouldn't need to loop through the variables to find recently modified ones. However, in our Final AI, we switched to looping because we wanted our forward checking to be consistent with Norvig check.*

I.B Describe your Minimal AI algorithm's performance:

E.g. Generate around 60 boards of different difficulties and run your Minimal AI algorithm. Then provide a few words and a table like the following:

Board Size	Sample Size (n)	Boards Solved	Average # of backtracks
9x9 (easy)	15	15	220
12x12 (intermediate)	15	15	9086.93
16x16 (hard)	15	11	48434.64
25x25 (Expert)	15	0	--
Total Summary	60	41	19247.19

II. Final AI

II.A. How did integrating advanced techniques (LCV, MRV, MAD, or NOR) into the Final AI change its solving strategy compared to Minimal AI?

Compared to randomly selecting values and variables in Minimal AI, Final AI's advanced techniques select values and variables in a way to reduce the time it takes to find a board solution.

- *For **MRV** and **MAD**, the solver selects a variable that would have the most likely chance to fail to reduce the number of backtracks the solver will take.*
- *For **LCV**, the solver selects the value that rules out the fewest choices for neighboring variables to maximize the number of valid possible solutions in a track.*
- *For **NOR**, the solver propagates constraints to neighbors and effectively looks for inconsistencies to backtrack as early as possible (which reduces the number of backtracks in total).*

II.B. Which of the advanced heuristics (LCV, MRV, MAD, or NOR) had the most significant impact on the performance, and why do you think that was?

NOR *(Norvig check) easily has the most significant impact on performance.* ***NOR*** *is the only heuristic from this selection that removes values from variable's domains, identifies invalid board permutations, and signals to the solver when to backtrack. This significantly reduces the number of backtracks and the time it takes to find a valid board solution.*

II.C Describe your Final AI algorithm's performance:

The Final AI's average number of backtracks and average time to find a board solution is significantly smaller than Minimal AI. Additionally, the Final AI is able to consistently solve hard and expert boards without timing out.

E.g. Use the same generated 60 boards from earlier and run your Final AI algorithm. Compare your results with Minimal AI performance, then provide a few words and a table like the following:

Board Size	Sample Size (n)	Boards Solved	Average # of backtracks
9x9 (easy)	15	15	0
12x12 (intermediate)	15	15	0.133
16x16 (hard)	15	15	2.933
25x25 (Expert)	15	15	16.267
Total Summary	60	60	4.833

III. Has this project altered your interest or perspective towards artificial intelligence? If so, how?

I believe this project broadens our view about AI. We saw that AI techniques can be utilized for a broad number of complex problems. It's all about collecting and utilizing information.

III. In about 1/4 page of text or less, provide suggestions for improving this project (*this section does <u>NOT</u> count as past of your two-page total limit.*)

Colin*:*

1. *I believe the project would be improved if we had access to a tutorial video that went through all the provided classes and functions in the Sudoku Solver and how they interact with each other.*

Ervin:

1) *I wished the starter code was more prone to "structural" modifications, allowing major changes on how each component works, thus allowing different type of algorithm solutions such as DLX. I would also like to see more information on how each component interacted for the starter code.*