

Mini-Project
CS 171 Intro to AI
Winter 2017

Due at EEE class dropbox: Friday March 17, 11:59PM

Work in pairs. (If you can't find a partner, email the TA by February 28 and one will be randomly assigned to you.) Program in Python or Java.

Option 1 (All students)

Using the provided Python or Java template software (btsolver.py and supporting files in Python; likewise for Java) available via the EEE class website mini-project page, *and no other external source of software*, write and test a Sudoku problem solver. Code up the following methods and heuristics (as defined in R&N Chapter 6 and in lectures) and slot them into the template provided: **Arc Consistency, MRV, Degree, LCV, Forward Checking, Naked Pairs, and Naked Triples.**

For each combination of methods and for each problem instance you try, record the *number of backtracks* your solver does for each problem, the total running time divided by the number of backtracks, and whether the problem is solved successfully or not. For each combination of methods you try, and for each difficulty level in the problem database provided in the Python codebase, report the solution capability (success rate, and average number of backtracks) by testing on the difficulty-stratified problem database. Use problems PE1 - PE5 (5 easy problems), PM1 - PM5 (5 medium-difficulty problems), and PH1 - PH5 (5 hard problems) to test each combination of methods specified as follows. Which combinations of methods to try: (a) Try each method separately, or with the fewest other methods possible; (b) Find a "best" or "most productive" combination of methods as well as you can; explain in what sense it is best and what your evidence for that "best" conclusion is. (You may want to make a plot of some numerical results to provide the evidence.)

Option 2 (Extra credit)

First do Option 1. Then ...

Following a framework provided, build or improve a tournament framework in Python for computer agents (written in Python) that play the board game of "Go". Alternatively or additionally, build or improve such a computer agent; for example, start from the provided Monte Carlo Tree Search go-playing agent.

Questions to:

Piazza site; Discussion section on Wednesday; TA (Junkyu Lee).