Brian Gauch

CS 360

Minesweeper

11/17/2014

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **10,000 samples each** | |  | eqn CPU | my fast CPU | my CPU | eqn win% | my fast win% | my win% |
| beginner | 8x8 | 0.15 | 3008 | 1770 | 5126 | 71.91% | 61.70% | 78.99% |
| intermediate | 13x15 | 0.2 | 6228 | 3757 | 41672 | 36.46% | 21.68% | 42.85% |
| expert | 16x30 | 0.2 | 16844 | 7829 | 42999 | 25.78% | 11.87% | 26.52% |
|  |  |  |  |  |  |  |  |  |
| **1,000 samples each** | |  |  |  |  |  |  |  |
| custom | 10x10 | 0.15 | 470 |  | 960 | 72.29% |  | 81.98% |
| custom | 10x10 | 0.17 | 532 |  | 1742 | 63.94% |  | 70.57% |
| custom | 10x10 | 0.19 | 455 |  | 2082 | 52.25% |  | 57.60% |
| custom | 13x13 | 0.15 | 762 |  | 1489 | 71.18% |  | 78.90% |
| custom | 13x13 | 0.17 | 825 |  | 1867 | 64.93% |  | 70.29% |
| custom | 13x13 | 0.19 | 707 |  | 3790 | 48.50% |  | 55.56% |
| custom | 15x15 | 0.15 | 1044 |  | 1142 | 70.56% |  | 80.46% |
| custom | 15x15 | 0.17 | 966 |  | 2118 | 56.55% |  | 66.90% |
| custom | 15x15 | 0.19 | 941 |  | 2875 | 46.47% |  | 58.25% |

Table 1

\*CPU times are totals, and are in milliseconds.

\*\*Win% are discounting first-click losses.

It was not clear how to count the number of times the eqn strategy guessed, so I could not perform a meaningful comparison. I could have counted the number of times it probed, but most of those would not have been guesses. From number of probes we could only have determined average game lengths.

My “fast” strategy uses the same CSP solver as my main strategy, but only uses the constraints on a given Square to calculate its probability of being a mine, instead of counting all possible board states in which a Square is a mine. We see that this loses much more often; having played a lot of minesweeper personally, I suspect that it often loses in the endgame when a handful of unexplored squares remain and a guess is forced in one of the corners.

My main strategy falls back on the “fast” strategy when there are too many Squares which are constrained (“on the frontier”), because counting states is fundamentally exponential, so it has to fall back on something or it may never complete.

The “fast” strategy uses distance to the center as a tiebreaker. I found that that **maximizing** distance to the center increased my win%, contrary to what we discussed in class about the center being a good place to guess. I think that the reason is that, because the corners and edges have fewer neighbors, they are more likely to contain a 0. As any minesweeper player will know, you can’t really start playing safely until you find a group of 0’s to build off of.

Overall, my main strategy is slower than the eqn solver but consistently outperforms it in terms of win%.

I suspect that the eqn solver is also a CSP solver, and merely has a lower maximum frontier size cutoff.

**Future Ideas:**

The maximum allowed frontier size before falling back on “fast” was maxF=30, but it would be interesting to try various allowed sizes. I suspect that most of the benefit of the state-counting guessing strategy comes in the lategame, so that small values of maxF, say maxF=10, will have most of the benefit of maxF=30 over “fast” (“fast” is effectively maxF=0).

It seems even more promising to implement a hybrid approach, so that instead of all-or-nothing counting board states or falling back on using local constraints, for each square S we count the number of neighborhood-states in which S is a bomb, i.e. states are all possible assignments of Squares within some distance of S, e.g. 2. This seems promising because it does not blow up exponentially but still gives us some benefit over the “fast” strategy; fundamentally, we are using a bounded amount of local data like “fast”, just considering a larger region “local”.