R15/Sanit Gupta/160100010

August 6, 2020

The paper attempts to solve Dots-And-Boxes which is a game popular among children across the word.

Dots-And-Boxes: There's a m x n square grid of dots. In one turn, a player draws a line joining two adjacent dots. If this line completes a 1 x 1 box, then that box is said to be captured by that player and they get an extra turn. The player with the most boxes captured at the end of the game wins.

Even though the game sounds very simplistic, its state space grows extremely quickly with the size of the gird. At the time this paper was published, games larger than 4 x 4 hadn't been solved. The authors of the paper managed to solve the game on a 4 x 5 grid.

The authors state that the optimal strategy no matter what the current situation is to capture as many boxes as possible. So, an unscored state can be maintained which only stores which edges have been filled (not who has filled them or who has captured the boxes) and no useful information would have been omitted. A m x n grid has $p = m(n+1) + n(m+1)$ edges and 2^p potential unscored states.

One thing that makes the game a little simpler is that wherever there are chains the number of possibly optimal moves is very small which makes computation easier.

They use transposition tables to make sure that the same state is not explored again and again. The impartiality of the game also helps.

Horizontal/vertical symmetry, mirror symmetry to reduce the size of the search space and another symmetry that helps them reduce the branching factor of nodes that have a pair of unfilled corner edges. For the ordering heuristic, they simply push moves that fill the third line of a box to the end, and remaining edges are considered in an order radiating downwards from the center of the board. This heuristic turned out to be very good (17x improvement in runtime).

I found their experiments and analysis to be pretty comprehensive. They tested their method on a large number of problems and they also ran tests with each enhancement individually removed to evaluate the relative contribution of each enhancement. Filling in chains turns out to be the most useful component. An interesting observations is that solving for win margin and win/loss takes similar amount of time, so it makes sense to solve for the margin which contains more information.