POST YOUR APPROACH

vdave
March 31 Marathon Match 125

Here's the usual thread for discussing solution.

I found the problem fun and interesting, I did not participate in MM5 (which apparently was something similar), so this particular game was completely new to me. My approach was fairly standard, I believe, for a given board I tried to play all possible moves, assign some kind of score to each resulting board and then simply pick the move with the highest score.

My scoring looks at lines of length 5, 6 and 7 individually and just scores each line separately and sums the score. If a line has two different colors, I simply give it a score of 0, I don't try to "fix" broken lines (e.g. one move to remove B from AABAA and then replace it with an A in a second move). For lines which only contain a single color and empty cells, I simply look up a score from a table of 32 / 64 / 128 values. To get an optimal combination for these 32+64+128=224 weights, I tried some kind of training where I initialized those 224 weights with multiple sets of somewhat randomized values (not fully random, but a weight based on number of bits set and the number of holes between set bits), run 20-50 seeds with all different weight combinations and then pick the best ones and try to mutate them slightly to get new weight candidates (this might be some sort of genetic algorithm). I trained separate weights for all different N / C combinations, so I have altogether 5 * 7 * 224 numbers hard-coded in my solution.

My initial scoring only used the lookups above, but once that was done, I picked the top 50 best moves and did an additional round of scoring, which tried to ensure that holes can be filled, e.g. if a line of length 6 matched "A.A.AA", this second round of scoring tried to find "A"s on the board which could be moved into the empty cells. In case there were no available As, I also tried to compute some kind of probability that an A will appear in the right empty-cell component.

Some things I tried but did not work out.

- Giving a penalty for moves which increase the number of connected components of empty cells, I felt this should help, but all my attempts resulted in a drop of score or being neutral.
- After I observed in the visualizer that my solution is often too optimistic about whether it makes sense to wait for a line of size 6 (as opposed to finishing it with a size of 5), I tried to use the fullness of the board as an input and trying to quickly finish short lines once the board got too full.

Things I haven't tried, but probably should have.

- I never tried to look more into the future than one single move. I could have added a third round of scoring taking the best move candidates, and play out a few scenarios with the 3 next balls appearing randomly.
- I never looked at lines of length 8 or 9, based on manually playing, getting such a big line is very difficult even on the easiest board (N=11, C=3).

COMMENTS

AmAtUrECoDeR

March 31, 2021 5:50PM

I found this problem interesting and very difficult, and for the entire match I was stuck with using rudimentary greedy solutions. My final solution is a very messy combination of the following strategies, each of which is only used for certain cases of (N,C) (deciding which strategy to use when was determined with empirical tests):

- creating the longest line of length >=5 in one move, and then outputting that move
- creating the longest line of length >=5 in two moves, and then outputting the first of those two moves
- extending the longest line of any length in one move, taking into account the "potential" of that line
  - I have two methods of measuring this "potential": the first is to pretend that empty spaces are the same color as the color of the line we are examining (the "target color"); the second is to take into account how many dots of the target color are currently on the board, not accounting for any dots that will appear in the future. Depending on (N,C), I use one or the other method.
- making the move that maximizes the length of the longest line that can be formed in the next move (whether through "fixing" broken lines or forming lines of length 6 or more)
  - this strategy sometimes formed lines of 8 or more, but I didn't go down this route of trying to make really long lines, since it seems that the extra number of random dots placed on the board during the process of making a really long line often outweighs the higher score gain from making such a line.

Sidenote: it is technically possible to remove 13 dots in a single move by having 12 same-colored dots be positioned as shown below, and then placing the 13th dot in the middle (although this seems like a bad strategy because of the reasons I stated above):

```
....1....
....1....
....1....
....1....
1111.1111
```

Quote · Reply

sullyper

March 31, 2021 9:38PM

@AmateurCoder, you can get much more than 13, manually you can create 29, and with luck 33 is possible when dots appear. You forgot the diagonals.

Personnally my approach is boring given I didn't have much time for this problem, so I went on implementing a solution similar to what I read on the forum of MM05, kind of feel like cheating though.
I scored the board as the sum of the costs of all 5 long segments.
The cost of a segment was: A * exp(B * (C * #max - #dots))
A, B, and C where picked manually with a few tries, and the results changes a lot, so I bet we can learn a much smarter cost for each segment, my quick attempt failed though, and I wanted to go ML, but given I started yesterday, it was too late to really learn anything.

Then I look at all the moves, pick the best 10, and look one depth ahead.

I tried to have a different cost for diagonal segment, given in general it's bad to form diagonal line in open board, they block the mobility, but somehow it seemed worst.

I tried to do some simulation to add randomly the next balls, but it worked in some case, not in other, and I didn't have the time to look at if I could know when it does and when it doesnt (like based on C and N).

So at the end my solution was really just what I read on the forum as every personnal attempt to improve failed =(

Quote · Reply

dimkadimon

March 31, 2021 7:48PM

@vdave I enjoyed reading your approach. I really liked your idea to learn weights for each N/C combination. Do you know how much improvement this gave you compared to using manually selected weights?

I am surprised that the penalty for number of connected empty components did not help. Intuitively it seems that it is a good idea to keep this number low to allow better movement. I wonder how can one make this work?

Quote · Reply

LEAVE A COMMENT