Shawn Frye

Minecraft Solver

TESTING NOTES

When experimenting with an 8x8 board containing 4 mines across 100 games:

* Using only the probabilistic approach, the robot achieved a mere 2% win rate without considering definite moves based on neighboring cells.
* Incorporating neighbor-based move checking elevated the win rate to a consistent 90%, with a variability of 2-3%.
* However, introducing an A\* heuristic prior to the basic probabilistic approach and interpreting the game board as a type of graph reduced the win rate to a maximum of 74%, with a 2-4% variability.

For a larger 20x20 board containing 20 mines, and having optimized processing by removing a 1-second delay between moves across 10,000 games:

* Implementing the A\* heuristic before the basic probabilistic approach and after definite neighbor checking resulted in an 82.9% win rate.
* Excluding the A\* heuristic but including definite move checking based on neighbors led to a 92.13% win rate.
* Relying solely on the probabilistic approach without considering definite neighbor-based moves resulted in a 0.6% win rate.

Conclusion: Incorporating the A\* heuristic method surprisingly reduces the win rate, especially in Minecraft. Furthermore, in Minecraft, it's essential to combine multiple strategies to achieve an optimal win rate, as no single method proves sufficiently effective on its own.