

Games

Daniel Gigliotti

1 Games

Games are search problems.

Minimax

The most common search technique in game playing is the Minimax algorithm. It is a depth-first, depth-limited, general-purpose search procedure, but it can be used in games to determine the optimal decision to make in an adversarial setting. An adversarial game is a game in which you and your opponent have opposite goals: your goal is for you to win and your opponent's goal is for them to win. Playing optimally is about ensuring the best outcome for yourself even in the worst case scenario; the worst case scenario in this case is that your opponent is playing optimally too.

Minimax(s):

```
    if Terminal(s):
        return Value(s)

    if Player(s) == MAX:
        value = -inf
        for a in Actions(s):
            value = max(value, Minimax(Result(s, a)))
        return value

    if Player(s) == MIN:
        value = inf
        for a in Actions(s):
            value = min(value, Minimax(Result(s, a)))
        return value
```

- Terminal(state) will tell us if the state is terminal (no more moves left);
- Value(state) will assign a value to the state; the value should be something that the MAX player want to maximize and that the MIN player want to minimize;
- Player(state) will tell us which player needs to do the move;
- Actions(state) will tell us which actions are available to the player that must do the move;
- Result(state, action) takes a state and an action and returns the new state resulted from applying the action to the old state.