Outline

- Games of Imperfect Information
- Games of Chance
- α-Pruning
- Resource Limits
- Perfect Play
Pruning to allow deeper search (McCarthy, 1956)

Machine learning to improve evaluation accuracy (Samuel, 1952; 57)

First chess program (Turing, 1951)

Non, 1950

Finite horizon, approximate evaluation (Zuse, 1945; Wiener, 1948; Shannon)

Algorithm for perfect play (Zermelo, 1912; von Neumann, 1944)

Computer considers possible lines of play (Bababage, 1846)

Plan of attack:

\[ \text{Time limits} \implies \text{unlikely to find goal, must approximate} \]

Specificity a move for every possible opponent reply

"Unpredictable" opponent solution is a strategy
<table>
<thead>
<tr>
<th></th>
<th>Nuclear War</th>
<th>Bridge, Poker, Scrabble</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monopoly</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chess, Checkers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Types of Games
Game tree (2-player, deterministic, turns)
Perfect play for deterministic, perfect-information games

Minimax

E.G., 2-play game:

Idea: Choose move to position with highest minimax value

= best achievable payoff against best play

Max

MIN
Algorithm

function MINIMAX-VALUE(state)

return the lowest MINIMAX-VALUE of SUCCESSORS(state)

else

return the highest MINIMAX-VALUE of SUCCESSORS(state)

else if MAX is to move in state then

return UTILITY(state)

else if TERMINAL-TEST(state) then

return a utility value

end function

function MINIMAX-DECISION(state, game)

return an action

such that MINIMAX-VALUE(game, state) is maximized

action, state -> the a s in SUCCESSORS(state)

function MINIMAX-ALGORITHM
Resource Limits

= estimated desirability of position

• evaluation function

• e.g., depth limit (perhaps add quiescence search)

• cutoff test

Standard approach:

Suppose we have 100 seconds, explore 10^4 nodes/second

\[ 10^6 \text{ nodes per move} \]
\[(s)^f m + \cdots + (s)^f m + (s)^f m = (s)^f m\]

For chess, typically linear weighted sum of features.
Payoff in deterministic games acts as an ordinal utility function. Only the order matters.

Behavior is preserved under any monotonic transformation of $\text{EVAL}$.
4-ply lookahead is a hopeless chess player!

\[ 4 = m \leq 35 = q \leq 10^6 \]

Does it work in practice?

1. TERMINAL is replaced by \texttt{MAX}
2. \texttt{UTILITY} is replaced by \texttt{MIN}
3. \texttt{MIN} and \texttt{MAX} are identical to \texttt{MINMAXVALUE} except

Cutting off search
α-β pruning example
relevant (a form of metareasoning)

A simple example of the value of reasoning about which computations are

can easily reach depth 8 and play good chess ⇐
doubles depth of search

With "perfect ordering" time complexity = \( \frac{2^n}{m!} \)O

Good move ordering improves effectiveness of pruning

Pruning does not affect final results

Properties of \( \mathcal{P} \)
suggest plausible moves. In 60, \( q < 300 \), so most programs use pattern knowledge bases to bad. \[ \text{Go: human champions refuse to compete against computers, who are too too good.} \]

Othello: human champions refuse to compete against computers, who are some lines of search up to 40 ply.

uses very sophisticated evaluation and undisclosed methods for extending Game match in 1997. Deep Blue searches 200 million positions per second.

Chess: Deep Blue defeated human world champion Gary Kasparov in a six-

positions. positions involving 8 or fewer pieces on the board, a total of 443,748,401,247

Tinsley in 1994. Used an endgame database defining perfect play for all

Checkers: Chinook ended 40-year-reign of human world champion Marion

Deterministic Games in Practice
Simplied example with coin-flipping:

In nondeterministic games, chance introduced by dice, card-shuffling.

**Non-deterministic Games in General**
TD-GAMMON uses depth-2 search + very good EVAL.

α pruning is much less effective.

As depth increases, probability of reaching a given node shrinks.

\[
\text{depth 4} = 20 \times 2^3 \times 22 \times 0.9 \approx 0.3 \times 20 \times 2^3 \times 22 \times 0.9
\]

Backgammon 20 legal moves can be 6,000 with I-1 roll.

Dice rolls increase by 21 possible rolls with 2 dice.

Non-deterministic Games in Practice
2) Picking the action that wins most tricks on average

1) Generating 100 deals consistent with bidding information

CIB, current best bridge program, approximates this idea by

Special case: if an action is optimal for all deals, it's optimal.

then choose the action with highest expected value over all deals

Idea: compute the minimax value of each action in each deal,

Seems just like having one big dice roll at the beginning of the game

Typically we can calculate a probability for each possible deal

E.g., card games, where opponent's initial cards are unknown

Games of Imperfect Information
Four-card bridge/whist/hearts hand, MAX to play first

Example
Four-card bridge/whist/hearts hand, Max to play first.

Example
Four-card bridge/wiust/heartts hand. MAX to play first.

Example
Games are to AI as Grand Prix racing is to automobile design.

Uncertainty constrains the assignment of values to states.

They illustrate several important points about AI.

Good idea to think about what to think about.

Perfection is unattainable. Must approximate.

Summar