Game Playing in Artificial Intelligence

Param Singh

14MI517
Computer Science and Engineering Department
National Institute of Technology, Hamirpur

19th March, 2017

Introduction

- One of the most studied and most interesting areas of Artificial Intelligence
- Fun but a little hard to create.

Definition

- Game artificial intelligence refers to techniques used to produce the illusion of intelligence in the behavior of the computer.
- Emphasis is on developing rational agents that match or exceed human performance in the game.
- This generally works really well, and computer abilities must be toned down to make the experience better for human players.

Definition (cont...)

- AI has continued to improve, with aims set on a player being unable to tell the difference between computer and human players.
- A game AI must generally have the following characteristics:
  - Should feel “natural”.
  - Should obey the laws of the game.
  - Should be competitive.

Why are games relevant to AI?

- Game playing is considered an intelligent human activity.
- They're fun!
- Clear, well-defined rules with an easy evaluation of success.
- Huge state spaces
- Teaches us how to deal with other agents actively working against us.

History

- Humble beginnings in the 1940s with some theories
- 1956 - A conference at Dartmouth, Arthur Samuel develops a program that plays Checkers which learned from its previous games
- 1958 - alpha beta pruning is developed!
- 1962 - A computer defeats Robert Nealy, a master Checkers player for the first time.
- 1997 - Gary Kasparov, dominant Chess World Champion is beaten by IBM's Deep Blue
- Humans stop winning Chess and Checkers at all against the top computers.
- 2016 - Google's AlphaGo beats Lee Sedol, ranked among the best players of Go in the world.

Game playing as Search problems

It is easy to model game playing as a search problem defined by:

- An initial state
- A successor function
- A goal test
- A path cost

Problems that are faced here include:

- Big state spaces
- Time limits
- Unpredictable opponents

Minimax Algorithm

- An algorithm for perfect play in deterministic, perfect information games
- The basic idea is simulate all moves and calculate their minimax values i.e best achievable payoff against perfect play by the opponent and play the move with the best minimax value.
- Used as a general approach for nearly the entire history of game playing programs.

Minimax: Example

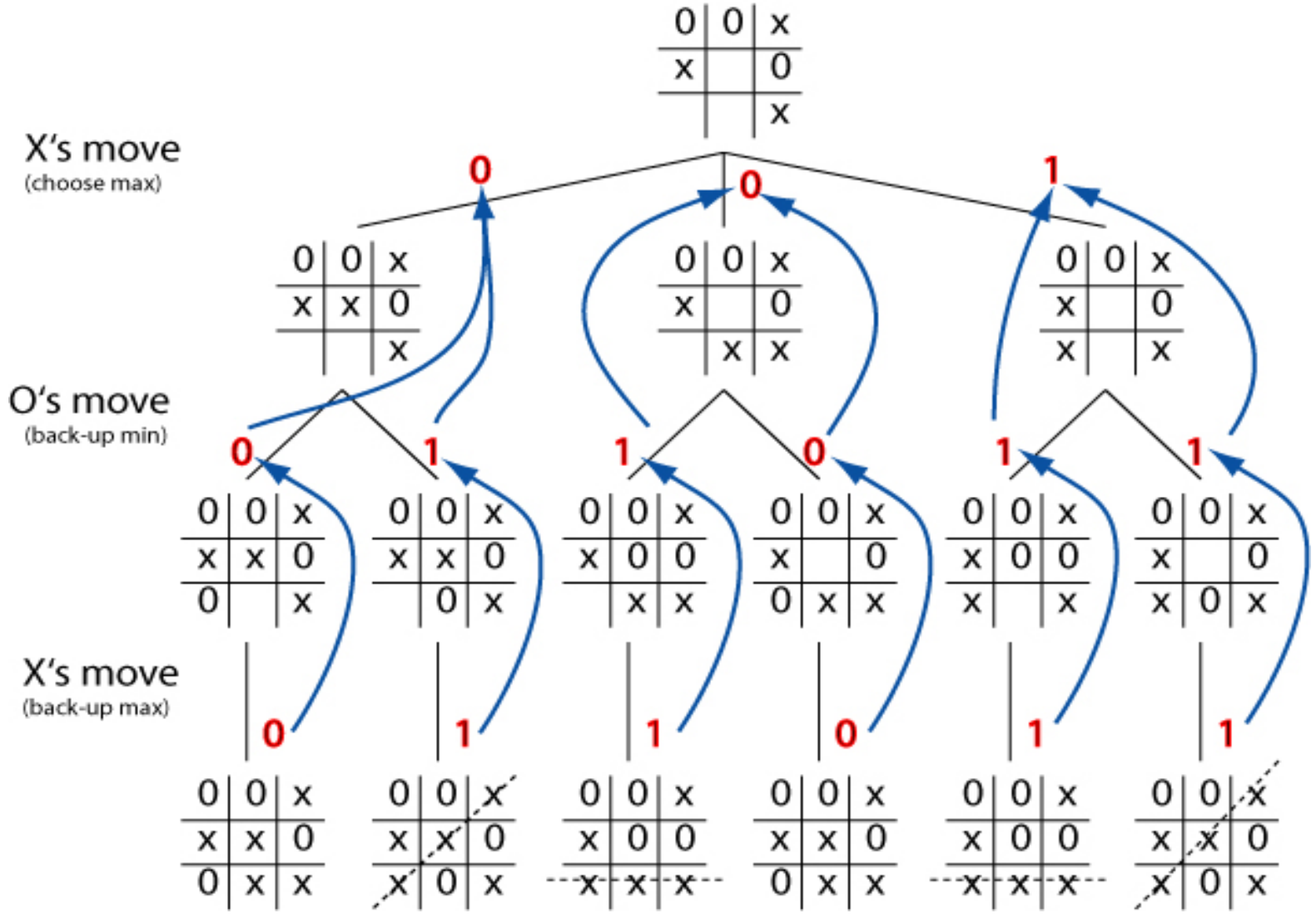

Figure: Game tree for Tic-tac-toe

Minimax: Properties

- Complete: Yes, if tree is finite (Chess has specific rules for this)
- Optimal: Yes, against an optimal opponent.
- Time: $O(b^m)$
- Memory: $O(bm)$

Resource Limits

- Can't move through the entire search tree.
- Chess, for example, has $b = 35$ and $m = 100$ for average games.
- Standard approach involves cutting off search after certain depth.
- Hence, we have to use evaluation functions for values of positions in the game.

Evaluation Functions

- Used to estimate the desirability of a particular position in the game for a player.
- In Chess, measured in centipawns.
- A typical evaluation function is a weighted sum of features.

$$EVAL(s) = w_1 f_1(s) + w_2 f_2(s) + \ldots + w_n f_n(s)$$

Cutting off Search

- Cutting off search in a brute force minimax does not work in practice.
- A hopeless chess player can see ahead 4 plys.
- A master sees ahead 8 plys on average while Carlsen sees around 12 plys ahead.
- Stockfish, the world's best chess engine, is known to find mates in 16 easily.
- As a result, we need to prune parts of the game tree.

Alpha-Beta Pruning

- The basic idea is to not evaluate (prune) parts of the game tree that will never be reached according to our algorithm.
- The pruning has no effect on final result at the root.
- The values of intermediate nodes may be incorrect though.
- With perfect ordering, time complexity drops to $O(b^{m/2})$, which is effectively doubling solvable depth.

Alpha-beta pruning: Example

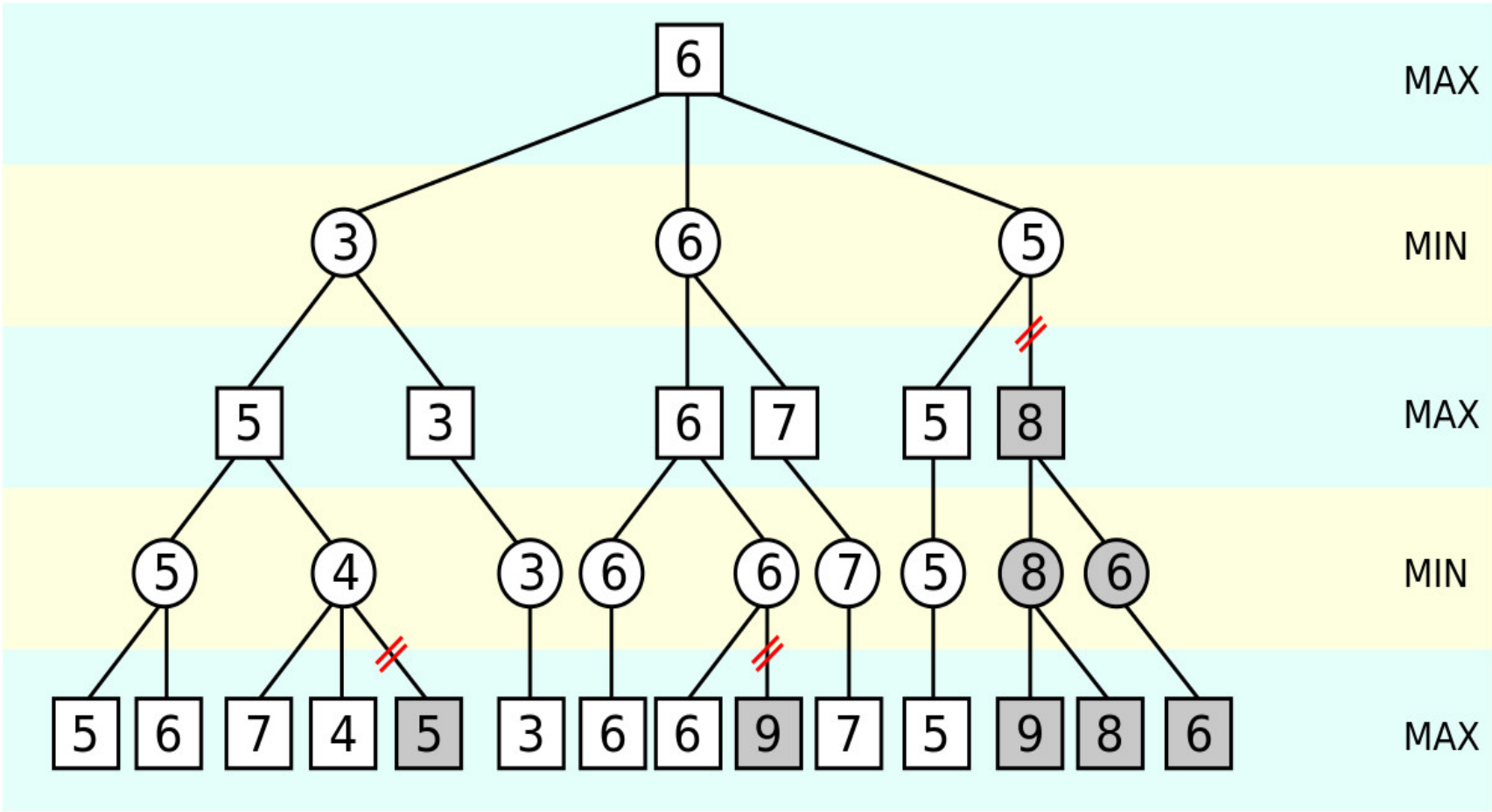

Figure: Alpha-Beta pruning

Other optimizations

- Iterative Deepening: Useful in games where moves are timed.
- Involves going upto depth 1, then if time remains go until depth 2 and so on.

Search vs Lookup

- It is easy to keep some common positions in the program's database so that it doesn't have to work on searching for best plays in these situations.
- Examples include endgames and openings in Chess and Checkers.
- Chinook, the Checkers engine that ended a 40-year reign of then world champion Marion Tinsley, kept a database defining perfect play for all positions involving 8 or fewer pieces on the board, a total of 443,748,401,247 positions.
- Today, opening tables in Chess contain trillions of master games to make sure that opening play is optimal without much wastage of resources.

Probabilistic games

- Games involving chance, such as backgammon, use a modification of the minimax algorithm called the Expectiminimax algorithm.
- Here, the basic idea is to take the expected values of the minimax value returned and then try to choose the best value.
- Of course, this is not as good as perfect information, deterministic games.

The state-of-the-art in game-playing tech

- The open-source Chess engine, Stockfish, is widely considered to be unbeatable by humans at its best.
- Google's AlphaGo created waves in 2016 when it beat 18-time world champion Lee Sedol in a 5 game match by 4-1.
- AlphaGo uses Monte Carlo search with experience based on a large database of master games and games played by itself as well.

Queries?

Questions?

For Further Reading

Peter Norvig and Stuart Russell
Artificial Intelligence: A Modern Approach.