Role of Search in Computer Games

CS344 Seminar Presentation

Group 4
Team Members
(In the lexicographic order of roll number):

- Adhip Agarwal (07005009)
- Raman Sharma (07005010)
- Gaurav Malpani (07005011)
- Sumit Somani (07005012)
- Abhinav Gokari (07005029)

Motivation

Search for Osama
Courtesy Asia News

The Atlantis City
http://whoyoucallingaskeptic.wordpress.com/2009/02/17/atlantis/

Search for Holy Grail

Hide & Seek

Search Engines

Search for AIDS cure

Search for extra-terrestrial intelligence
The Allen Telescope Array.
(Credit: Image courtesy of SETI Institute)

Search

Search is to Try and Find something

In AI, a problem is “a goal and a set of means for achieving that goal”.
[Courtesy: **Beyond Adversarial**: The Case for Game AI as Storytelling]

Search is the process of exploring the possible ways in which these means can be applied to realize the goal.

Attempt the end, and never stand to doubt.
Nothing's so hard but search will find it out.
- Robert Herrick

Computer Games

A computer game is a game that involves interaction with a user interface to generate visual feedback on a video device.

(Courtesy : Wikipedia)

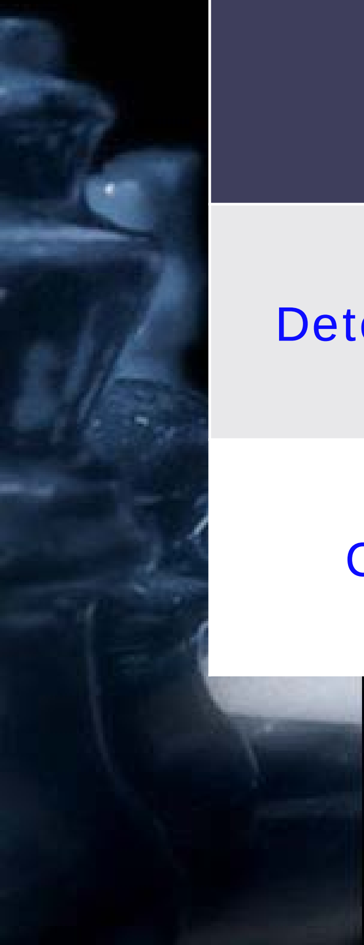
Assassin’s Creed

Crysis

Classification of Games

S.No	Type of Game	Examples
1.	Real Time Strategy Games (RTS)	AOE, AOM, WoW.
2.	First-Person Shooter Games (FPS)	Wolfenstein, Counter Strike, Hitman.
3.	Role-Playing Games(RPG)	Sims, Tribal Wars, Khan Wars.
4.	Turn-Based Strategy Games (TBS)	Civilization 4, Heroes of Might & Magic.
5.	Simulation Games (SIM)	GL117, Sims
6.	Sports Games(SPT)	FIFA, PES, Cricket.

Classification of Games (contd.)



	Perfect Information	Imperfect Information
Deterministic	Go, Chess, Othello, Checkers, etc.	Stratego, Battleships, etc.
Chance	Backgammon, Monopoly, etc.	Bridge, Poker, Scrabble, etc.

When (& Where)

God is a challenge because there is no proof of his existence and therefore the search must continue. – Donald Knuth

If God is everywhere then search is everywhere.

But search in comp. games is not as vague as search for God :P.

It can be summarized as states or places where

- There are more than one choices available.
- There is controlled change to the outcome of the game.

When (& Where)

"There are more than one choices available."

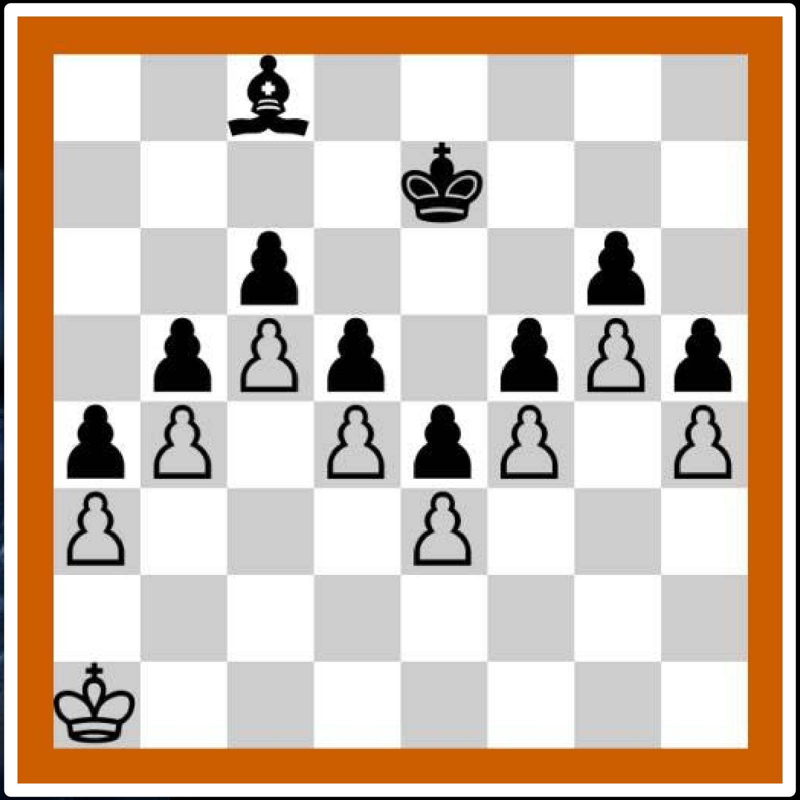
In checkers if one player's piece, the other player's piece, and an empty square are lined up, then the first player must "jump" the other player's piece.

OR in chess, if one's king is under check, and he has only one move to make for the king, then there is no choice for the player but to move the king to that particular block.

No choice → No Search

When (& Where)

"There is controlled change to the outcome of the game."



A possible chess game scenario – The pawn wall

White – Human Player

Black – AI player

Clearly choices are available, but there would be no change to the outcome of the game so search is useless.

What to Search

"Computers are useless. They can only give you answers" – Pablo Picasso

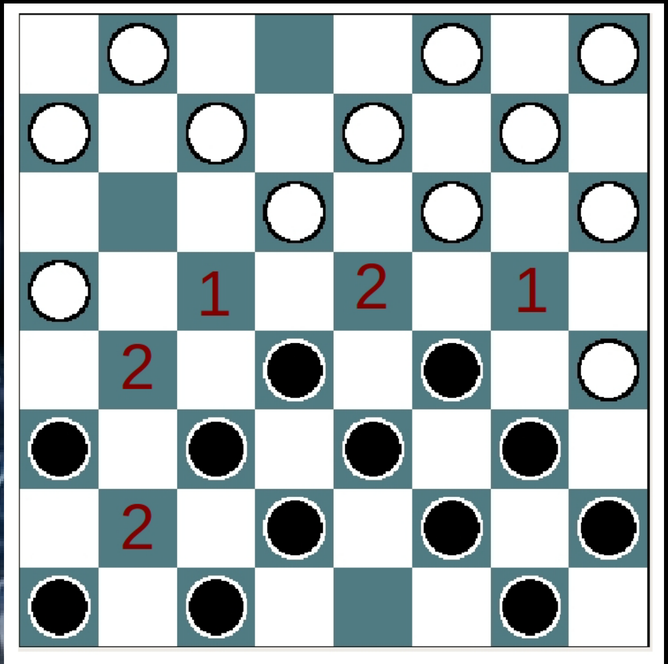
So let us make them useful by giving them the correct question to find answer to.

Non Trivial and Requires Human Intelligence

In subsequent slides we will study what all we search for in a computer game

What to Search (contd.)

Search for the optimal choice



A possible checkers game scenario

Search for the optimal choice among the available choices – This search involves looking for choices available and also metrics for search.

What to Search

Search for information

Application of ray casting for vision
*Courtesy : **Illusion of Intelligence** by C. Butcher and J. Griesemer*

- In games with incomplete information at times, player (AI specifically) has to search for information that too in manner that tends to imitate the human capabilities.
- *"This is my goal. Where should I be standing?"* Need a discrete answer to a continuous problem
- E g:- AI in Halo imitates techniques like vision, hearing, touch, sense, etc to get information and behave accordingly. In Halo it is done by using multiple ray-casting for vision

What to Search (contd.)

Search for strategies and patterns

A screenshot of the game PES

- Search & Identify for strategies and patterns of regular player
- Develop Counter strategies or improve itself against the same.
- Konami PES (2008 onwards) included a component to their AI called as Team Vision which is focussed in improving or evolving over time
- But most of our focus would be on Search for the optimal choice among the available choices.

Game Tree

- **Initial State**: Initial board position and player.
- **Operators**: One for each legal move
- **Terminal states**: A set of states that mark the end of the game.
- **Utility function**: Assigns numeric values to each terminal state.
- **Game tree**: Represents all possible game scenarios.

How to Search

- Brute-force or Uninformed search
- A* algorithm
- Minimax and its derivatives
  - Normal
  - Alpha-Beta Pruning
  - Iterative Deepening (can also be used with other algorithms)
  - Expectiminimax for Chance Games
- Many more

Brute-force or Uninformed Search

Given the graph of states in a 2-player game, this search uses basic BFS to search for the most optimal move at a given state.

Chinook and Deep Blue are examples of *search by brute force,* the exploitation of search and knowledge on a heretofore-unprecedented scale. Each of them had a search engine that explored enormous sub trees and supported that search with extensive opening and closing books.

Courtesy : ***Game Playing: The Next Moves*** *by Susan L. Epstein*

Minimax algorithm

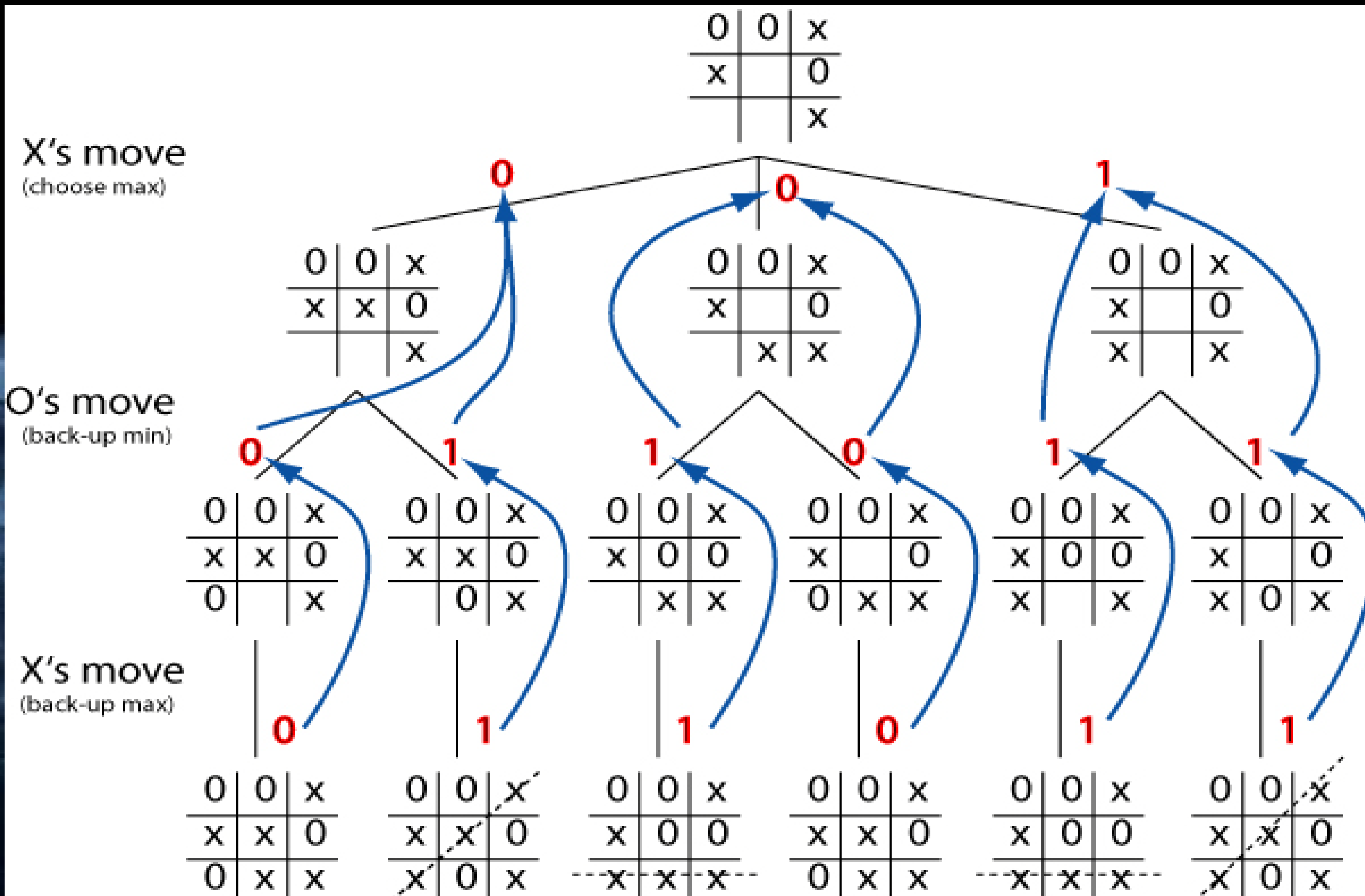
Competitive environments, in which the agents' goals are in conflict, give rise to **adversarial search** problems – alternatively known as games.

The most prevalent adversarial search algorithm is ***minimax*** (Invented by Von Neumann & Morgenstern in 1944 as part of game theory).The idea behind minimax is that an AI player looks ahead to predict the outcome of the different possible moves available to them.

The player's goal is to maximize payoff, while the opponent's goal is to maximize their own payoff which amounts to minimizing the player's payoff (due to the *zerosum* nature of the game).

$$\text{Minimax}(n) = \begin{cases} \text{Utility}(n) & \text{if } n \text{ is a terminal node} \\ \text{Max}_{s \in \text{successor}(n)}(s) & \text{if } n \text{ is a max node} \\ \text{Min}_{s \in \text{successor}(n)}(s) & \text{if } n \text{ is a min node} \end{cases}$$

An example



Courtesy : www.cogs.susx.ac.uk/courses/kr/lec05.html

Properties of Minimax

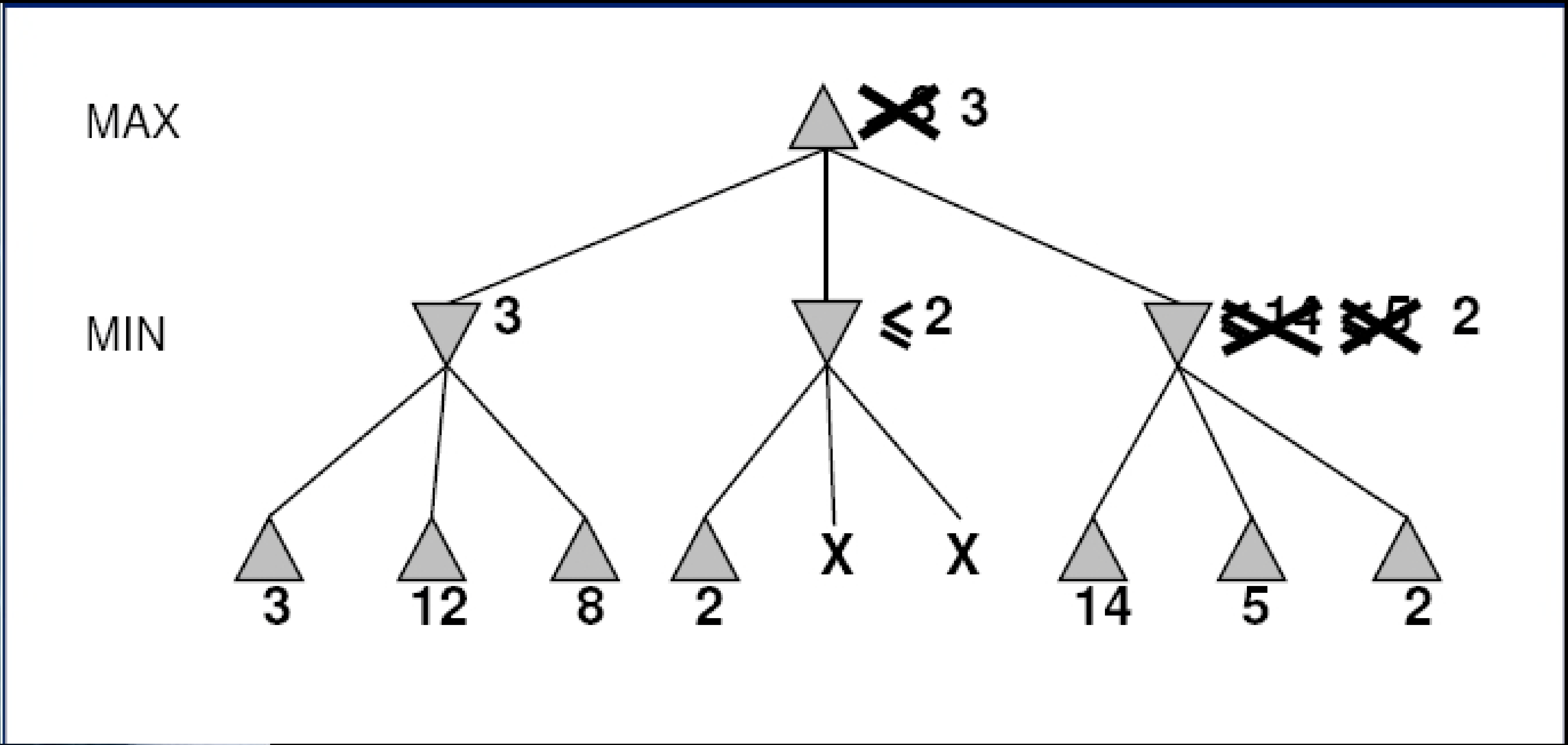
b – branching factor or legal moves at a point

m – maximum depth of the tree

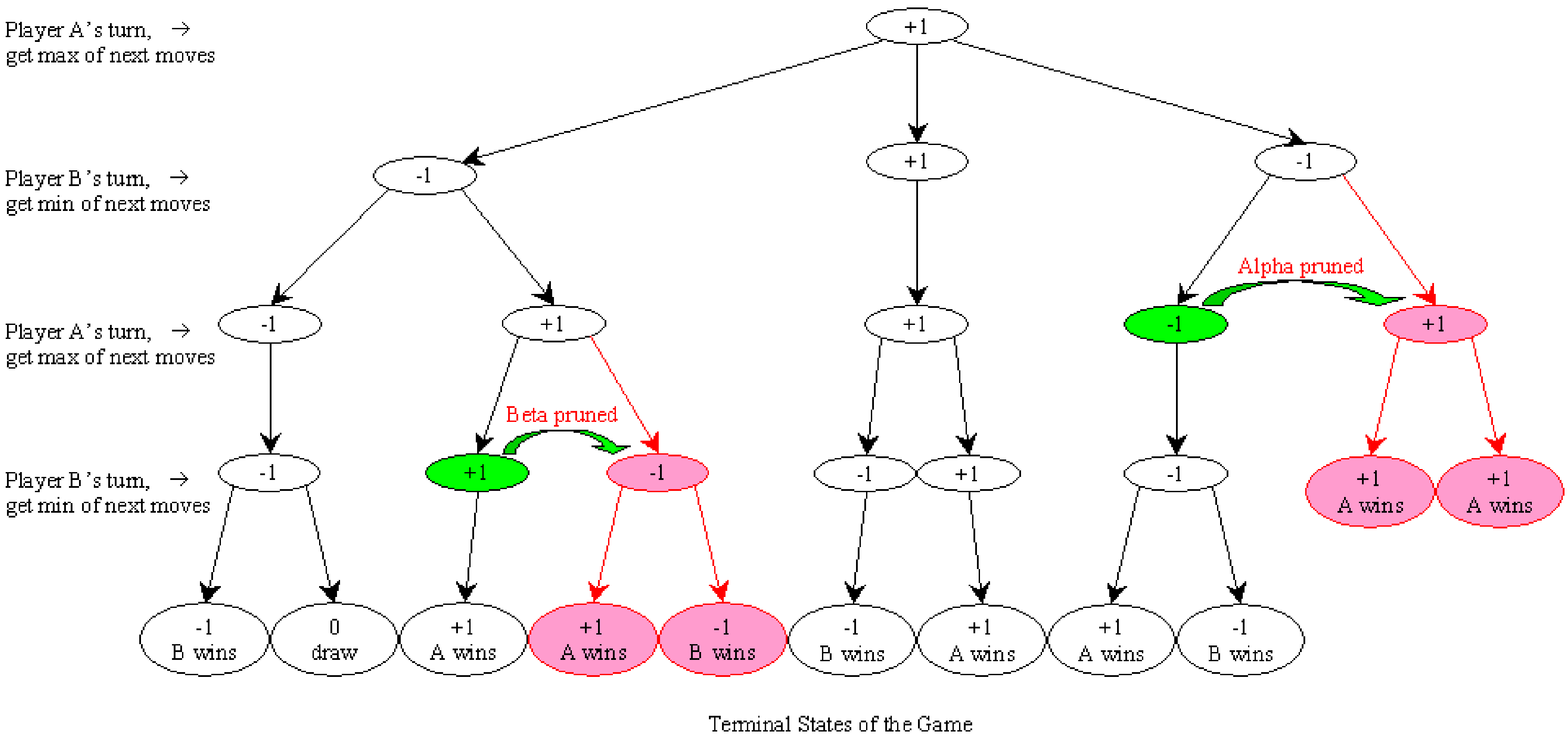
- **Complete**: Yes, if tree is finite (chess has specific rules for this)
- **Optimal**: Yes, against an optimal opponent. Otherwise??
- **Time complexity**: $O(b^m)$
- **Space complexity**: $O(bm)$ (depth-first exploration)

For chess, b ~ 35, m ~ 100 for “reasonable” games => exact solution completely infeasible

Alpha-Beta Pruning



Alpha-Beta Pruning



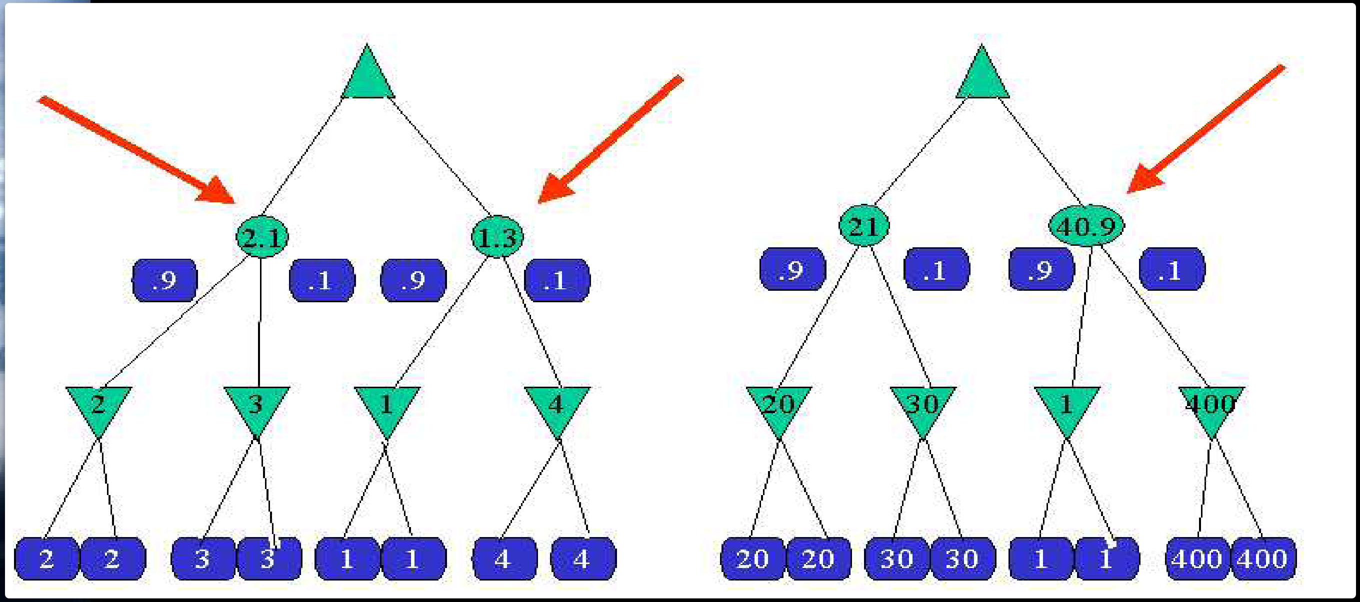
The state in green is where one can determine that the states in pink can be pruned.
Note that the values of the children of the green states must be calculated before the value of the green state itself can be determined..

α – highest value found along the path so far for MAX
β – lowest value found along the path so far for MIN

Element of chance

Backgammon is a game which combines both luck and skill. Dice are rolled at the beginning of a player's turn to determine the legal moves, thus a standard game tree can't be constructed.

A game tree in backgammon must include “chance nodes” in addition to MAX and MIN modes.



A typical chance game tree

Limiting Search

The critical problem of search is the amount of time and space necessary to find a solution. As the chess and checkers estimates suggest, exhaustive search is rarely feasible for nontrivial problems. Examining all sequences of n moves, for example, would require operating in a search space in which the number of nodes grows exponentially with n. Such a phenomenon is called a *combinatorial explosion*.

Limitations of minimax

It requires search all the way to the terminal nodes which becomes infeasible in the context of games which require a move within a certain time limit. The solution is to cut off search earlier. (*Programming a computer for playing chess, Shannon, 1950)*

The required modifications are:

- **Utility function => Eval function : Returns an estimate** of the expected utility of a game tree node heuristically.
- **Terminal test => Cutoff test:**

  if CUTOFFTEST(state, depth) then return EVAL(state)

Evaluation function

Uses a heuristic for estimation of the expected utility of a game tree node. Humans do it all the time because of their limitations in search.

Question: How to design good evaluation functions?

Properties:

✔ Eval(terminal nodes) must be equal to Utility(terminal nodes).

✔ It must be computationally fast.

✔ It must actually give an indication of actual *“chances of winning”* via its value of a node.

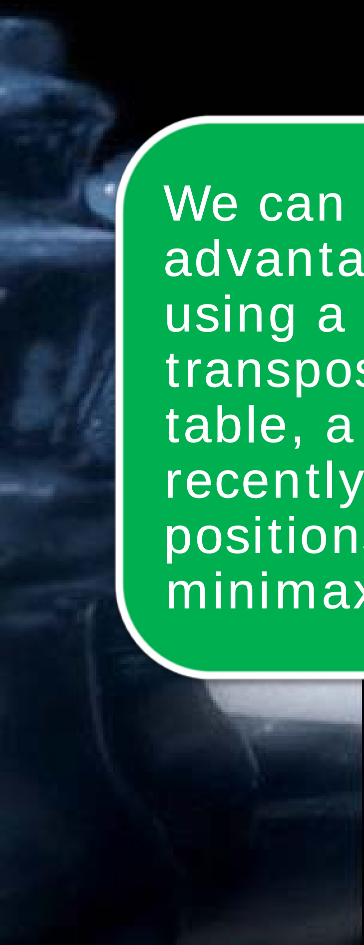
Cutting off search

In general, **CUTOFFTEST(state, depth)** is designed such that it returns true for all depth ≥ d, and also for all terminal nodes. 'd' is chosen appropriately, so as to meet the time constraints.

"Iterative Deepening" is a robust way of applying this concept, with the feature that when time runs out, the program returns the move selected by the deepest completed search.

Speeding Up Search

- Database of standard openings
- Thinking on the opponent's time
- Iterative Deepening
- Transposition table
- Move ordering
- Specialized hardware



We can take advantage by using a transposition table, a cache of recently visited positions and their minimax values.

Courtesy **Adversarial Search** By *Dana S. Nau,* University of Maryland

Search in End games

- Aims at making end-game playing completely flawless.
- Calculate for all positions for a few end pieces and develop database for all position leading to flawless game.
- Uses – It effectively increases the depth algorithms can go while analyzing their moves, as soon as if they reach one of the solved states the job is done.
- E.g. – Chinook, the state of the art computer program for checkers, used a database of 444 billion positions with 8 pieces or less to make its endgame flawless.
- Deep Blue also uses an enormous end game database containing all positions with 5 pieces and many with 6 pieces.

Search in MMORPG

- MMORPG is Massively Multiplayer Online Role Playing Games E.g. –World of Warcraft, Everquest, Tribal wars.
- Search to detect problematic behaviour, work against teammates, undue advantages, code hacks, violation of game play conventions
- Search for obscene language in text-based communication -an NLP problem
- Search for information in databases, to manage games - a database and IT system problem

BEYOND ADVERSARIAL SEARCH

- ✓For industrial developers primary design consideration → "fun"
- ✓Common in RTS, individual non-player characters (NPCs) in FPS.
- ✓Invincible AI creates disinterest
- ✓Designers thus dumb down AI systems by limiting
  - ➢Perceptual Abilities
  - ➢Computational Resources
- ✓West proposes that the use of "intelligent mistakes" is a better approach [6]

E.g. of ILLUSION of INTELLIGENCE

- The classic arcade game Pac Man makes the player believe that the enemies hunting him - the ghosts - are intelligent pack-hunters. In fact this perception of group-intelligence is only an illusion. To make sure that the ghosts do not all follow the same route through the maze and to provide them with an individual personality, they are each given a slightly different variation of the same algorithm which is a very simple alternative selection of the direction whenever the ghosts reach a junction in the maze. If a junction is reached the ghost needs to decide whether it should change it's direction or not - sometimes the ghost changes it's direction to move in the direction of the player, sometimes it chooses a random direction.
- One ghost may move in a random direction 75% of the time and in the direction of the player in the other 25% of cases when it reaches a junction. Another ghost would have the random choice of direction weighted at 50% of the time etc.

GAMES, SEARCH and AI

- Historical Relevance – Search algorithms have been applied to games.
- Game Playing – one of the first tasks undertaken by AI
- Chess was tackled by
  - Konrad Zuse,
  - Claude Shannon,
  - Norbert Weiner and
  - Alan Turing
- AI researchers find abstract nature of games, an appealing subject for study
- Games unlike toy problems are interesting because they are too hard to solve

GAMES, SEARCH and AI

Games are thus interesting (and difficult too)

Some interesting points that games reveal about AI

- Perfection is unattainable → Must approximate
- Good idea is to think about what to think about
- Uncertainty constrains the assignment of values to states

Conclusion

- Search is everywhere
- Search is core to writing Game AIs
- Large number of techniques exist for search and its usage depends on the games involved.
- Games and AI problems have strong resemblances, thus generates interest
- Search has applications in evolutionary AI, and developing counter-strategies on-fly remains topic for active research.

References

1. Susan L. Epstein. Game playing: The next moves. In Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference, pages 987–993. Association for the Advancement of Artificial Intelligence (AAAI), Menlo Park, CA, USA, july 1999.
2. David L. Roberts, Mark O. Riedl, and Charles L. Isbell. Beyond adversarial: The case for game AI as storytelling. In Atkins Barry, Kennedy Helen, and Krzywinska Tanya, editors, Breaking New Ground: Innovation in Games, Play, Practice and Theory: Proceedings of the 2009 Digital Games Research Association Conference, London, September 2009.

References

3. *Eike Falk Anderson. Playing smart artificial* intelligence in computer games. In Proceedings of zfxCON03 Conference on Game Development. ZFX3D Entertainment, 2003.
4. *Stuart J. Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. ISBN* 0131038052. Prentice Hall, 2nd edition, December 2002.
5. *Dana S. Nau. Adversarial search. In Encyclopedia of Cognitive Science. Nature Publishing Group,* 2002.

References

6. *David Pinelle, Nelson Wong, Tadeusz Stach, and Carl Gutwin. Usability heuristics for networked* multiplayer games. In GROUP ’09: Proceedings of the ACM 2009 international conference on Supporting group work, pages 169–178, New York, NY, USA, 2009.
7. *Jaime Griesemer and Chris Butcher. The Illusion of Intelligence : The Integration of AI and* Level Design in Halo. Game Developers Conference, 2008.
8. *Mick West, Neversoft cofounder.* Intelligent Mistakes: How to Incorporate Stupidity Into Your AI Code. Game Developer magazine, Gamasutra.com, 2009.

References

9. http://www.gamedev.net/reference/list.asp?categoryid=18
10. http://www.intelligentgo.org/en/computergo/overview.html
11. http://www.gamasutra.com/features/index_ai.htm
12. www.cogs.susx.ac.uk/courses/kr/lec05.html

Thank You For Your Attention And Time

Questions
???