GAME PLAYING
MINIMAX ALGORITHM
ALPHA-BETA PRUNING

Fundamentals of Artificial Intelligence

Session 12

Pramod Sharma
pramod.sharma@prasami.com

2

Agenda

- Games
- Minimax Search
- Alpha-Beta Pruning
- Chance Games
- Nonzero Sum Games

3

Games

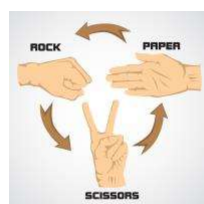
- So far single Agent
- There can be multi agent problems
- Good reasoning problems
  - Formal rules
  - Non-trivial
  - Time tested
- Direct comparison to human performance!

4

Type of Games

Sequence of moves

Opponent also moves



Both play simultaneously

- Rewards for each move
  - May be, may be not
- Defined win conditions

Search versus Games

- Search – no adversary
  - Solution is (heuristic) method for finding goal
  - Heuristics and Constraints Satisfaction Problems techniques can find optimal solution
  - Evaluation function: estimate of cost from start to goal through given node
  - Examples: path planning, scheduling activities

Search versus Games

- Games – adversary
  - Unpredictable opponent
    - You make a move,
    - Opponent makes a move
  - Solution is strategy
    - Strategy specifies move for every possible opponent's reply
  - Optimality depends on opponent. Why?
  - Time limits
    - Large problems, huge state space, not enough compute power
    - Force an approximate solution
  - Evaluation function: evaluate "goodness" of game position
  - Examples: chess, checkers, Othello, backgammon

7

Adversarial Search

- Examine the problems that arise when we try to plan ahead in a world where other agents are planning against us
- A good example is board games
- Adversarial games, while much studied in AI, are a small part of game theory in economics

pra-sâmi

8

Games - Typical Assumptions

- Two agents whose actions alternate
  - May be more than 2
- Utility values for each agent are the opposite of the other
  - Creates the adversarial situation
  - You want to maximize your position so does your opponent
  - Both functions are interdependent
  - Simply put, you want to maximize your position and your opponent wants to minimize your position
- Fully observable environments
- In game theory terms: Zero-sum games of perfect information.
- Generalizes to stochastic games, multiple players, non-zero-sum, etc.
- Relaxation in these assumptions make it more complex

pra-sâmi

9

Types of Games

	Deterministic	Chance Moves
Perfect Information	Chess, Checkers, Go, Othello	Backgammon, Monopoly
Imperfect Information (Initial Chance Moves)	Skat, Battleship, Kriegspiel	Bridge, Poker, Scrabble, Blackjack

Not Considered: Physical games like tennis, croquet, ice hockey, etc.
(but see “robot soccer” http://www.robocup.org/)

10

Perfect and Imperfect Information Games

- A perfect information or Markov game is fully observed

Chess

Backgammon

Othello

Go

Checkers

- An imperfect information game is partially observed
  - Scrabble
  - Poker

11

Environment Types

- Turn-taking: Semi-dynamic
- Deterministic and non-deterministic

4/13/2024

pra-sâmi

12

Size of Search Trees

- b = branching factor
- d = number of moves by both players
- Search tree is O(b^d)
- Chess
  - b ~ 35
  - D ~100
    - Search tree is 35^{100} = 2.5516E+154 (!!)
    - Completely impractical to search this
- Game-playing emphasizes being able to make optimal decisions in a finite amount of time
  - Somewhat realistic as a model of a real-world agent
  - Even if games themselves are artificial

4/13/2024

pra-sâmi

13

Two Fundamental Branches

- Reinforcement Learning:
  - The environment is initially unknown
  - The agent interacts with the environment
  - The agent improves its policy

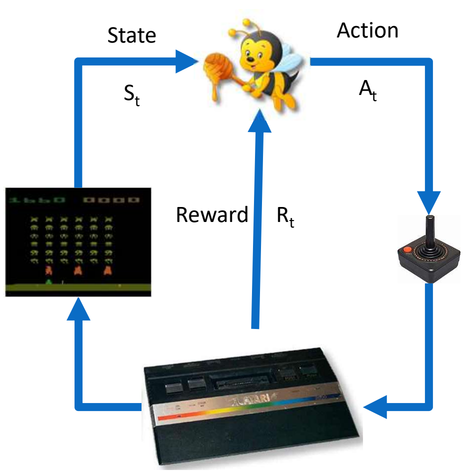
- Planning:
  - A model of the environment is known
  - The agent performs computations with its model (without any external interaction)
  - The agent improves its policy a.k.a. deliberation, reasoning, introspection, pondering, thought, search

4/13/2024

pra-sâmi

14

Atari Example: Reinforcement Learning

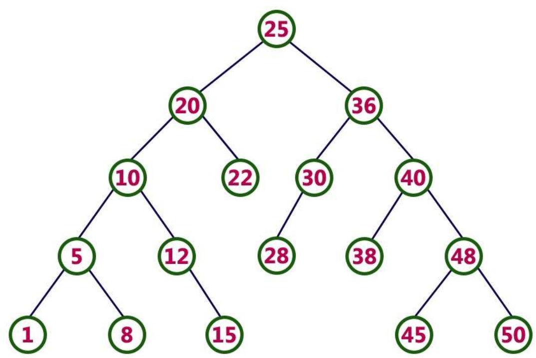


- Rules of the game are unknown
- Learn directly from interactive game-play
- Pick actions on joystick, see pixels and scores

4/13/2024

pra-sâmi

15
Atari Example : Planning



- Rules of the game are known
- Can query emulator
  - Perfect model inside agent's brain?
- If I take action 'a' from state 's':
  - What would the next state be?
  - What would the score be?
- Plan ahead to find optimal policy
  - Tree search

16
Game Setup

- A two-player game has two (alternating) players: MAX and MIN
  - MAX moves first and they take turns until the game is over
- Games as search:
  - Initial state: e.g. board configuration of chess
  - Successor function: list of (move, state) pairs specifying legal moves.
  - Terminal test: Is the game finished?
  - Utility function: Gives numerical value of terminal states
- A zero sum game has equal and opposite rewards for black and white
  - $R_w + R_b = 0$
  - Winner gets reward, loser gets penalty
  - Tic-Tac-Toe : win (+1), lose (-1) and draw (0)
  - Chess : win (+1), lose (-1) and draw ($\frac{1}{2}$)
- Both methods are applicable in such cases
  - Game tree search (i.e. planning)
  - Self-play reinforcement learning

17

Game Tree (2-player, Deterministic, Turns)

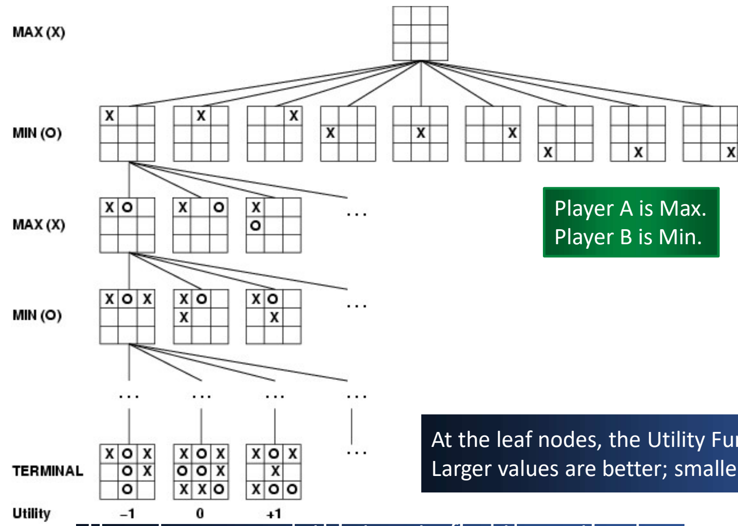
Player A

Player B

Player A

Player B

Leaf Nodes



Player A is Max.
Player B is Min.

At the leaf nodes, the Utility Function is employed. Larger values are better; smaller values are worse.

How do we search this tree to find the optimal move?

pra-sâmi

18

Minimax Terminology

- Move: a move by both players
- Ply : one players move; a half-move
- Utility Function : function applied to leaf node
- Backed-up values
  - For Max : the values of largest successor
  - For Min : the values of smallest successor
- Minimax Procedure:
  - Search down several levels
  - At the bottom apply utility function
  - Back-up values all the way up to root node
  - Root selects the move

pra-sâmi

19

Minimax Search

- Formal definition as a search problem:
  - Initial state: Set-up specified by the rules, e.g., initial board configuration of chess.
  - Player(s): Defines which player has the move in a state.
  - Actions(s): Returns the set of legal moves in a state.
  - Result(s, a): Transition model defines the result of a move.
  - Successor function: list of (move, state) pairs specifying legal moves
  - Terminal-Test(s): Is the game finished? True if finished, false otherwise.
  - Utility function(s, p): Gives numerical value of terminal state 's' for player p.

- MAX uses search tree to determine next move

4/13/2024

pra-sâmi

20

An Optimal Procedure: The Minimax Method

- Designed to find the optimal strategy for Max and find best move
  - Assuming an infallible MIN opponent

- Assumption: Both players play optimally!

- Definition of optimal play for MAX assumes MIN plays optimally:
  - Maximize worst-case outcome for MAX

- But if MIN does not play optimally, MAX will do even better

4/13/2024

pra-sâmi

21

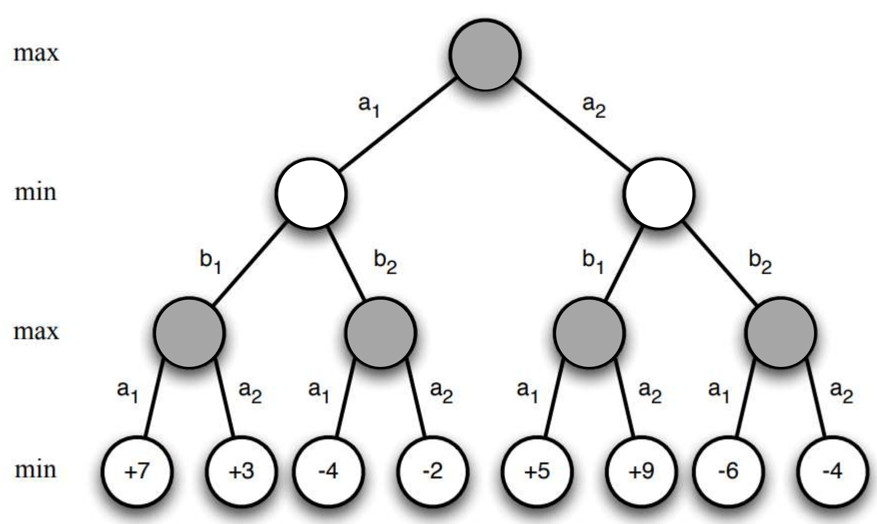
An Optimal Procedure: The Minimax Method

- Step 1. Generate the whole game tree, down to the leaves
- Step 2. Apply utility (payoff) function to each leaf
- Step 3. Back-up values from leaves through branch nodes:
  - a Max node computes the Max of its child values
  - a Min node computes the Min of its child values
- Step 4. At root: choose the move leading to the child of highest value
- Minimax values can be found by depth-first game-tree search
- Introduced by Claude Shannon: Programming a Computer for Playing Chess
  - Ran on paper!

pra-sâmi

22

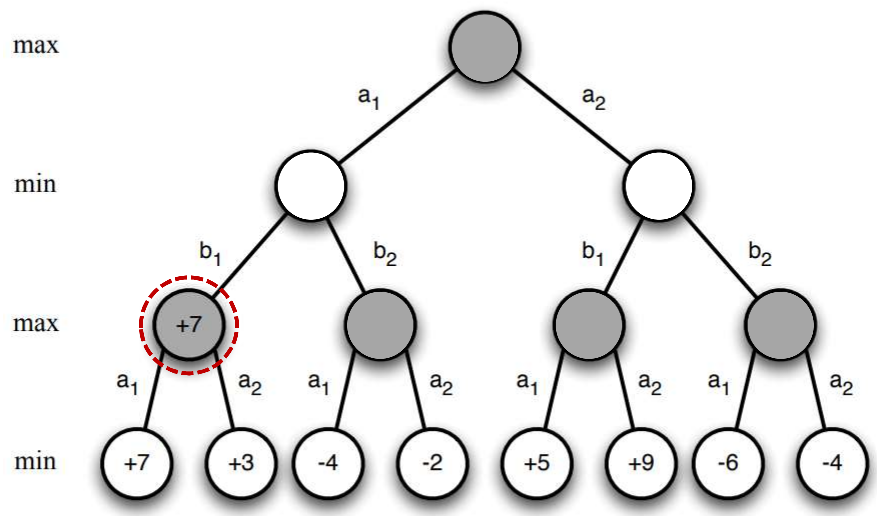
Minimax Search Example



pra-sâmi

23

Minimax Search Example

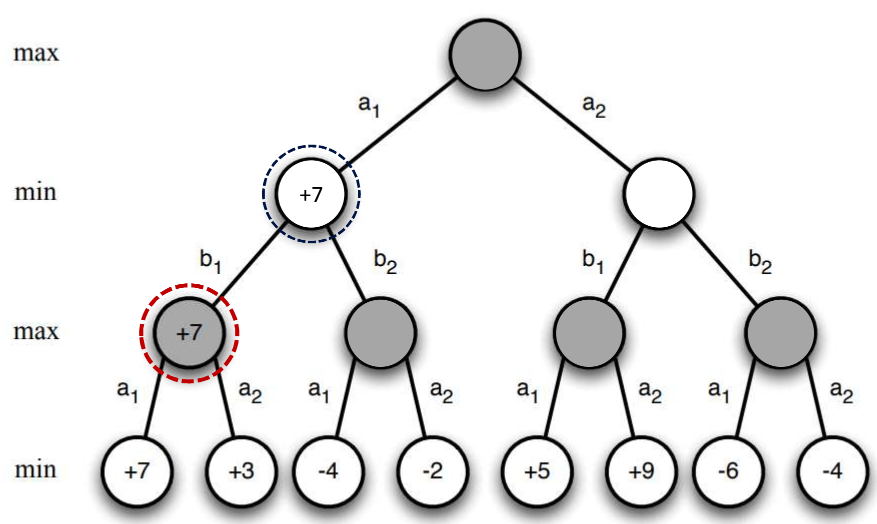


4/13/2024

pra-sâmi

24

Minimax Search Example

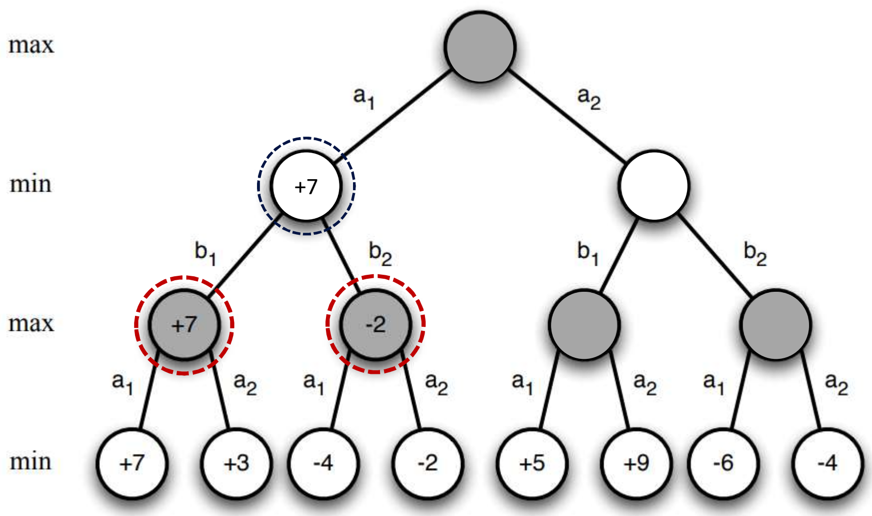


4/13/2024

pra-sâmi

25

Minimax Search Example

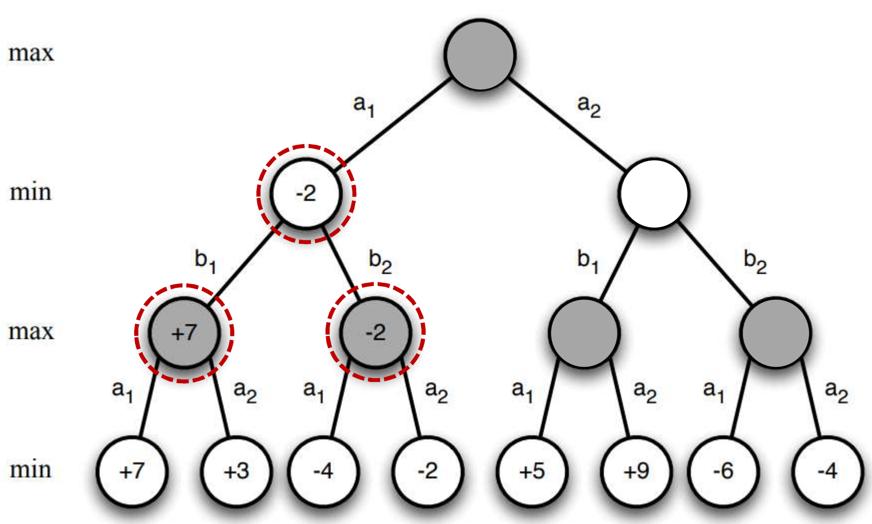


4/13/2024

pra-sâmi

26

Minimax Search Example

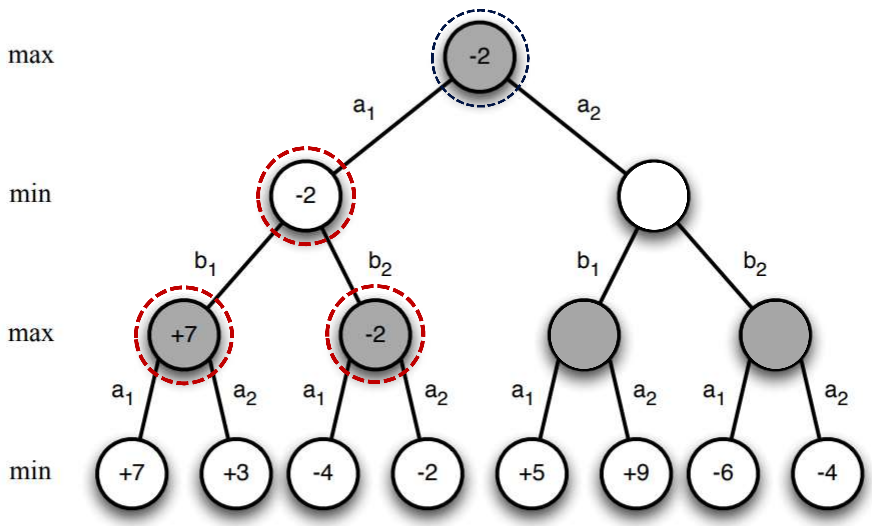


4/13/2024

pra-sâmi

27

Minimax Search Example

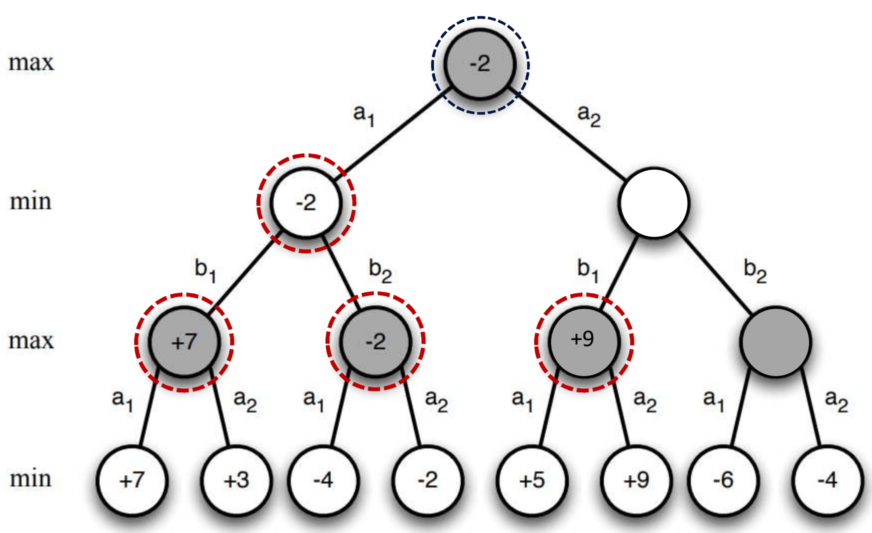


4/13/2024

pra-sâmi

28

Minimax Search Example

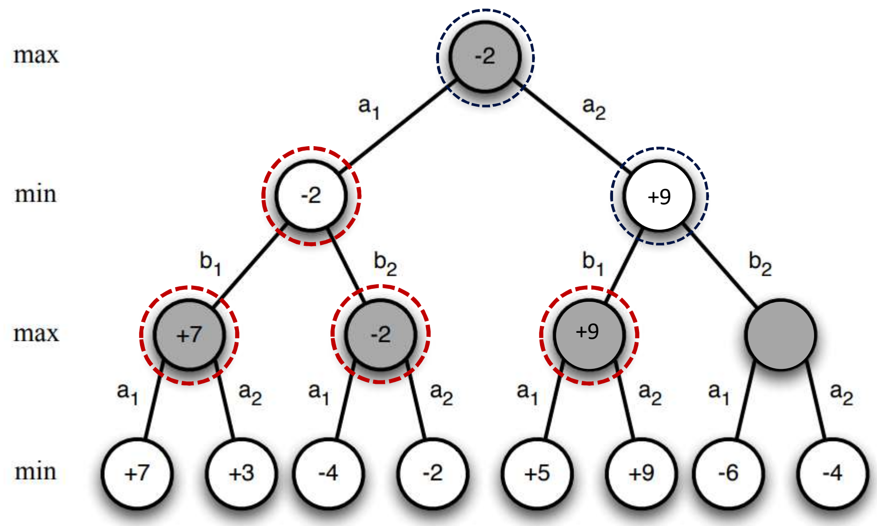


4/13/2024

pra-sâmi

29

Minimax Search Example

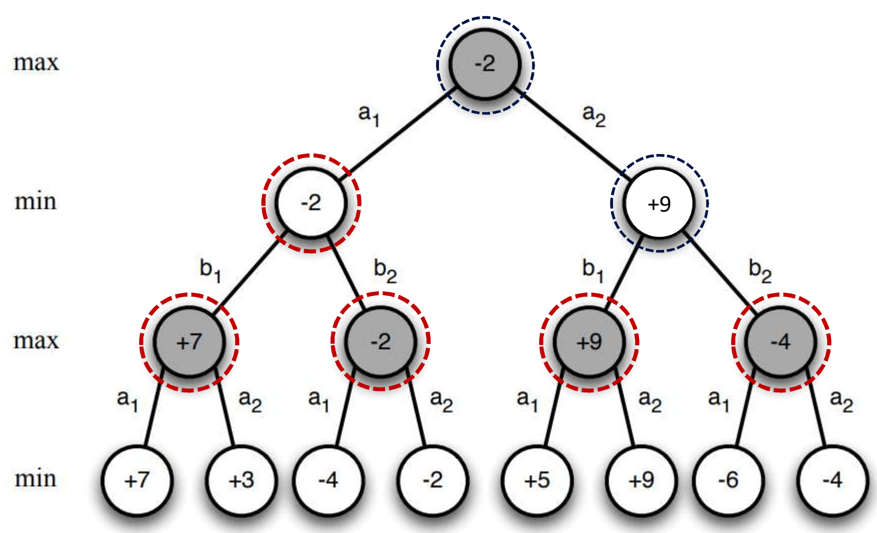


4/13/2024

pra-sâmi

30

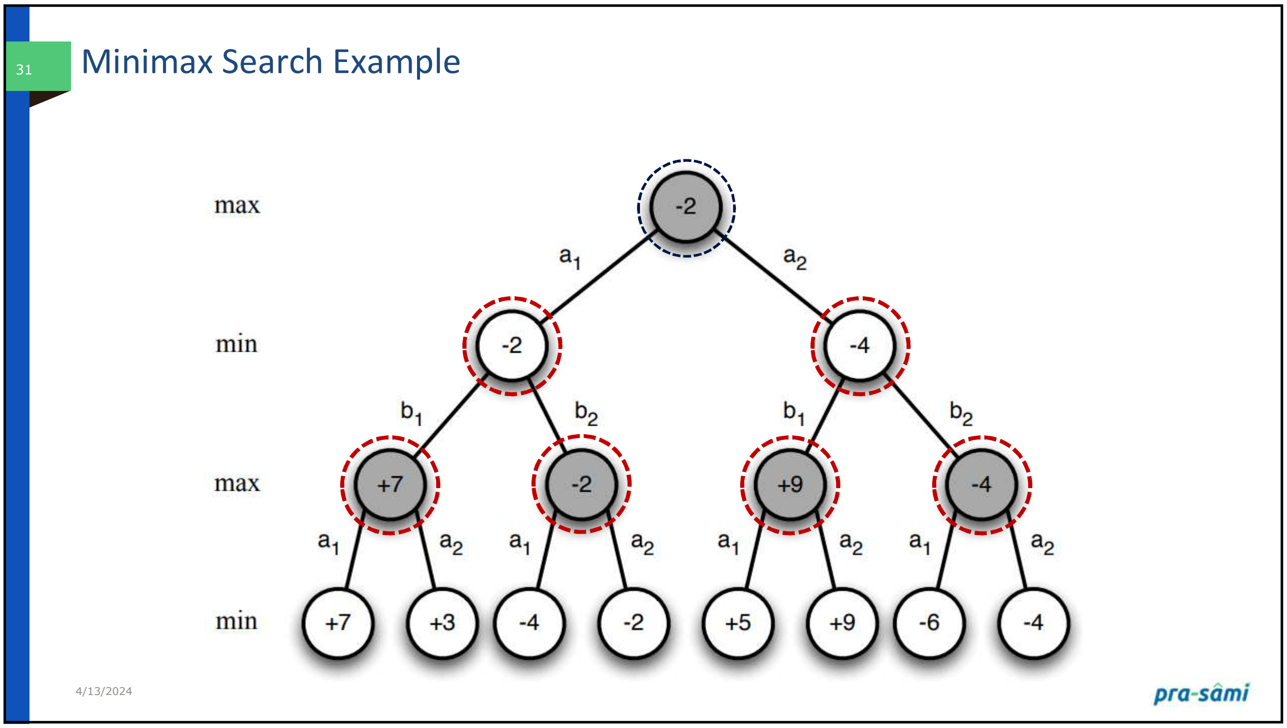
Minimax Search Example



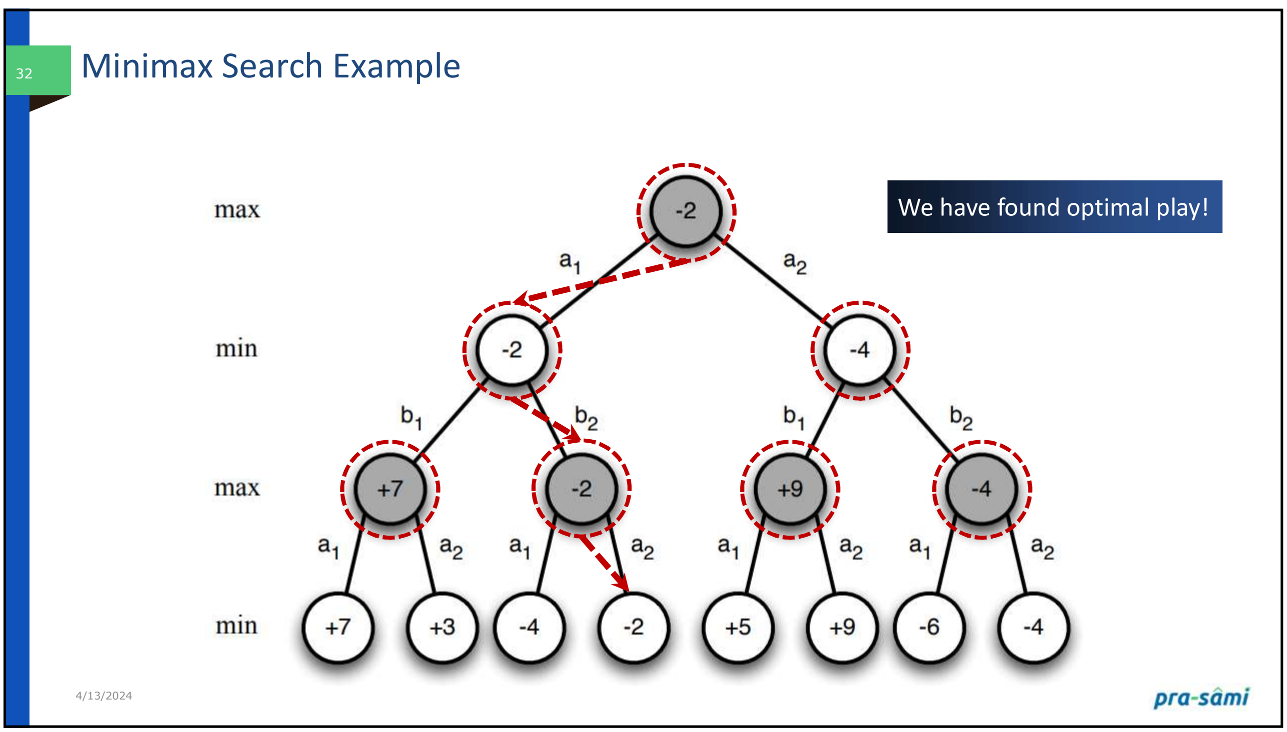
4/13/2024

pra-sâmi

Minimax Search Example



Minimax Search Example



33

Static (Heuristic) Evaluation Functions

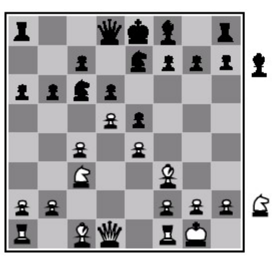
- An Evaluation Function:
  - Estimates how good the current board configuration is for a player.
  - Typically, one figures how good it is for the player, and how good it is for the opponent, and subtracts the opponents score from the players
  - Othello: Number of white pieces - Number of black pieces
  - Chess: Value of all white pieces - Value of all black pieces
- Typical values from -infinity (loss) to +infinity (win) or [-1, +1].
- If the board evaluation is X for a player, it's -X for the opponent.
- Many clever ideas about how to use the evaluation function.
  - e.g. null move heuristic: let opponent move twice.
- Example: Evaluating chess boards, Checkers, Tic-tac-toe

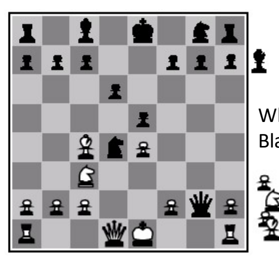
4/13/2024 pra-sâmi

34

Evaluation Function

Black to move
White in better position





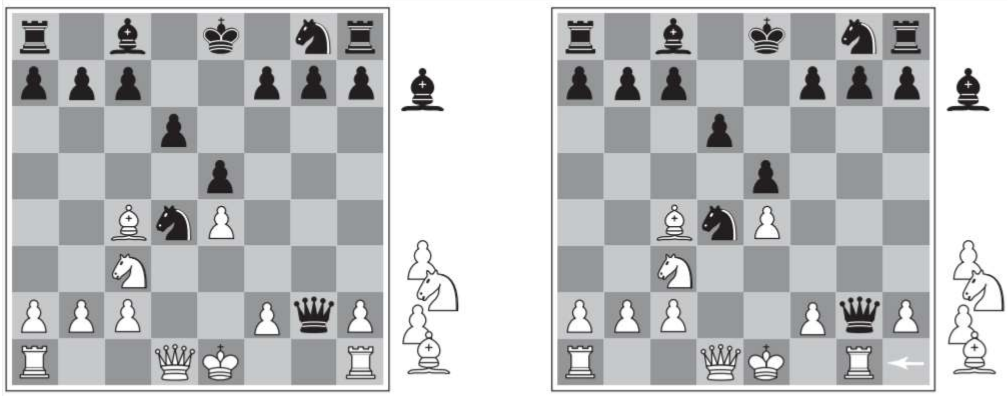
White to move
Black Winning

- For Chess, use linear weighted sum of features
  - E(s) = w_1 x $f_1(s)$ + w_2 x $f_2(s)$ + … + w_n x $f_n(s)$
  - Where w_n is weight assigned to any coin (say w_n = 9 for queen and w_n = 1 for Pawn)
  - $f_n(s)$ = number of white pieces – number of black pieces
    - For 2 white knight and 1 black knight; $f_{knight}(s)$ = 1

4/13/2024 pra-sâmi

35

Evaluation Function



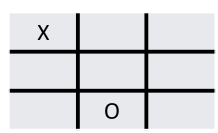
(a) White to move (b) White to move

Two chess positions that differ only in the position of the rook at lower right., In (a), Black has an advantage of a knight and two pawns, which should be enough to win the game, In (b), White will capture the queen, giving it an advantage that should be strong enough to win. Adding up the values of features apparently reasonable, but involves a strong assumption; The contribution of each feature is independent of the values of the other features. For example, assigning the value 3 to a bishop ignores the fact that bishops are more powerful in the endgame, when they have a lot of space to maneuver

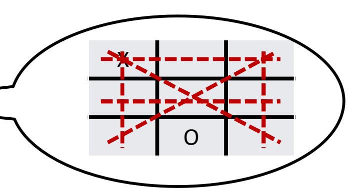
36

Applying Minimax to Tic-tac-toe

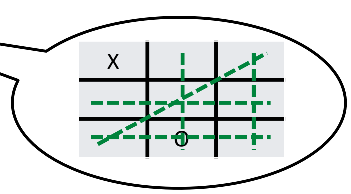
- The static heuristic evaluation function E(n) = M (n) – O (n)
  - M (n) : total of Max's possible winning lines
  - O (n) : total of Min's possible winning lines
  - E(n) : total Evaluation of the State



- X has 6 possible win paths



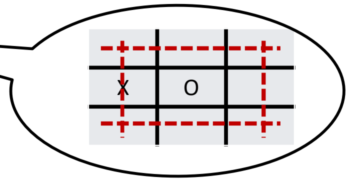
- O has 5 possible win paths
- Therefore E(n) = 6 - 5 = 1



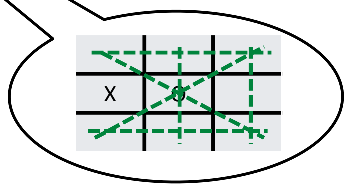
37

Applying Minimax to Tic-tac-toe

- X has 4 possible win paths



- O has 6 possible win paths



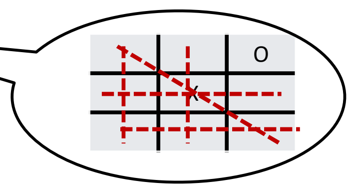
- Therefore E(n) = 4 - 6 = -2

pra-sâmi

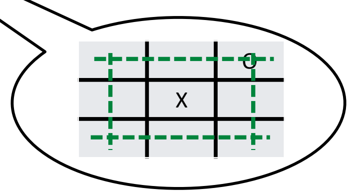
38

Applying Minimax to Tic-tac-toe

- X has 5 possible win paths



- O has 4 possible win paths

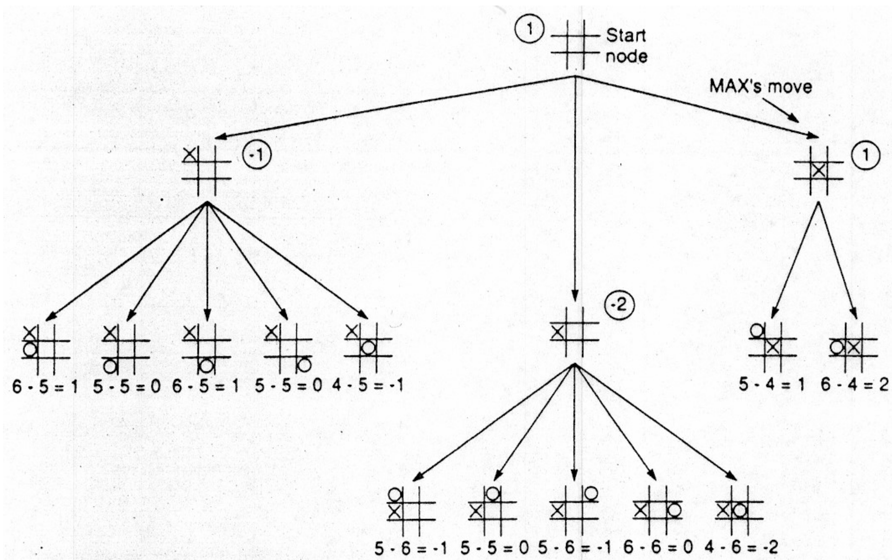


- Therefore E(n) = 5 - 4= 1

pra-sâmi

39

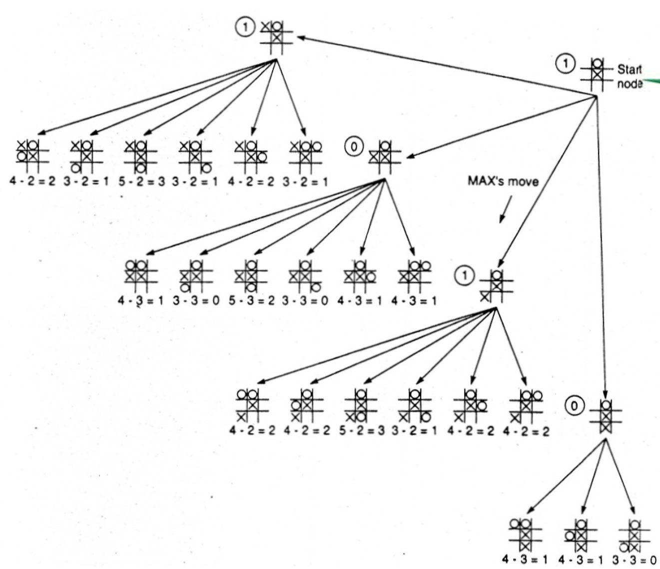
Two-ply Minimax Applied to the Opening Move of Tic-tac-toe



4/13/2024 pra-sâmi

40

Two-ply Minimax Applied to X's Second Move of Tic-tac-toe

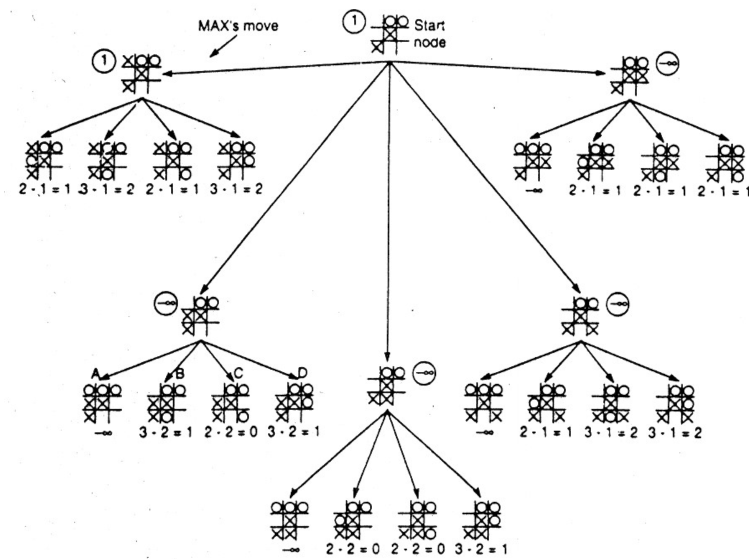


Assumed Min Move

4/13/2024 pra-sâmi

41

Two-ply Minimax Applied to X's Move Near End Game



4/13/2024

pra-sâmi

42

Limitation

Minimax search is suitable for games like tic-tac-toe and not suitable for game of chess.

- Time complexity of the search is O(b^d)
  - b : number of branches
  - d : depth of the tree
- Hence for the example with d = 3
  - Time complexity will be 2^3 = 8
- In regular game of chess ,
  - Average choices are 35,
  - Average game lasts for about 50 moves by a player → Total 100 moves
- Hence complexity will be 35^{100} = 2.5516 X 10^{154}

4/13/2024

pra-sâmi

43

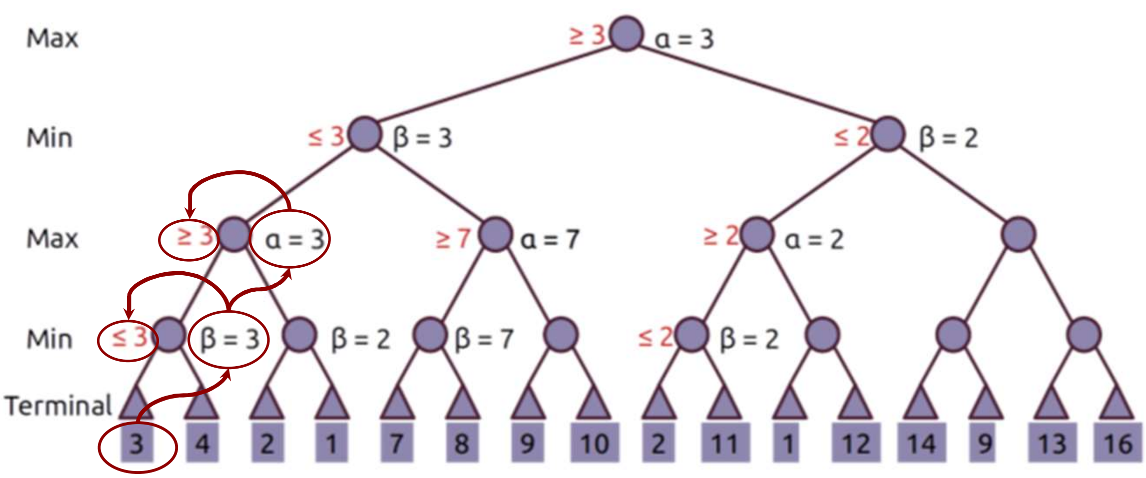
Solution to the Complexity Problem

- Two solutions
- Dynamic pruning of redundant branches
  - Identify a provably suboptimal branch of the search tree before it is fully explored
  - Eliminate the suboptimal branch
  - It is called Alpha – Beta Pruning
  - Some branches will never be played by rational players since they include sub-optimal decisions for either player
- Early cutoff the search tree
- Use imperfect minimax value estimate for non-terminal states (positions)
  - α = highest-value choice that we can guarantee for MAX so far in the current subtree.
  - β= lowest-value choice that we can guarantee for MIN so far in the current subtree.
  - Update values of α and β during search and prunes remaining branches as soon as the value is known to be worse than the current α or β value for MAX or MIN respectively.

4/13/2024 pra-sâmi

44

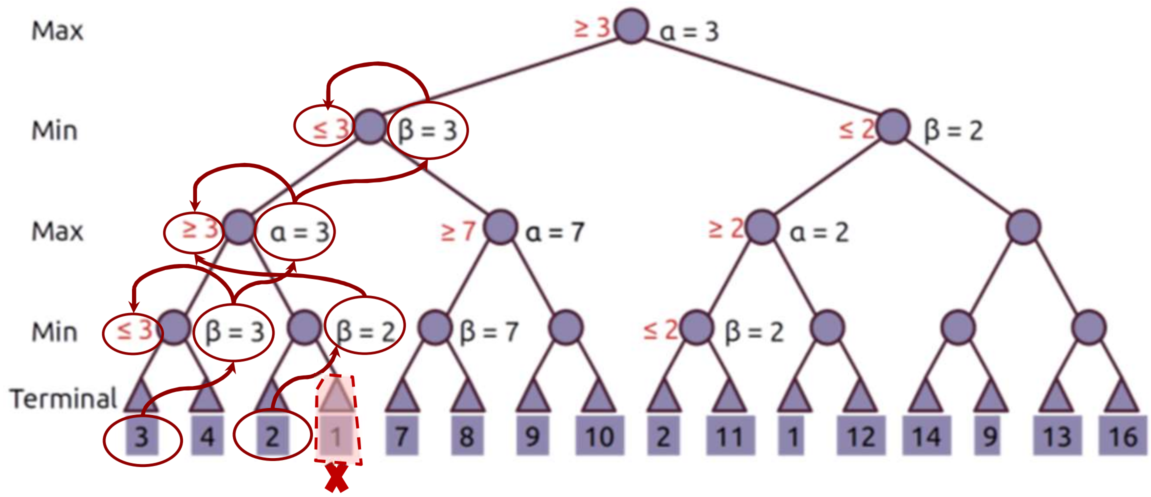
Alpha Beta Pruning



4/13/2024 pra-sâmi

45

Alpha Beta Pruning

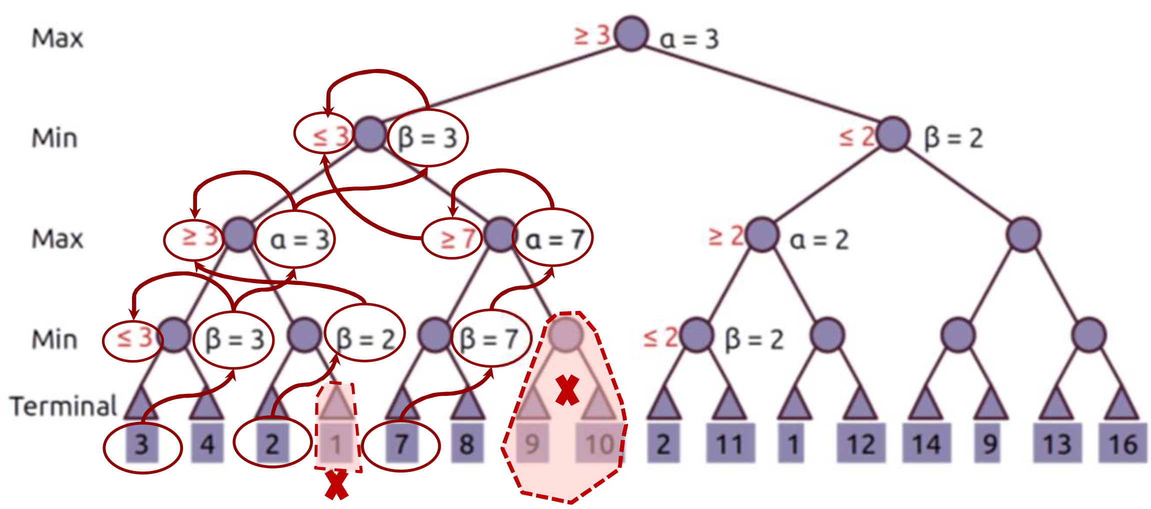


4/13/2024

pra-sâmi

46

Alpha Beta Pruning

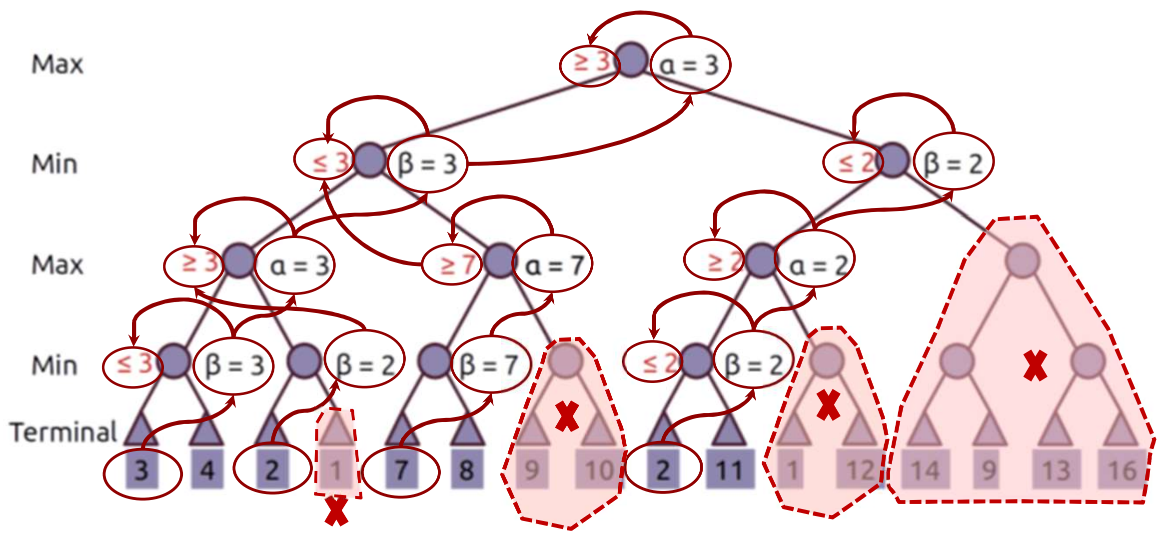


4/13/2024

pra-sâmi

47

Alpha Beta Pruning



pra-sâmi

48

Effectiveness of Alpha-Beta Search

- Worst-Case
  - Branches are ordered so that no pruning takes place
  - Gives no improvement over exhaustive search
- Best-Case
  - Each player’s best move is the left-most child (i.e., evaluated first)
  - In practice, performance is closer to best rather than worst-case
    - E.g., sort moves by the remembered move values found last time
    - E.g., expand captures first, then threats, then forward moves, etc.
    - E.g., run Iterative Deepening search, sort by value last iteration
- In practice often get O($b^{(\frac{d}{2})}$) rather than O(b^d)
  - This is the same as having a branching factor of sqrt(b),
    - $\sqrt{b}^d = b^{(\frac{d}{2})}$,i.e., we effectively go from b to square root of b
  - e.g., in chess go from b ~ 35 to b ~ 6
    - This permits much deeper search in the same amount of time

pra-sâmi

49

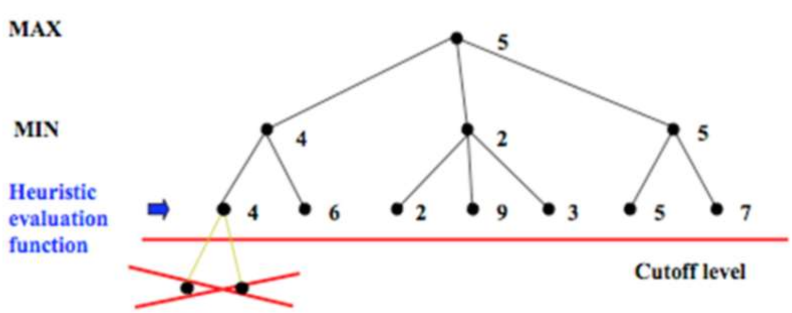
Final Comments about Alpha-Beta Pruning

- Pruning does not affect final results
- Entire subtrees can be pruned
- Good move ordering improves effectiveness of pruning
- Repeated states are again possible
  - Store them in memory = transposition table

50

Early Cutoff the Search Tree

- Using Minimax value estimates
  - Cutoff the search tree before the terminal state is reached
  - Use imperfect estimate of the minimax value at the leaves
    - Evaluation function



51

Practical Implementation

- How do we make these ideas practical in real game trees?
- Standard approach:
  - Cutoff test: (where do we stop descending the tree)
    - Depth limit
    - Better: iterative deepening
    - Cutoff only when no big changes are expected to occur next
  - Evaluation function
    - When the search is cut off, we evaluate the current state by estimating its utility using an evaluation function.

52

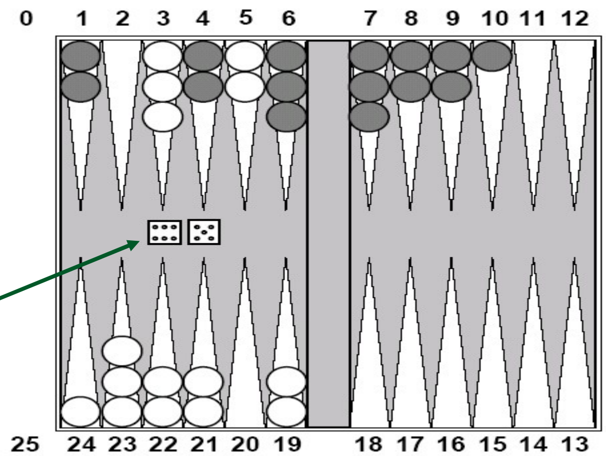
Iterative (Progressive) Deepening

- In real games, there is usually a time limit T on making a move
- How do we take this into account?
  - Using alpha-beta we cannot use "partial" results with any confidence unless the full breadth of the tree has been searched
  - So, we could be conservative and set a conservative depth-limit which guarantees that we will find a move in time < T
    - Disadvantage is that we may finish early, could have done more search
- In practice, iterative deepening search (IDS) is used
  - IDS runs depth-first search with an increasing depth-limit
  - When the clock runs out we use the solution found at the previous depth limit

53

Chance Games

Backgammon



Element of chance

4/13/2024

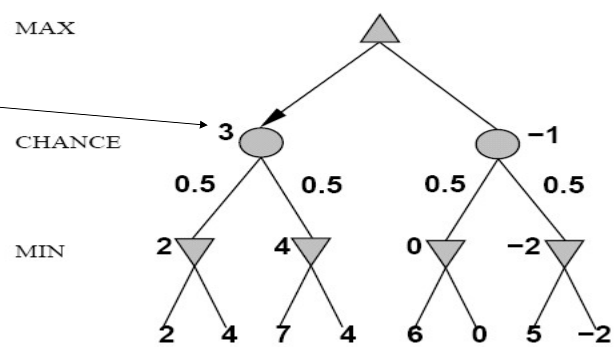
pra-sâmi

54

Expected Minimax

$$v = \sum_{chance\ nodes} P(n) \times Minimax(n)$$

$$3 = 0.5 \times 4 + 0.5 \times 2$$



Interleave chance nodes
with min/max nodes

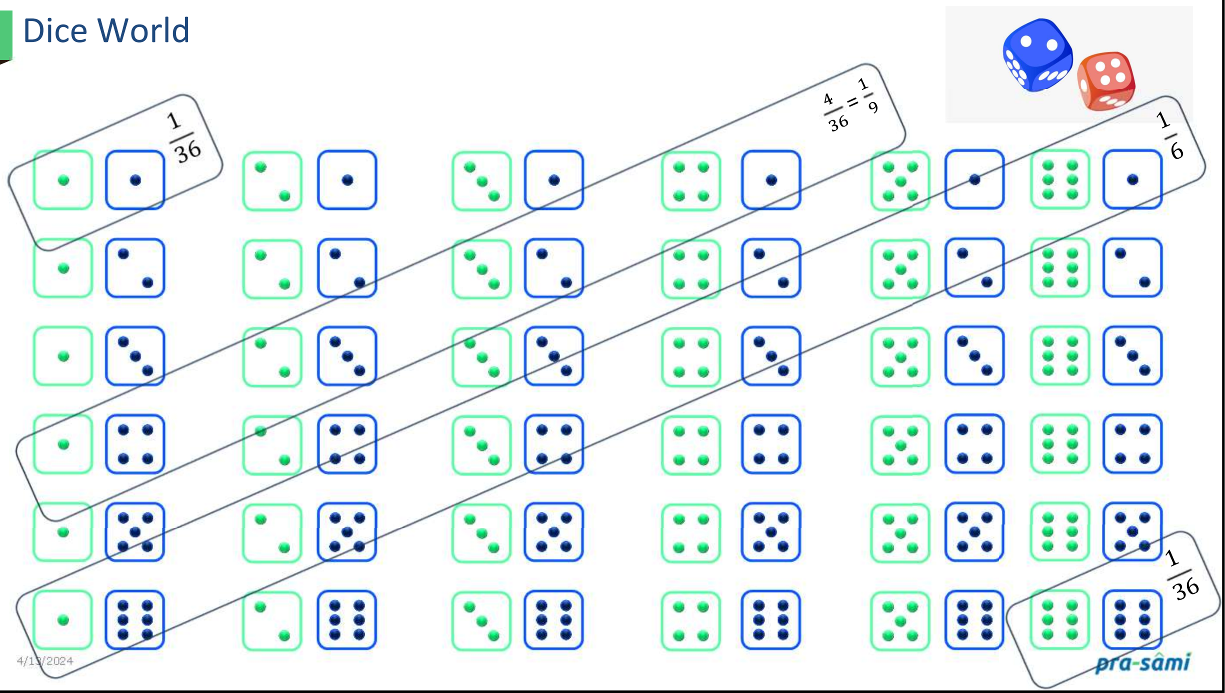
Again, the tree is constructed
bottom-up

4/13/2024

pra-sâmi

55

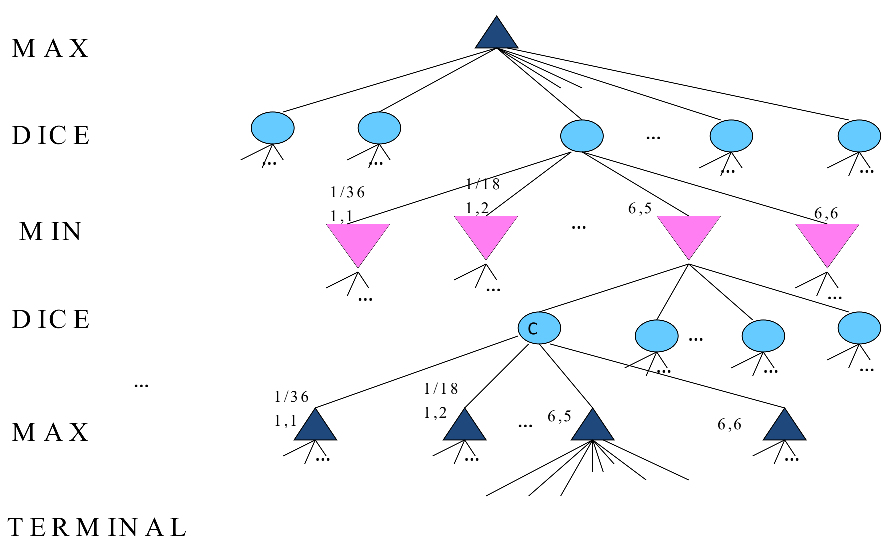
Dice World



4/13/2024 pra-sâmi

56

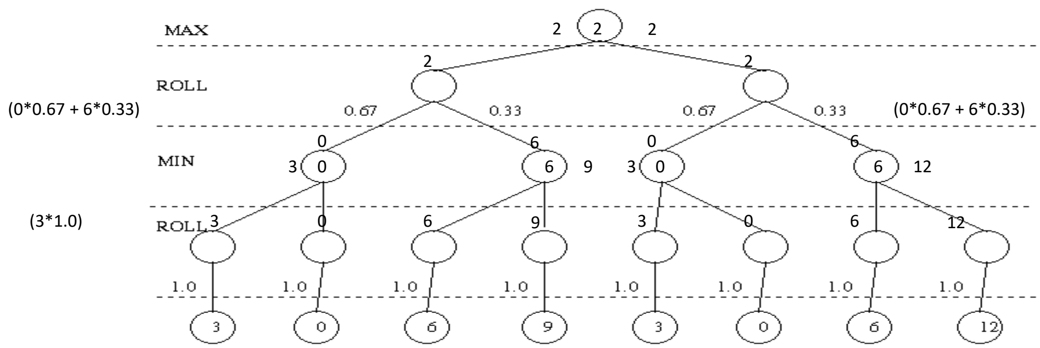
Game Tree for Backgammon



4/13/2024 pra-sâmi

58

Expectiminimax Example

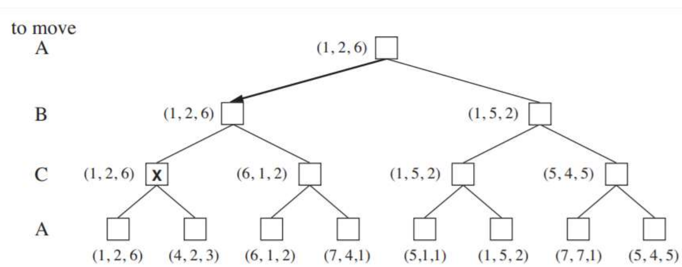


4/13/2024

pra-sâmi

59

Optimal Decisions in Multiplayer Games



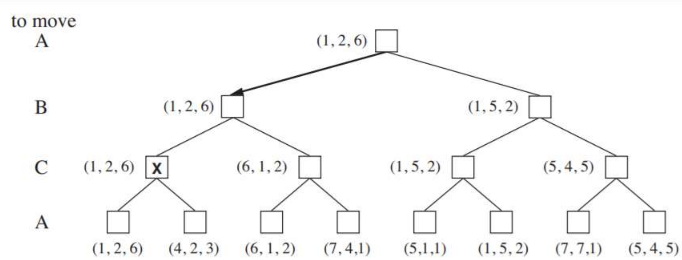
- A vector of values instead of a single value for each node
- Multiplayer games usually involve alliances, among the players
  - May be formal or informal
- Alliances are made and broken as the game proceeds
- Suppose A and B are in weak positions and C is in a stronger position
- Then it is often optimal for both A and B to attack C rather than each other
  - Collaboration emerges from purely selfish behavior

4/13/2024

pra-sâmi

60

Optimal Decisions in Multiplayer Games



- Of course, as soon as C weakens under the joint onslaught, the alliance loses its value
- Either A or B could violate the agreement
- Explicit alliances, if any, merely make concrete what would have happened anyway
- In some cases, a social stigma attaches to breaking an alliance
  - Immediate advantage of breaking an alliance vs the long-term disadvantage of being perceived as untrustworthy

4/13/2024

pra-sâmi

61

Nonzero Sum Game Trees

- The idea of “look ahead, reason backward” works for any game tree with perfect information.
  - I.e., also in cooperative games
- In AI, this is called retrograde analysis.
- In game theory, it is called backward induction or subgame perfect equilibrium.
- Can be extended to many games with imperfect information (sequential equilibrium).

4/13/2024

pra-sâmi

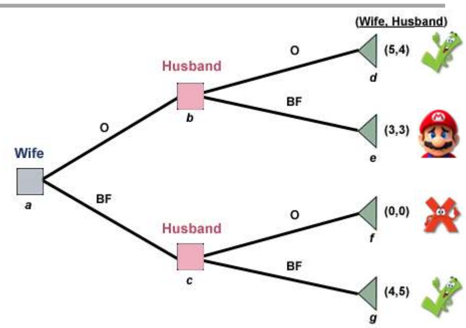
62

A Bullfight or the Opera

- Background
  - A husband and wife are deciding whether to spend the evening at a Bullfight or the Opera.
  - The husband prefers the bullfight to the opera, but the wife prefers the opera to the bullfight.
  - Both would prefer to be together at either event rather than spending the evening apart.
  - The worst outcome would be for the husband to spend the evening alone at the opera and the wife to spend the evening alone at the bullfight

- This game draws attention to the problem of co-ordination
  - Players have divergent preferences
  - A common interest in coordinating their strategies.

		Husband	
		O	BF
Wife	0	5, 4	3, 3
	BF	0, 0	4, 5



4/13/2024

63

Compare to "Prisoner's Dilemma"

- "Deterministic, NON-turn-taking, NON-zero-sum game of Imperfect information"

A prisoner's dilemma is a situation where individual decision-makers always have an incentive to choose in a way that creates a less than optimal outcome for the individuals as a group.

There are methods of overcoming prisoner's dilemmas to choose better collective results despite apparently unfavorable individual incentives.

- Dave and Henry - Two members of a gang of bank robbers; arrested ; being interrogated in separate rooms; No other witnesses; need confession
- Be loyal and remain silent; the authorities will only be able to convict them on a lesser charge
  - One year in jail for each (1 year for Dave + 1 year for Henry = 2 years total jail time)
- If one testifies and the other does not
  - The one who testifies will go free and the other will get five years (0 years for the one who defects + 5 for the one convicted = 5 years total)
- If both testify against the other
  - Each will get two years in jail for being partly responsible for the robbery (2 years for Dave + 2 years for Henry = 4 years total jail time)

4/13/2024

pra-sâmi

64

Win as much as you can!

- Four Teams, choose either X or Y
- Pay off Schedule
  - 4 X's:
    - Lose $ 1.00 each
  - 3 X's, 1 Y :
    - Xs: Win $1.00 each; Ys : Lose $3.00
  - 2 X's, 2 Y's :
    - Xs: Win $ 2.00 each; Ys : Lose $ 2.00 each
  - 1 X , 3 Y's:
    - Xs : Win $ 3.00; Ys: Lose $ 1.00 each
  - 4 Y's:
    - Win $ 1.00 each

	Round	Your Choice	Group's pattern of Choice	Payoff	Balance
	1	X Y	_X _Y		
	2	X Y	_X _Y		
	3	X Y	_X _Y		
	4	X Y	_X _Y		
Bonus Round (Payoff X 3)	5	X Y	_X _Y		
	6	X Y	_X _Y		
Leader's Conference	7	X Y	_X _Y		
Bonus Round (Payoff X 5)	8	X Y	_X _Y		
	9	X Y	_X _Y		
Bonus Round (Payoff X 10)	10	X Y	_X _Y		

65

Summary

- Game playing can be effectively modeled as a search problem
- Game trees represent alternate computer/opponent moves
- Evaluation functions estimate the quality of a given board configuration for the Max player.
- Minimax is a procedure which chooses moves by assuming that the opponent will always choose the move which is best for them
- Alpha-Beta is a procedure which can prune large parts of the search tree and allow search to go deeper
- For many well-known games, computer algorithms based on heuristic search match or out-perform human world experts.

66

Summary

- Game systems rely heavily on
  - Search techniques
  - Heuristic functions
  - Bounding and pruning techniques
  - Knowledge database on game
- For AI, the abstract nature of games makes them a ... ct for study:
  - State of the game is easy to represent;
  - Agents are usually restricted to a small ... whose outcomes are defined by precise rules
- Game playing was one of the ... undertaken in AI as soon as computers became programmable (e.g., Turing, Sh..., Wiener tackled chess).
- Game playing research has spawned a number of interesting research ideas on search, data structures, databases, heuristics, evaluations functions and many areas of computer science.

Games are fun:
Teach your computer how to play a game!

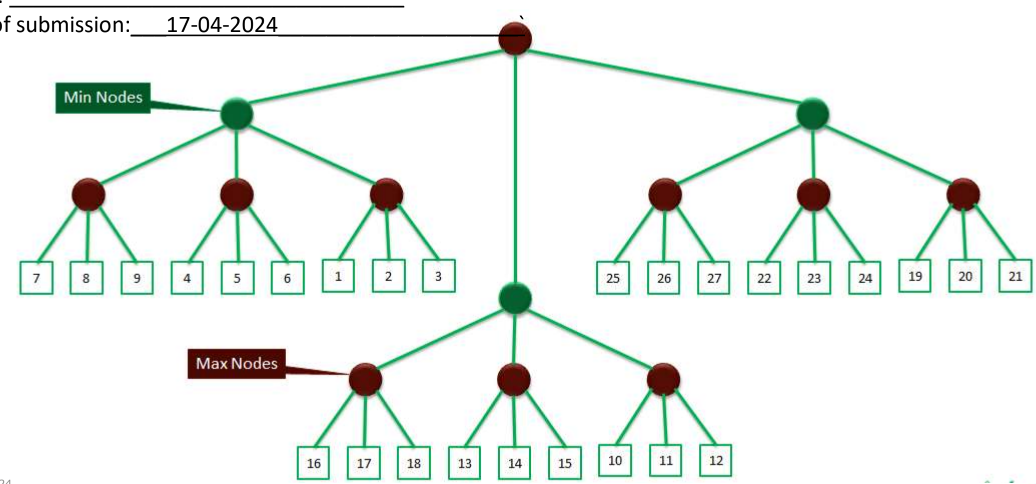
pra-sâmi

67

PS_03: Alpha Beta Pruning

Course & Batch : ______________________
Roll No: ______________________________
Name: ________________________________
Date of submission:___17-04-2024___________________

Task:
- Perform alpha-beta pruning of following tree.
- Explain benefit you achieved over exhaustive search.



pra-sâmi

68

Reflect…

- Which search methodology may not necessarily guarantee the optimal solution but is generally faster in practice?
  A. Uninformed search
  B. Informed search
  C. Heuristic search
  D. Depth-first search
- Ans: C. Heuristic search
- Which classical search algorithm can get trapped in infinite loops if cycles are present in the search space?
  A. Depth-first search
  B. Breadth-first search
  C. A* search algorithm
  D. Uniform-cost search
- Ans: A. Depth-first search
- In parallel search, what is the primary challenge associated with coordinating multiple processors or threads?
  A. Load balancing
  B. Increased memory usage
  C. Decreased search space
  D. Slower execution time
- Ans: A. Load balancing
- Which component of a search engine is responsible for determining the relevance of web pages to a given query?
  A. Indexing
  B. Query processing
  C. Ranking algorithms
  D. User authentication
- Ans: C. Ranking algorithms

4/13/2024 pra-sâmi

69

Reflect…

- Which adversarial search algorithm uses a minimax strategy to make decisions?
  A. Alpha-beta pruning
  B. Monte Carlo Tree Search (MCTS)
  C. Minimax
  D. Expectimax
- Ans: C. Minimax
- Which type of search can be visualized as exploring a tree structure representing all possible states?
  A. Informed searc
  B. Uninformed search
  C. Adversarial search
  D. Heuristic search
- Ans: B. Uninformed search
- Which classical search algorithm expands the node with the lowest path cost g(n) plus an estimated cost h(n) to the goal?
  A. Depth-first search
  B. Breadth-first search
  C. A* search algorithm
  D. Uniform-cost search
- Ans: C. A* search algorithm
- In parallel search, what is a potential drawback of using a large number of processors or threads?
  A. Increased memory usage
  B. Higher energy consumption
  C. Slower execution time
  D. Difficulty in implementation
- Ans: B. Higher energy consumption

4/13/2024 pra-sâmi

70

Reflect…

- Which algorithm is commonly used to solve CSPs by systematically searching through the problem space?
  A. A* algorithm
  B. Depth-First Search (DFS)
  C. Breadth-First Search (BFS)
  D. Hill Climbing
- Ans: B) Depth-First Search (DFS)
- In combinatorial optimization problems, what is the objective?
  A. To find a feasible solution that satisfies a set of constraints
  B. To find the solution that maximizes a certain criterion
  C. To minimize the number of constraints
  D. To randomly explore the search space
- Ans: B) To find the solution that maximizes a certain criterion
- Which of the following is an example of a combinatorial optimization problem?
  A. Sorting
  B. Linear Regression
  C. Traveling Salesman Problem
  D. Binary Search
- Ans: C) Traveling Salesman Problem
- Which algorithm is commonly used to solve combinatorial optimization problems by iteratively improving a candidate solution?
  A. Simulated Annealing
  B. Genetic Algorithm
  C. Particle Swarm Optimization
  D. Hill Climbing
- Ans: D) Hill Climbing

4/13/2024

pra-sâmi

71

Reflect…

- Which of the following techniques is NOT typically used in Constraint Satisfaction Search?
  A. Forward checking
  B. Genetic Algorithm
  C. Arc consistency
  D. Backtracking
- Ans: B) Genetic Algorithm
- What does the term "arc consistency" refer to in Constraint Satisfaction Problems?
  A. It ensures that all variables in a CSP are assigned a value
  B. It prunes inconsistent values from the domains of variables
  C. It involves traversing the graph representing the CSP
  D. It is a heuristic function used in search algorithms
- Ans: B) It prunes inconsistent values from the domains of variables
- Which of the following is a limitation of the hill climbing algorithm in combinatorial optimization?
  A. It always finds the global optimum
  B. It is susceptible to getting stuck in local optima
  C. It is computationally expensive
  D. It requires complete knowledge of the search space
- Ans: B) It is susceptible to getting stuck in local optima
- Which of the following is NOT a step in the typical problem-solving process for Constraint Satisfaction Search?
  A. Formulating the problem as a CSP
  B. Propagating constraints to reduce the search space
  C. Randomly assigning values to variables
  D. Searching for a solution using a suitable algorithm
- Ans: C) Randomly assigning values to variables

4/13/2024

pra-sâmi

72

Reflect…

- Which algorithm is commonly used to solve CSPs by systematically searching through the problem space?
  A. A* algorithm
  B. Depth-First Search (DFS)
  C. Breadth-First Search (BFS)
  D. Hill Climbing

- Ans: B) Depth-First Search (DFS)

- In combinatorial optimization problems, what is the objective?
  A. To find a feasible solution that satisfies a set of constraints
  B. To find the solution that maximizes a certain criterion
  C. To minimize the number of constraints
  D. To randomly explore the search space

- Ans: B) To find the solution that maximizes a certain criterion

- Which of the following is an example of a combinatorial optimization problem?
  A. A) Sorting
  B. B) Linear Regression
  C. C) Traveling Salesman Problem
  D. D) Binary Search

- Ans: C) Traveling Salesman Problem

- Which algorithm is commonly used to solve combinatorial optimization problems by iteratively improving a candidate solution?
  A. A) Simulated Annealing
  B. B) Genetic Algorithm
  C. C) Particle Swarm Optimization
  D. D) Hill Climbing

- Ans: D) Hill Climbing

4/13/2024

pra-sâmi

73

Reflect…

- Which of the following techniques is NOT typically used in Constraint Satisfaction Search?
  A. Forward checking
  B. Genetic Algorithm
  C. Arc consistency
  D. Backtracking

- Ans: B) Genetic Algorithm

- What does the term "arc consistency" refer to in Constraint Satisfaction Problems?
  A. It ensures that all variables in a CSP are assigned a value
  B. It prunes inconsistent values from the domains of variables
  C. It involves traversing the graph representing the CSP
  D. It is a heuristic function used in search algorithms

- Ans: B) It prunes inconsistent values from the domains of variables

- Which of the following is a limitation of the hill climbing algorithm in combinatorial optimization?
  A. It always finds the global optimum
  B. It is susceptible to getting stuck in local optima
  C. It is computationally expensive
  D. It requires complete knowledge of the search space

- Ans: B) It is susceptible to getting stuck in local optima

- Which of the following is NOT a step in the typical problem-solving process for Constraint Satisfaction Search?
  A. Formulating the problem as a CSP
  B. Propagating constraints to reduce the search space
  C. Randomly assigning values to variables
  D. Searching for a solution using a suitable algorithm

- Ans: C) Randomly assigning values to variables

4/13/2024

pra-sâmi

75

Next Session

- Introduction to Data Analytics
- Type of Data
- Introduction to Big Data
- Hadoop Eco System
- Tasks of AI Using Big Data

4/13/2024

pra-sâmi

76

4/13/2024

pra-sâmi

EXTRA MATERIAL

State of Games

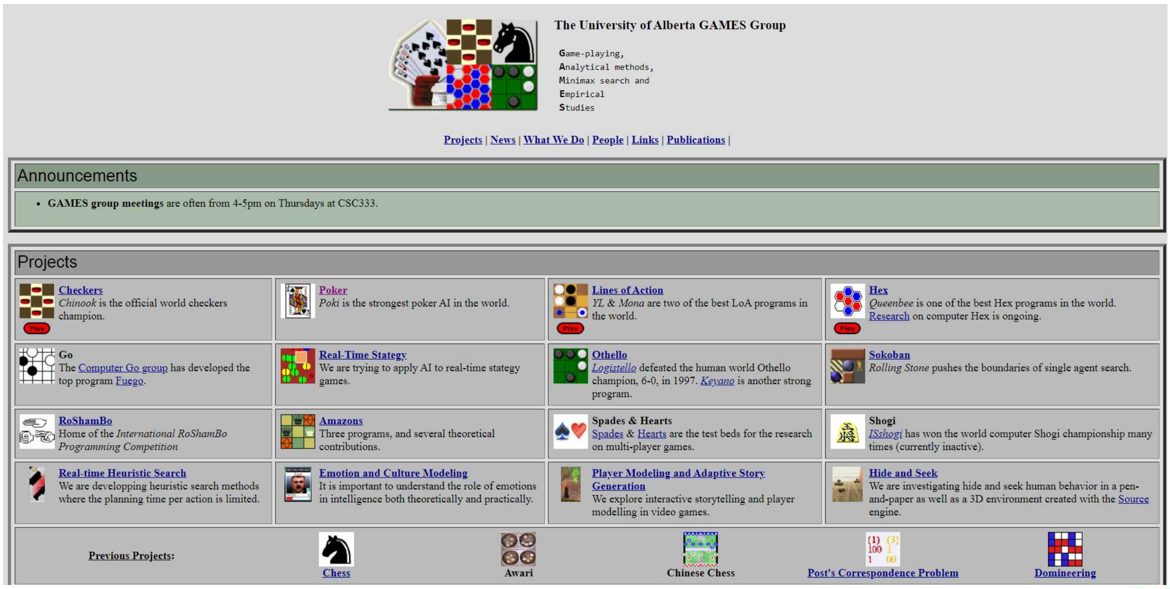
78

The State of Play

- Checkers:
  - Chinook ended 40-year-reign of human world champion Marion Tinsley in 1994.
- Chess:
  - Deep Blue defeated human world champion Garry Kasparov in a six-game match in 1997.
- Othello:
  - Human champions refuse to compete against computers: they are too good.
- Go:
  - Human champions refuse to compete against computers: they are too bad b > 300 (!)

79

The University of Alberta; CA



pra-sâmi

80

Deep Blue

- 1957: Herbert Simon
  - "within 10 years a computer will beat the world chess champion"
- 1997: Deep Blue beats Kasparov
- Parallel machine with 30 processors for "software" and 480 VLSI processors for "hardware search"
- Searched 126 million nodes per second on average
  - Generated up to 30 billion positions per move
  - Reached depth 14 routinely
- Uses iterative-deepening alpha-beta search with transpositioning
  - Can explore beyond depth-limit for interesting moves

pra-sâmi