

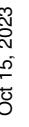


IS/SE/SS ZC444: Artificial Intelligence

Og Beyond Classical Search Methods

Dr. Kamlesh Tiwari

Associate Professor, Department of CSE,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA



Oct 15, 2023

ONLINE

WILP @ BITS-Pilani [July-Dec 2023]

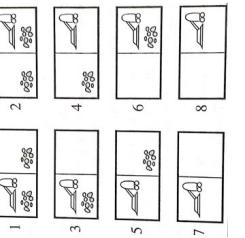
<http://ktiwari.in/ai>

Search with Non-Deterministic Actions

Non-Deterministic: not sure what would be the next state¹

Consider erratic vacuum world

sometime 1) also cleans neighboring room 2) deposit dirt



¹Percepts would tell where have we reached.

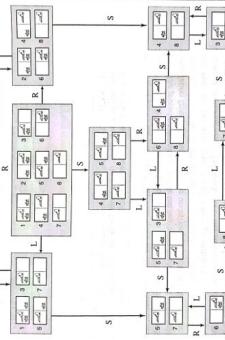
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 3/23

Searching with Partial Observations

When percepts do not suffice to pin down the exact state

- Sensor less. consider [right,suck,left,suck] guarantees to reach in state 8 that is a goal state (traverses through belief states)
- All possible belief states may not be reachable (only 12 out of 28)

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 3/23



Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 5/23

Local Search in Continuous State

Issue: Number of next states (branching factor) becomes infinite

Example: Induct three new airports in Romania

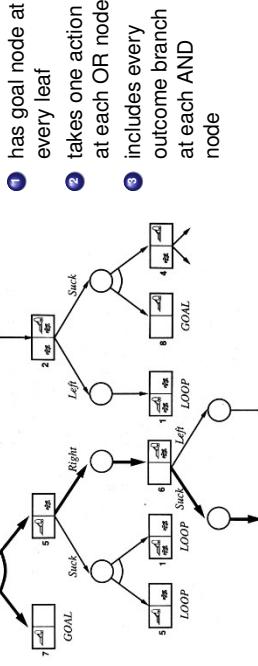
- Let at $(x_1, y_1), (x_2, y_2)$ and (x_3, y_3) on the map
- Minimize sum of distances of all the cities from its nearest airport

$$f(x_1, y_1, x_2, y_2, x_3, y_3) = \sum_{i=1}^3 \sum_{c \in C_i} ((x_i - x_c)^2 + (y_i - y_c)^2)$$

- If you discretize the neighborhood then there are only 12 next state (move only $\pm\delta$ in one step). One can apply hill climbing.
- If you attempt to use gradient $\nabla f = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial y_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial y_2}, \frac{\partial f}{\partial x_3}, \frac{\partial f}{\partial y_3})$ it cannot be solved as **globally** finding ∇f is not possible.
- Given **locally** correct values of $\frac{\partial f}{\partial x_i} = 2 \sum_{c \in C_i} (x_i - x_c)$ one can perform steepest-ascent using $x \leftarrow x + \alpha \nabla f$

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 2/23

AND-OR Search Tree

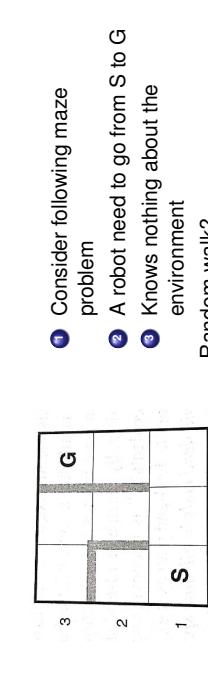


Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 4/23

Online Search and Unknown Environment

Agent interleaves computation and action

- Take action → observe environment → compute next action
- Online Search is necessary for unknown environment



No algorithm can avoid dead-end in all state space
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 6/23

Adversarial Search (game)

Agents having conflicting goals in competitive multiagent environment

- Deterministic, fully-observable, turn-taking, two-player, zero-sum
- Chess has roughly branching factor 35, moves 50 so tree search space is $35^{100} = 10^{154}$ however, graph has 10^{40} nodes
- Finding optimal move is infeasible but, needs an ability to decide

Game is between MAX and MIN (MAX moves first)

- S_0 : the initial state
- $\text{PLAYER}(s)$: defines which player has move to start
- $\text{ACTIONS}(s)$: returns set of legal moves in a state
- $\text{RESULT}(s, a)$: termination model defining result of a move
- $\text{TERMINAL_TEST}(s)$: is true when game is over
- $\text{UTILITY}(s, p)$: utility function defining reward (for chess +1,0,-1/2)

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 7/23

Two half moves is one ply

Given the game tree, optimal strategy can be determined from **minimax value** of each node.



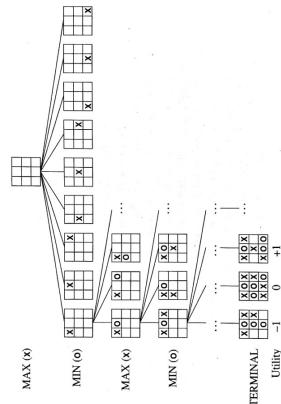
$$\text{MINIMAX}(s) = \begin{cases} \text{UTILITY}(s) & \text{if TERMINAL_TEST}(s) \\ \underset{a \in \text{Actions}(s)}{\text{argmax}} \text{MINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{PLAYER}(s) = \text{MAX} \\ \underset{a \in \text{Actions}(s)}{\text{argmin}} \text{MINIMAX}(\text{RESULT}(s, a)) & \text{if } \text{PLAYER}(s) = \text{MIN} \end{cases}$$

Action a_1 is the optimal choice ²
(essentially optimizing worst-case outcome for MAX)

²utility value for MAX of being in corresponding state (assuming then onwards both player play optimally)

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 9/23

Game Tree for tic-tac-toe



The search tree of the game has less than $9! = 362880$ nodes.

MAX must find a contingent **strategy**.

Analogous to AND-OR search (MAX plays OR and MIN plays AND)

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 8/23

MINIMAX Algorithm

Returns the action corresponding to best move

```
function MINIMAX-DECISION(state) returns an action
    return argmaxa ∈ ACTIONS(s) MIN-VALUE(RESULT(state, a))

function MIN-VALUE(state) returns a utility value
    if TERMINAL-TEST(state) then return UTILITY(state)
    v ← -∞
    for each a in ACTIONS(state) do
        v ← MAX(v, MIN-VALUE(RESULT(s, a)))
    return v

function MAX-VALUE(state) returns a utility value
    if TERMINAL-TEST(state) then return UTILITY(state)
    v ← ∞
    for each a in ACTIONS(state) do
        v ← MIN(v, MAX-VALUE(RESULT(s, a)))
    return v
```

Recursion proceeds all the way down to the leaves. Time complexity $O(b^m)$ that is impractical but provides a basis of solution

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 10/23

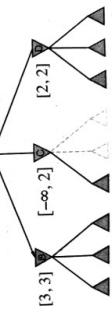
ALPHA-BETA Pruning

- Number of nodes to examine in minimax search is exponential in the depth of tree $O(b^m)$.
- Sometime we can make it $O(b^{m/2})$ using **alpha-beta pruning**
- When applied to standard minimax tree, it returns the same move as minimax but, prunes away branches that cannot possibly influence the decision.

Consider two unevaluated successors of node C have value x and y

$$\begin{aligned} \text{MINIMAX}(\text{root}) &= \max(\text{min}(3, 2), \text{min}(2, 8), \text{min}(14, 5, 2)) \\ &= \max(3, \text{min}(2, 8), 2) \\ &= \max(3, 2) \\ &= 3 \end{aligned}$$

where $z = \min(2, x, y) \leq 2$



Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 11/23

$\alpha =$ value of best choice (highest)	$\beta =$ value of best choice (lowest)
$\text{MINIMAX}(\text{root})$ $= \max(\text{min}(3, 2), \text{min}(2, 8), \text{min}(14, 5, 2))$ $= \max(3, \text{min}(2, 8), 2)$ $= \max(3, 2)$ $= 3$	$\text{MINIMAX}(\text{root})$ $= \max(\text{min}(3, 2), \text{min}(2, 8), \text{min}(14, 5, 2))$ $= \max(3, \text{min}(2, 8), 2)$ $= \max(3, 2)$ $= 3$

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 12/23

ALPHA-BETA Pruning

```

function ALPHA-BETA-SEARCH(s(state)) returns an action
    v  $\leftarrow$  MAX-VALUE(s(state),  $-\infty$ ,  $+\infty$ )
    return the action in ACTIONS(s(state)) with value v

function MAX-VALUE(s(state),  $\alpha, \beta$ ) returns a utility value
    if TERMINAL-TEST(s(state)) then return UTILITY(s(state))
    v  $\leftarrow -\infty$ 
    for each a in ACTIONS(s(state)) do
        v  $\leftarrow$  MAX(v, MIN-VALUE(RESULT(s,a),  $\alpha, \beta$ ))
        if v  $\geq \beta$  then return v
         $\alpha \leftarrow \text{MAX}(\alpha, v)$ 
    return v

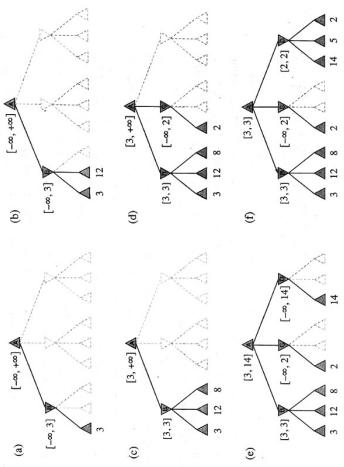
function MIN-VALUE(s(state),  $\alpha, \beta$ ) returns a utility value
    if TERMINAL-TEST(s(state)) then return UTILITY(s(state))
    v  $\leftarrow +\infty$ 
    for each a in ACTIONS(s(state)) do
        v  $\leftarrow$  MIN(v, MAX-VALUE(RESULT(s,a),  $\alpha, \beta$ ))
        if v  $\leq \alpha$  then return v
         $\beta \leftarrow \text{MIN}(\beta, v)$ 
    return v

```

Move Ordering

- With perfect ordering we need to examine only $O(b^{m/2})$ nodes
- When successors are examined in random order it needs to examine $O(b^{3m/4})$ nodes
- In chess, a strategy that **capture** \rightarrow **threat** \rightarrow **backward**, gets you to within about a factor of 2 of the best case $O(b^{m/2})$
- Try first the move that were found useful in past (**killer move**)
- Iterative deepening could help (adds constant fraction time)
- Hash table of previously seen positions (**transposition table**) can restrict re-computation of states

In-action: ALPHA-BETA Pruning



[Artificial Intelligence \(ZC444\)](#) | Sun (10:30-12:00PM) online@BITS-Pilani | [Lecture-09 \(Oct 15, 2023\)](#) | 13 / 23

Imperfect Real-time Decision

[Artificial Intelligence \(ZC444\)](#) | Sun (10:30-12:00PM) online@BITS-Pilani | [Lecture-09 \(Oct 15, 2023\)](#) | 14 / 23

Important Considerations

- Generating entire search space is overkill!
- Covering till large depth is not possible
- Idea is to cutoff the search earlier
- $H\text{-MINMAX}(s, d) = \begin{cases} \text{Eval}(s) & \text{argmin}_{a \in \text{Actions}(s)} H\text{-MINMAX}(RESULT(s, a), d+1) \\ \text{argmax}_{a \in \text{Actions}(s)} H\text{-MINMAX}(RESULT(s, a), d+1) & \text{if } CUTOFF_TEST(s, a) \\ \text{MAX} & \text{if } \text{PLAYER}(s) = \text{MAX} \\ \text{MIN} & \text{if } \text{PLAYER}(s) = \text{MIN} \end{cases}$

[Artificial Intelligence \(ZC444\)](#) | Sun (10:30-12:00PM) online@BITS-Pilani | [Lecture-09 \(Oct 15, 2023\)](#) | 16 / 23

Evaluation Function

- Returns the estimate of expected utility
 - Performance would deeply depend on it
 - $\text{Eval}(\text{win}) \geq \text{Eval}(\text{draw}) \geq \text{Eval}(\text{lose})$
 - Computation should be quick
 - Value should relate to actual chance of winning
 - Can use features (Number of pawns, Number of queens, ...)
 - Can have various categories (all pawn vs one pawn, etc) Suppose experience suggests 72% of states encountered in the two-pawn vs. one pawn category lead to win, 20% lose, and 8% in draw. Then eval could be
- $$0.72 \times 1 + 0.20 \times 0 + 0.08 \times 0.5 = 0.76$$

- Weighted linear function** are also possible

$$\text{Eval}(s) = w_1 \times f_1(s) + w_2 \times f_2(s) + w_3 \times f_3(s) + \dots$$

[Artificial Intelligence \(ZC444\)](#) | Sun (10:30-12:00PM) online@BITS-Pilani | [Lecture-09 \(Oct 15, 2023\)](#) | 17 / 23

[Artificial Intelligence \(ZC444\)](#) | Sun (10:30-12:00PM) online@BITS-Pilani | [Lecture-09 \(Oct 15, 2023\)](#) | 18 / 23

³ can take a previous good move

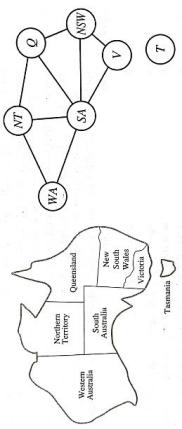
Constraint Satisfaction

Example: Map Coloring

Given three colors {red, green, blue}, can you color following map such that no two neighboring region have same color.

- A Constraint Satisfaction Problem (CSP) has three components
 - ➊ X as a set of variables $\{X_1, X_2, X_3, \dots, X_n\}$
 - ➋ D as a set of domains $\{D_1, D_2, D_3, \dots, D_n\}$
 - ➌ C set of constraints that specify allowable combinations of values
- Each D_i contains allowable set of values $\{v_1, v_2, v_3, \dots, v_k\}$ for X_i
- Each C_j contains a pair < scope, relation > such as $\langle (X_1, X_3), X_1 \neq X_3 \rangle$
- To solve CSP it needs **state space** and a notion of **solution**
- An assignment of variables such as $\{X_1 = v_1, X_2 = v_2, \dots, X_n = v_n\}$ that does not violates any constraints is called **consistent** or legal solution.
- Our target to have complete assignment that is consistent.

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 19/23

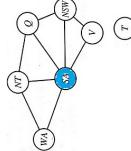


- Once you choose (SA=blue)
- you can conclude that none of its five members could take the color **blue**
- Without taking advantage of constraint propagation, search needs to consider $3^5 = 243$ assignments for the five neighbors.
- With constraint propagation, it needs $2^5 = 32$ only
- 87% reduction

Constraint Propagation

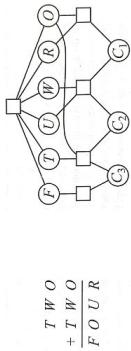
Example: Cryptarithmetic Puzzle

Consider following addition (we have to find digits)



- Once you choose (SA=blue)
- you can conclude that none of its five members could take the color **blue**
- Without taking advantage of constraint propagation, search needs to consider $3^5 = 243$ assignments for the five neighbors.
- With constraint propagation, it needs $2^5 = 32$ only
- 87% reduction

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 21/23



- Each letter represents a different digit
- We need AllDiff{F,T,U,W,R,O} and
- Additional constraints are

$$\begin{array}{rcl} O + O & = & R + 10 \times C_1 \\ C_1 + W + W & = & U + 10 \times C_2 \\ C_2 + T + T & = & O + 10 \times C_3 \\ C_3 & = & F \end{array}$$

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 22/23

Thank you very much for your attention!

Queries ?

Thank You!

(Reference⁴)

⁴1) Book - AIMa, ch-03+04, Russell and Norvig. 2) Book - Machine Learning, ch-09, Tom Mitchell

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 23/23

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-09 (Oct 15, 2023) 20/23