Game Playing (Tic-Tac-Toe), ANDOR graph

Outline of the Talk

- Game Playing
- Tic-Tac-Toe
- Minimax Algorithm
- Alpha Beta Prunning
- AndOr graph and AO* Algorithm
- Summary
- References

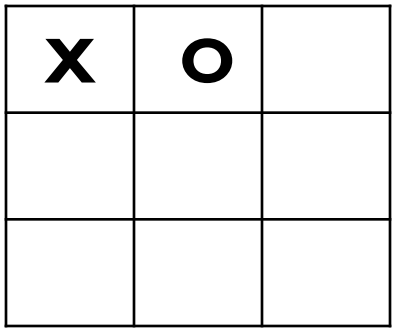
Games vs Search Problems

- "Unpredictable" opponent : specifying a move for every possible opponent reply
- Time limits : unlikely to find goal, must approximate

Game Playing Strategy

- Maximize winning possibility assuming that opponent will try to minimize (Minimax Algorithm)
- Ignore the unwanted portion of the search tree (Alpha Beta Pruning)
- Evaluation(Utility) Function
  - A measure of winning possibility of the player

Tic-Tac-Toe



e(p) = 6 - 5 = 1

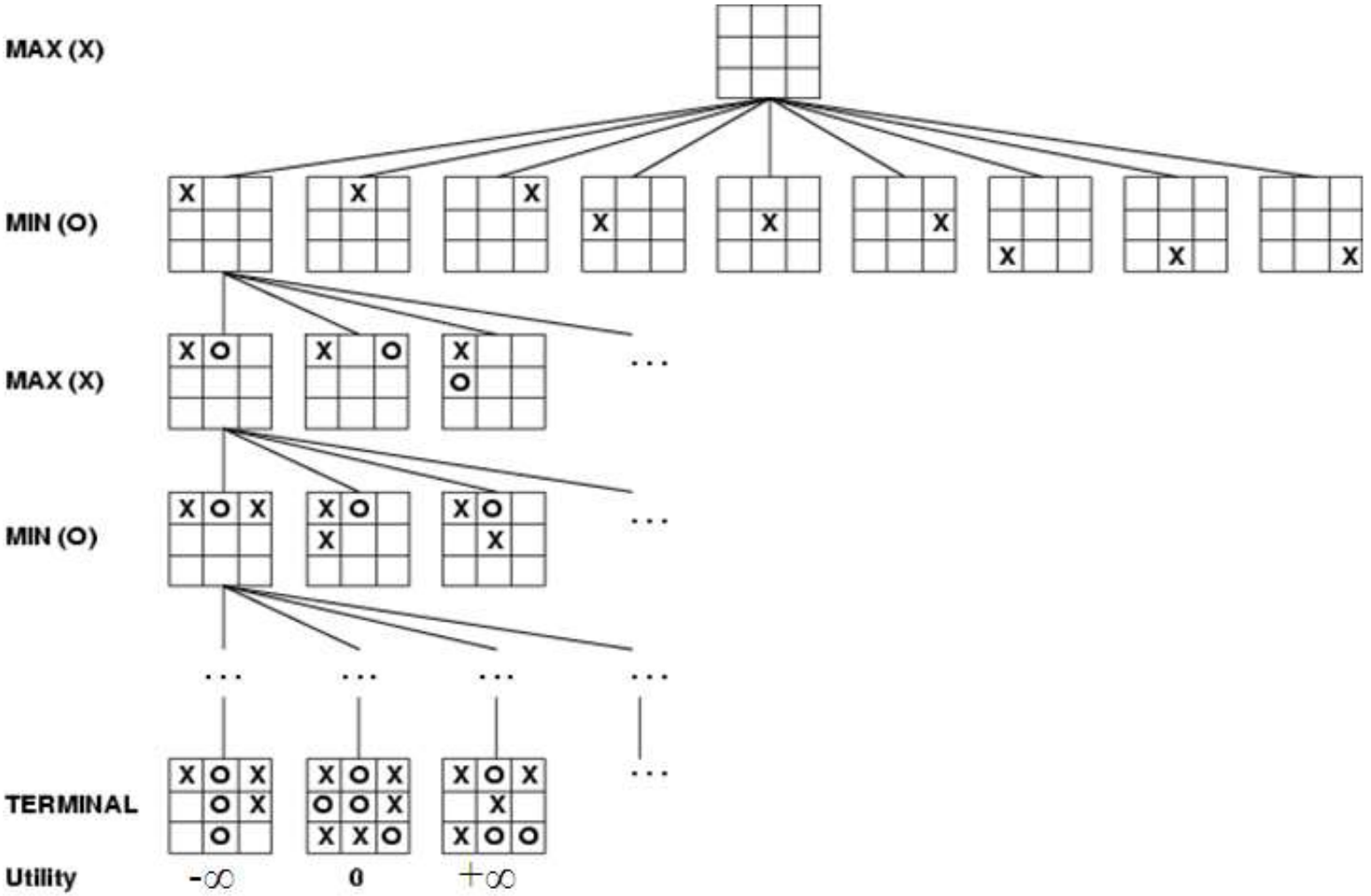
- **Initial State**: Board position of 3x3 matrix with 0 and X.
- **Operators:** Putting 0's or X's in vacant positions alternatively
- **Terminal test:** Which determines game is over
- **Utility function**:

  e(p) = (No. of complete rows, columns or diagonals are still open for player) – (No. of complete rows, columns or diagonals are still open for opponent)

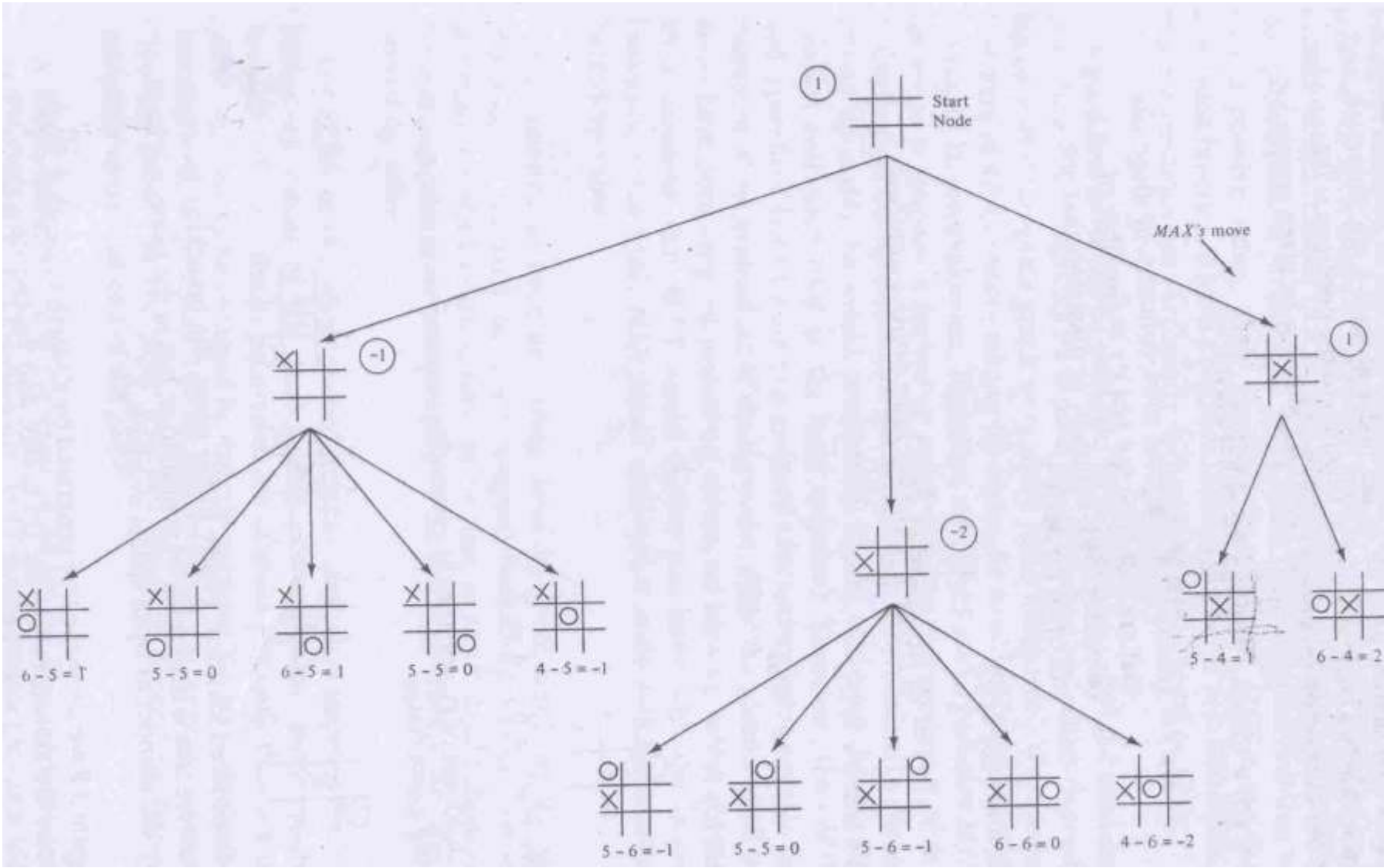
Minimax Algorithm

- Generate the game tree
- Apply the utility function to each terminal state to get its value
- Use these values to determine the utility of the nodes one level higher up in the search tree
  - From bottom to top
  - For a max level, select the maximum value of its successors
  - For a min level, select the minimum value of its successors
- From root node select the move which leads to highest value

Game tree for Tic-Tac-Toe



Courtesy : Artificial Intelligence and Soft Computing. Behavioural and Cognitive Modelling of the Human Brain



Courtesy : Principles of Artificial Intelligence , Nilsson

Properties of Minimax

- Complete : Yes (if tree is finite)
- Time complexity : $O(b^d)$
- Space complexity : O(bd) (depth-first exploration)

Observation

- Minimax algorithm, presented above, requires expanding the entire state-space.
- Severe limitation, especially for problems with a large state-space.
- Some nodes in the search can be *proven to be irrelevant to the outcome* of the search

Alpha-Beta Strategy

- Maintain two bounds:

  Alpha (α): a lower bound on best that the player to move can achieve

  Beta (β): an upper bound on what the opponent can achieve

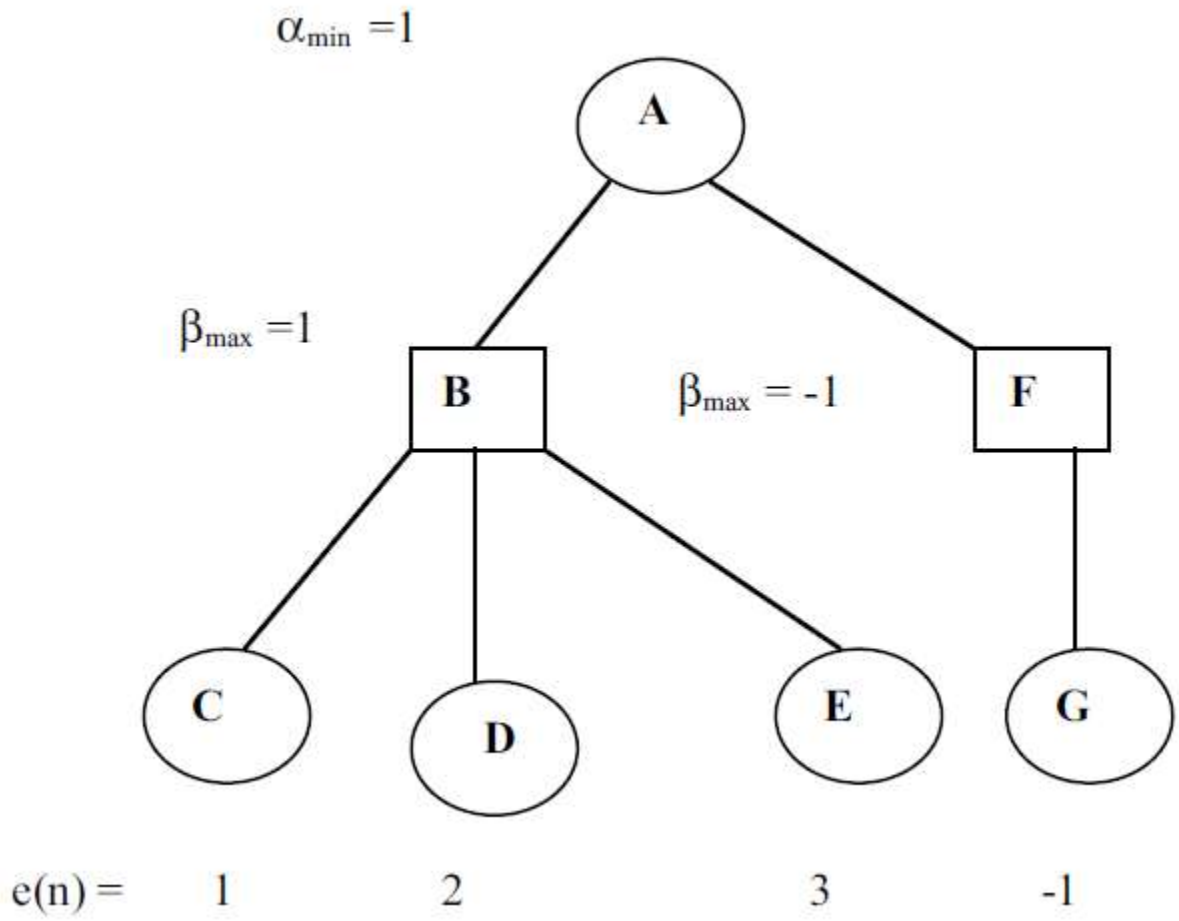
- Search, maintaining α and β
- Whenever $\alpha \geq \beta_{higher}$, or $\beta \leq \alpha_{higher}$ further search at this node is irrelevant

How to Prune the Unnecessary Path

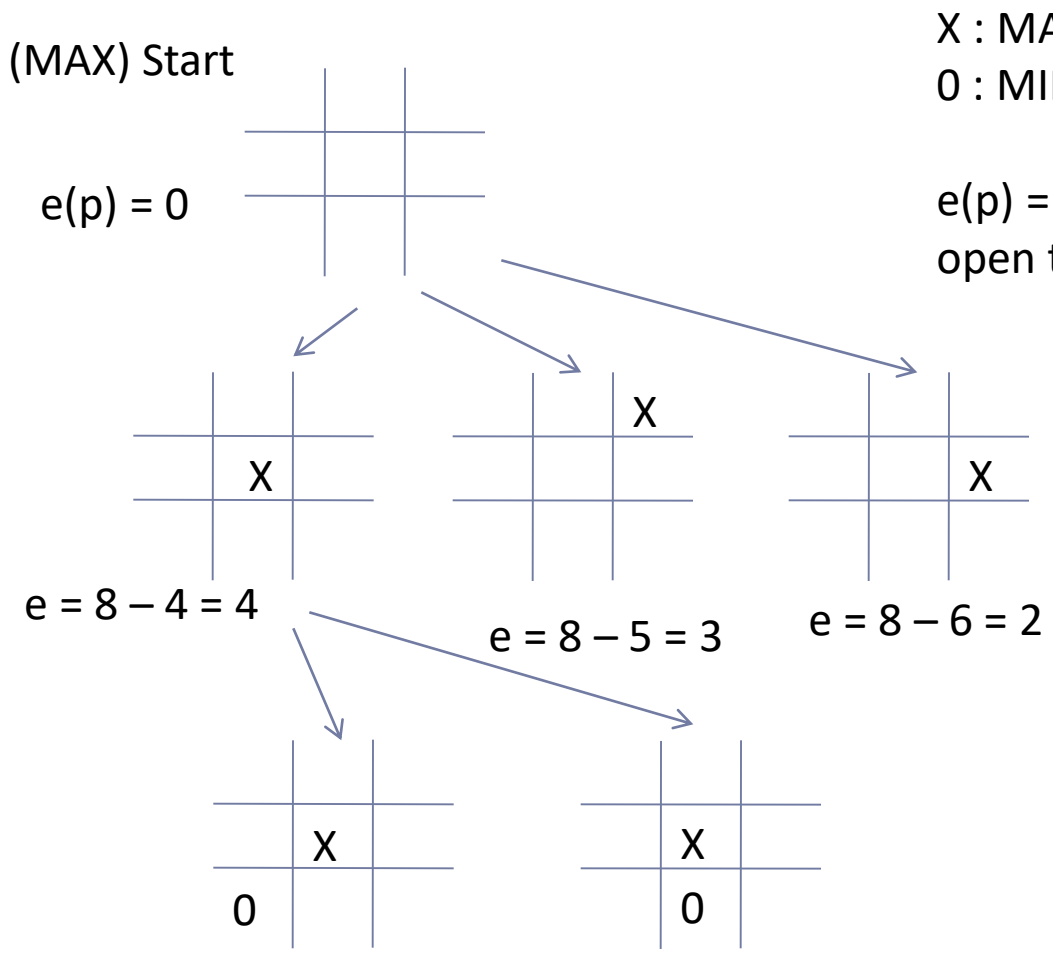
- **If** beta value of any MIN node below a MAX node is less than or equal to its alpha value, **then** prune the path below the MIN node.

- **If** alpha value of any MAX node below a MIN node exceeds the beta value of the MIN node, **then** prune the nodes below the MAX node.

Example



Tic-Tac-Toe



X : MAX player
0 : MIN player

e(p) = (rows + cols + diagonals open to 'X') – (Same to '0')

X's Turn

0's Turn

Courtesy : CS621-Artificial Intelligence , 2007, Prof. Pushpak Bhatacharya

Alpha-Beta Search Algorithm

- If the MAXIMIZER nodes already possess αmin values, then their current **αmin value = Max (αmin value, α'min);** on the other hand, if the MINIMIZER nodes already possess βmax values, then their current

  βmax value = Min (βmax value, β'max).

- If the estimated βmax value of a MINIMIZER node N is less than the αmin value of its parent MAXIMIZER node N' then there is no need to search below the node MINIMIZER node N. Similarly, if the αmin value of a MAXIMIZER node N is more than the βmax value of its parent node N' then there is no need to search below node N.

Alpha-Beta Analysis

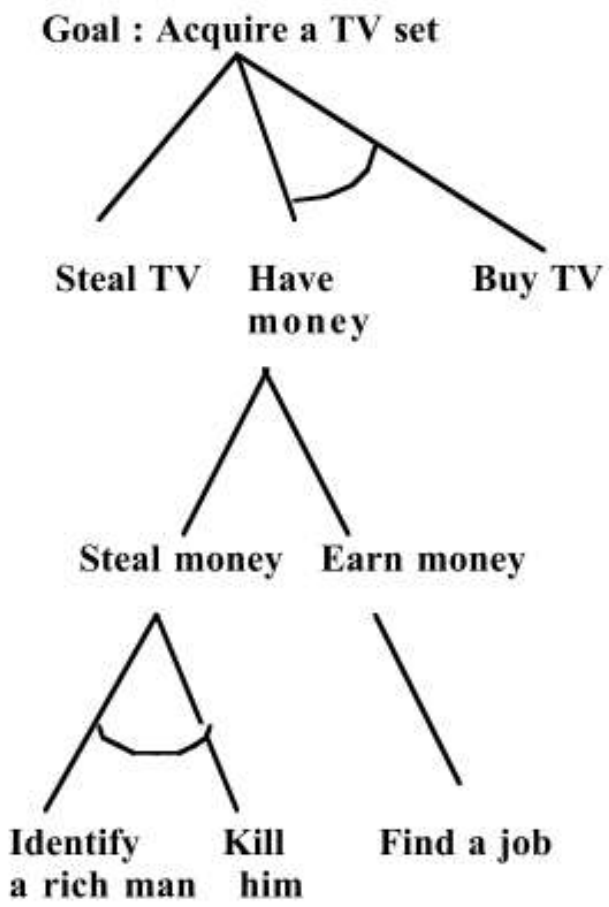
- Pruning does not affect the final result.
- Assume a fixed branching factor and a fixed depth
- Best case: $b^{d/2} + b^{(d/2)-1}$
- Approximate as $b^{d/2}$
- Impact ?

  Minmax: 10^9 = 1,000,000,000

  Alpha-beta: $10^5 + 10^4$ = 110,000
- But best-case analysis depends on choosing the best move first at cut nodes (not always possible)
- The worst case : No cut-offs, and Alpha-Beta degrades to Minmax

AND OR GRAPH

AND OR Graph



- OR graphs : generally used for data driven approach
- AND OR graphs: used for Goal driven approach
  - Problems solvable by decomposing into sub problems some of which is to be solved.
- Graph consisting of OR arcs and AND arcs
  - OR : the node has to be solved.
  - AND : all the nodes in the arc has to be solved

Courtesy : Artificial Intelligence and Soft Computing. Behavioural and Cognitive Modelling of the Human Brain

How to explore

- Expand nodes
- Propagate values to ancestors

- Futility
  - If the estimated cost of a solution becomes greater than futility then abandon the search
  - A threshold such that any solution with higher cost is too expensive to be practical

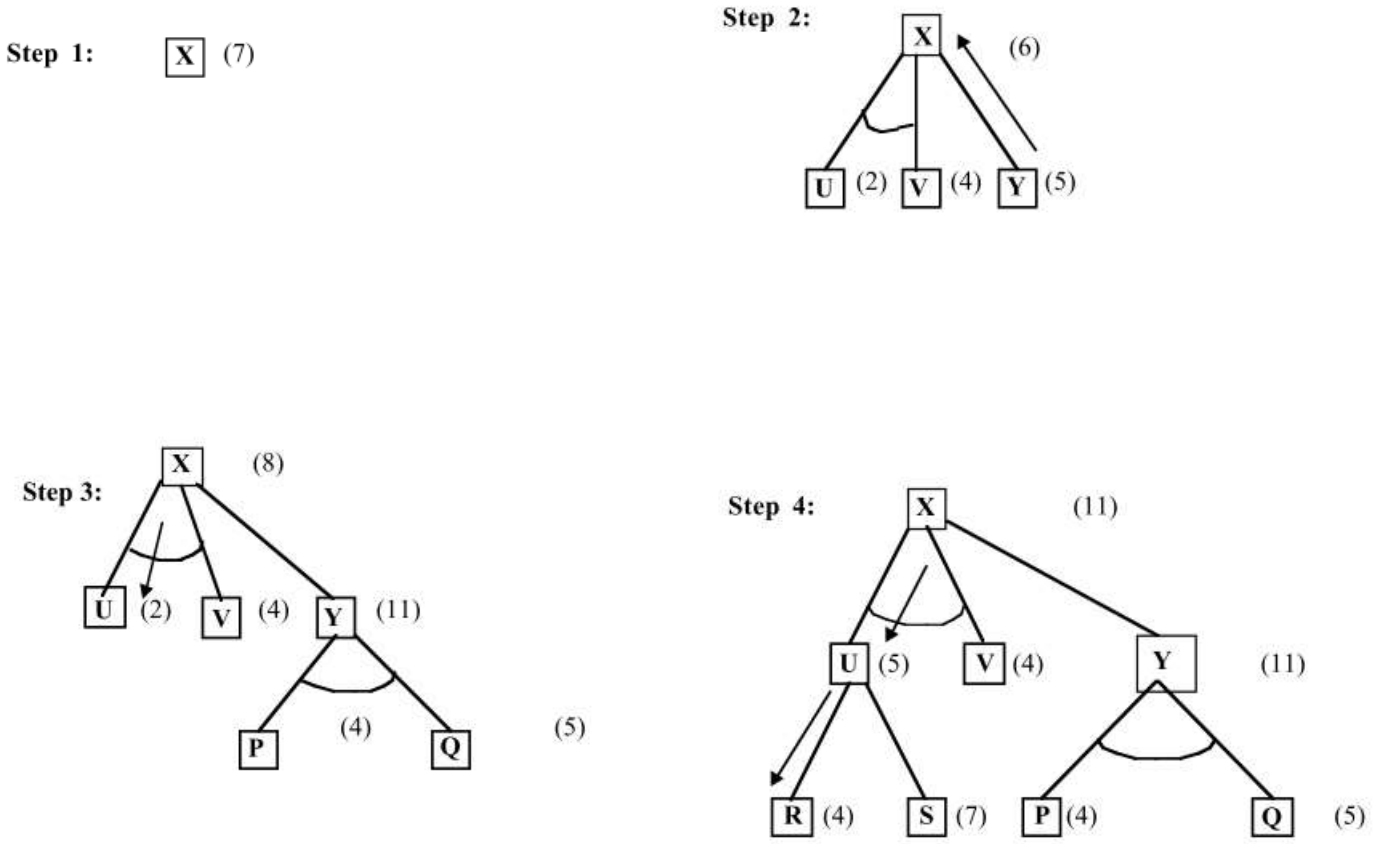
AO* (high level view)

1. Given the Goal node, find its possible off-springs.

2. Estimate the h values at the leaves. The cost of the parent of the leaf (leaves) is the minimum of the cost of the OR clauses plus one or the cost of the AND clauses plus the number of AND clauses. After the children with minimum h are estimated, a pointer is attached to point from the parent node to its promising children.

3. One of the unexpanded OR clauses / the set of unexpanded AND clauses, where the pointer points from its parent, is now expanded and the h of the newly generated children are estimated. The effect of this h has to be propagated up to the root by re-calculating the f of the parent or the parent of the parents of the newly created child /children clauses through a least cost path. Thus the pointers may be modified depending on the revised cost of the existing clauses.

AO* illustration



Courtesy : Artificial Intelligence and Soft Computing. Behavioural and Cognitive Modelling of the Human Brain

Summary

- Explore game tree
  - Min max
  - Alpha Beta
- When perfection is unattainable, we must approximate
- AND OR graph
  - How to explore
  - AO*

References

- D. E. Knuth and R. W. Moore. An analysis of alpha-beta pruning. Artificial Intelligence, 6:293–326, 1975
- Rich, E. and Knight, K., *Artificial Intelligence, McGraw-Hill, New* York, 1991.
- Nilson, J. N., *Principles of Artificial Intelligence, Morgan-Kaufmann,*

  San Mateo, CA, pp. 112-126, 1980.
- Russel, S. and Norvig, P., *Artificial Intelligence: A Modern Approach,*

  Prentice-Hall, Englewood Cliffs, NJ, 1995.
- Amit Konar , *Artificial Intelligence and Soft Computing Behavioral and Cognitive Modeling of the Human Brain,* CRC Press 2000.