



IS/SE/SS ZC444: Artificial Intelligence

10

Constraint Satisfaction



Dr. Kamlesh Tiwari

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

Oct 19, 2023

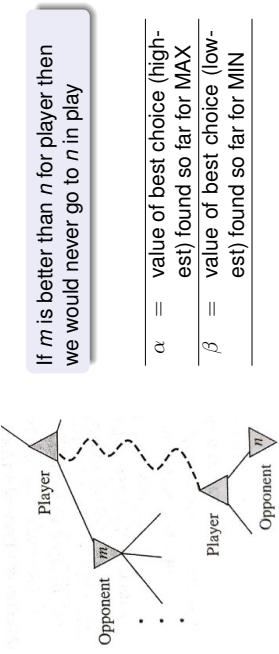
ONLINE

WILP @ BITS-Pilani [July-Dec 2023]

<http://ktiwari.in/ai>

ALPHA-BETA Pruning

- Alpha-beta pruning can be applied to trees of any depth, and it is often possible to prune entire subtree rather than just leaves.



```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 3/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 4/26
```

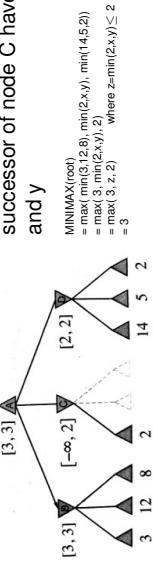
```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 5/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 6/26
```

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



```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 2/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 4/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 5/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 6/26
```

- Order matters.
- So, examine likely to be best successor first.

```
function ALPHA-BETA-SEARCH(state) returns an action
  v <- MAX-VALUE(state, -infinity, +infinity)
  return the action in ACTIONS(state) with value v

function MAX-VALUE(state, alpha, beta) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  v <- -infinity
  for each a in ACTIONS(state) do
    v <- MAX(v, MIN-VALUE(RESULT(s,a), alpha, beta))
    if v >= beta then return v
    alpha <- MAX(alpha, v)
  return v

function MIN-VALUE(state, alpha, beta) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  v <- +infinity
  for each a in ACTIONS(state) do
    v <- MIN(v, MAX-VALUE(RESULT(s,a), alpha, beta))
    if v <= alpha then return v
    beta <- MIN(beta, v)
  return v
```

- Is it possible?
- No

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 → threat → forward → 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

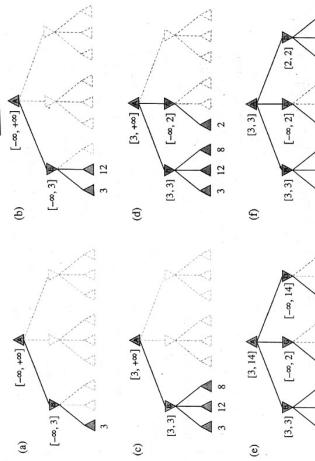
```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 3/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 4/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 5/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 6/26
```

In-action: ALPHA-BETA Pruning



```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 3/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 4/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 5/26
```

```
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 6/26
```

Imperfect Real-time Decision

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.
- $H\text{-MINIMAX}(s, d) = \begin{cases} \text{Eval}(s) & \text{if } \text{CUTOFF-TEST}(s, d) = \text{MAX} \\ \arg\max_{a \in \text{Actions}(s)} H\text{-MINIMAX}(\text{RESULT}(s, a), d+1) & \text{if } \text{PLAYER}(s) = \text{MAX} \\ \arg\min_{a \in \text{Actions}(s)} H\text{-MINIMAX}(\text{RESULT}(s, a), d+1) & \text{if } \text{PLAYER}(s) = \text{MIN} \end{cases}$

$$0.72 \times 1 + 0.20 \times 0 + 0.08 \times 0.5 = 0.76$$

- 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-10 (Oct 19, 2023) 7/26

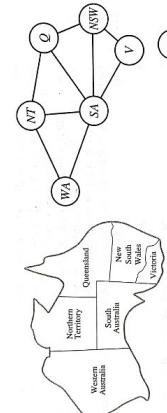
Important Considerations

- A Constraint Satisfaction Problem (CSP) has three components
 1. X as a set of variables $\{X_1, X_2, X_3, \dots, X_n\}$
 2. D as a set of domains $\{D_1, D_2, D_3, \dots, D_n\}$
 3. 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_i contains a pair $\langle \text{scope}, \text{relation} \rangle$ 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-10 (Oct 19, 2023) 9/26

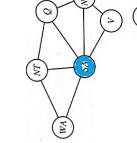
Example: Map Coloring

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



Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 10/26

Constraint Propagation



- 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
- $X = \{WA, NT, Q, NSW, V, SA, T\}$
- $C = \{SA \neq WA, SA \neq NT, SA \neq NSW, SA \neq NT, Q \neq NSW, NSW \neq V\}$
- There are many solution to the problem
 - It is helpful to visualize as a **constraint graph** (node: variable, edge: participate in constraint)

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 11/26

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 12/26

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 8/26

Example: Cryptarithmehic Puzzle

Consider following addition (we have to find digits)

$$\begin{array}{r}
 \begin{array}{r}
 T \ W \ O \\
 + T \ W \ O \\
 \hline
 F \ O \ U \ R
 \end{array}
 \\[1em]
 \begin{array}{l}
 \text{Each letter represents a different digit} \\
 \text{We need Alldiff}\{F,T,U,W,R,O\} \text{ and} \\
 \text{Additional constraints are}
 \end{array}
 \end{array}$$

- If SA do not like green color then use {red,blue} instead of {red,green,blue}

²General consistency is possible: using more than two variables in a constraint

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

Inference in CSP

- In CSP, an algorithm can either
 - 1 Search or
 - 2 Do inference: constraint propagation (that reduces number of legal values for another variable)

Enforcing local consistency in each part of the graph can cause inconsistency elimination throughout the graph.

- **Node consistency:** if all the values in variable's domain satisfy the variable's unary constraints². It is always possible to eliminate all unary constraints by applying Node consistency
- **Arc consistency:** X_i is arc consistent with X_j if for every value in current domain D_i there is some value in D_j satisfying binary constraint on (X_i, X_j) . Note³

²If SA do not like green color then use {red,blue} instead of {red,green,blue}

³General consistency is possible: using more than two variables in a constraint

$$\begin{array}{r}
 \begin{array}{r}
 T \ W \ O \\
 + T \ W \ O \\
 \hline
 F \ O \ U \ R
 \end{array}
 \\[1em]
 \begin{array}{l}
 \text{Each letter represents a different digit} \\
 \text{We need Alldiff}\{F,T,U,W,R,O\} \text{ and} \\
 \text{Additional constraints are}
 \end{array}
 \end{array}$$

Preferential Constraints

- Many real world CSPs include **preference constraints**
 - Indicating some solution are preferred over other
 - Consider university class scheduling problem
 - Apart from absolute constraints such as no professor could simultaneously teach two classes
 - There are some preferential constraints such as Prof. A prefer teaching in morning whereas Prof. B prefer teaching on evening.
 - A solution that schedules Prof. A in evening and Prof. B in morning is still ok
 - But, we do not prefer it
 - Such problems are sometimes called **constraint optimization problem** (COP)

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 14/26

AC-3 Algorithm

```

function AC-3(csp) returns false if an inconsistency is found and true otherwise
inputs: csp, a binary CSP with components ( $X$ ;  $D$ ;  $C$ )
local variables: queue, a queue of arcs, initially all the arcs in csp
while queue is not empty do
  ( $X_i$ ,  $X_j$ )  $\leftarrow$  REMOVE-FIRST(queue)
  if REVISE(csp,  $X_i$ ,  $X_j$ ) then
    if size of  $D_i = 0$  then return false
    for each  $X_k$  in  $X_i$ .NEIGHBORS -  $\{X_j\}$  do
      add  $(X_k, X_i)$  to queue
    return true

function REVISE(csp,  $X_i$ ,  $X_j$ ) returns true iff we revise the domain of  $X_i$ 
revised  $\leftarrow$  false
for each  $x$  in  $D_i$  do
  if no value  $y$  in  $D_j$  allows  $(x,y)$  to satisfy the constraint between  $X_i$  and  $X_j$  then
    delete  $x$  from  $D_i$ 
    revised  $\leftarrow$  true
  return revised

```

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 16/26

Takes $O(cd^3)$ time in worst case



Example: Sudoku

- Arc consistency can help if some domain becomes empty, or size of every domain reduces to 1
- **Path consistency:** two variable set $\{X_i, X_j\}$ is path consistent wrt X_m if for every $\{X_i = a, X_j = b\}$ consistent with constraints on $\{X_i, X_j\}$ there is an assignment to X_m that satisfies constraints on $\{X_i, X_m\}$ and $\{X_m, X_j\}$
- **K-consistency:** for any set of $k - 1$ variables and for any constraint assignment to those variables, a consistent value can always be assigned to any k^{th} variable.

A CSP is strongly k -consistent if it is **k -consistent**, **$k-1$ -consistent**, **$k-2$ -consistent, ..., 1 -consistent**⁴

- **Global Constraints:** like **alldiff**. If all m variables involved in alldiff have only n possible values where $m > n$ then there is no solution.

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 17/26

⁴Finding such consistency is hard

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 18/26

Backtracking Search for CSP

Backtracking Search

- When inference only do not work, use search
 - CSP with n variable and d domain size can have branching factor nd^d at top level. Then $(n - 1)d^d$ in next level and so on.
 - Tree with $!n \cdot d^n$ leaves get generated (however valid assignments are only d^n)
 - It is why we have ignored **commutativity**⁵
 - So consider single variable at a node.
 - Now we need to **backtrack**, if no legal value is left for assignment
-
- ⁵No effect of assignment order

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 21/26

Backtracking Search

- Which variable to choose next? **minimum remaining value**⁶ or “fail first”
 - Degree heuristic**, choose one which is involved in many constraints
 - Which value to choose? **least constrained value**
 - Forward checking**: interleaving search and inference would help. Whenever a variable X is assigned, forward checking establishes arc consistency for it.
 - Intelligent Backtracking**: Some time it is needed to backtrack upward more than a single step to resolve the inconsistency. Conflicting set is used to find most suitable node.
-
- ⁶choose whose domain have fewer entries

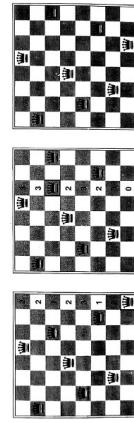
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 21/26

```
function BACKTRACKING-SEARCH( $csp$ ) returns a solution, or failure
  return BACKTRACK( $\{\}$ ,  $csp$ )
  
  function BACKTRACK( $assignment$ ,  $csp$ ) returns a solution, or failure
    if  $assignment$  is complete then return  $assignment$ 
     $var \leftarrow$  SELECT-UNASSIGNED-VARIABLE( $csp$ )
    for each value in ORDER-DOMAIN-VALUES( $var$ ,  $assignment$ ,  $csp$ ) do
      if value is consistent with  $assignment$  then
        add  $\{var = value\}$  to  $assignment$ 
         $inferences \leftarrow$  INFERENCE( $csp$ ,  $var$ ,  $value$ )
        if  $inferences \neq failure$  then
          add  $inferences$  to  $assignment$ 
           $result \leftarrow$  BACKTRACK( $assignment$ ,  $csp$ )
          if  $result \neq failure$  then
            return  $result$ 
        remove  $\{var = value\}$  and  $inferences$  from  $assignment$ 
      return  $failure$ 
```

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 20/26

Local Search for CSP

Complete state space formulation can also be used for search



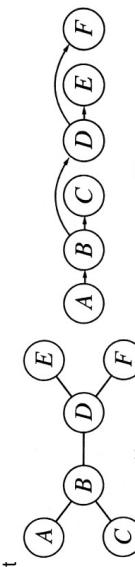
```
function MIN-CONFFLICTS( $csp$ ,  $max\_steps$ ) returns a solution  $r$  failure
inputs:  $csp$ , a constraint satisfaction problem
 $max\_steps$ , the number of steps allowed before giving up
 $current \leftarrow$  an initial complete assignment for  $csp$ 
for  $i = 1$  to  $max\_steps$  do
  if  $current$  is a solution for  $csp$  then return  $current$ 
   $var \leftarrow$  a randomly chosen conflicted variable from  $csp$ .VARIABLES
   $value \leftarrow$  the value  $v$  for  $var$  that minimizes CONFLICTS( $var, v, current, csp$ )
  set  $var = value$  in  $current$ 
return  $failure$ 
```

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 22/26

Local Search for CSP

Sometime structure of the problem could help to find solution

- Independent Subproblems**: see that Tasmania is not connected to mainland in Australia map.
- Compare $O(d^c n/c)$ with $O(d^n)$ it is linear where each sub problem has c variables
- Tree structured CSP** is solvable in linear time $O(nd^2)$ with topological sort



- Cut set conditioning**: Assign few, to get tree from remaining vars
- $O(d^c(n - c)d^c)$
- Another approach is **tree decomposition**

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 24/26

⁷Airline wants to repair its schedule with minimum changes.

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 23/26

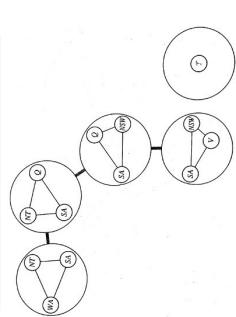
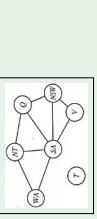
Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani Lecture-10 (Oct 19, 2023) 24/26

Tree Decomposition for CSP

Thank You!

Divide the problem in sub-problems

- Every variable in original problem appears at least one of the subproblems
- If two variables are connected by a constraint in original problem, then they must appear together in at least one of the subproblem
- If a variable appears in two subproblems in the tree, it must appear in every subproblem along the path connecting those subproblems



(Reference⁸)

8.1) Book - AIM/A ch-06, Russell and Norvig.

Lecture-10 (Oct 19, 2023) 26 / 26

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani

Lecture-10 (Oct 19, 2023) 26 / 26

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani

Thank you very much for your attention!

Queries ?

Lecture-10 (Oct 19, 2023) 26 / 26

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani

Lecture-10 (Oct 19, 2023) 26 / 26

Artificial Intelligence (ZC444) Sun (10:30-12:00PM) online@BITS-Pilani