CSL020U4E: Artificial Intelligence Lecture-06 (Problem Decomposition - AO*-Algorithm)

Sumit Kumar Pandey

November, 2025

The algorithm AO^* operates in a cycle of two phases.

- In the forward phase, the algorithm follows a set of markers that identify the cheapest partial solution and extends it.
- In the backward phase, it propagates the revised costs up towards the root. It terminates when the label solved is propagated up to the root, or if the cost estimate is beyond an acceptable limit.

AO^*(start, Futility)
  add start **to** G
  compute h(start)
  solved(start) $\leftarrow$ FALSE
  while solved(start) = FALSE **and** h(start) < Futility
    label: FORWARD PHASE
    label: PROPAGATE BACK
  if solved(start) = TRUE
    return the marked subgraph from start **node**
  else return null

AO^*

label: FORWARD PHASE

$U \leftarrow$ **trace marked paths in** G **to a set of unexpanded nodes**

$N \leftarrow$ **select a node from** U

children $\leftarrow$ Successors(N)

if children **is empty**

  $h(N) \leftarrow$ Futility

else

  check for looping in the members of children

  remove any looping members from children

  for each S $\in$ children

    add S **to** G

    compute $h(S)$

    if S **is primitive**

      solved(S) $\leftarrow$ TRUE

label: PROPAGATE BACK
  $M \leftarrow \{N\}$ /* set of modified nodes */
  while M **is not empty**
   $D \leftarrow$ **remove deepest node from** M
   compute best cost of D **from its children**
   mark best option at D **as** MARKED
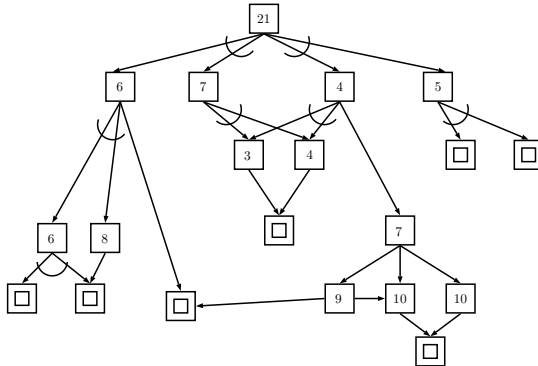   **if all nodes connected through marked arcs are solved**
    solved(D) $\leftarrow$ TRUE
   if D **has changed**
    add all parents of D **to** M

Exercise

Find an optimal path using AO^* algorithm:

1. Each edge has cost 10.
2. Each solved node has cost 0.

Thank You