2 The shortest path problem on a weighted acyclic digraph

The problem. Given an acyclic digraph $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ with $|\mathcal{N}| = n$ and $|\mathcal{A}| = m$ and a cost function $w : \mathcal{A} \mapsto \Re$, find the shortest path from a given node $s \in \mathcal{N}$ to a given node $t \in \mathcal{N}$. An digraph is acyclic if and only if it does not contain any circuit.

Step 1: sequencing the decisions. Since $\mathcal{D}$ is acyclic, it is possible to sort its nodes in pre-topological order in $O(m)$. If $\mathcal{D}$ is also layered, it is possible to sort the layers and it is not necessary to sort the nodes in each layer (any order fits). We indicate by $Pred(i) \subset \mathcal{N}$ the set of predecessors of each node $i \in \mathcal{N}$:

$$Pred(i) = \{j \in \mathcal{N} : (j, i) \in \mathcal{A}\}.$$

Step 2: defining the state. The state consists of the last reached node. All $s - i$ paths leading to node $i \in \mathcal{N}$ correspond to sub-policies leading to the same state. Hence states have the following (trivial) form: $\{i\}$, with $i \in \mathcal{N}$. The cost associated with each state $\{i\}$ is indicated by $c(i)$.

Step 3: state extension.

- Initialization: $c(s) = 0$.
- Extension: $c(i) = \min_{j \in Pred(i)}\{c(j) + w_{ji}\}$.

The optimal value is $c(t)$ when the algorithm stops.

Complexity. When extending states, each arc is considered only once. Hence the complexity is $O(m)$. Since each node has a single *label* $c(i)$ and its value is computed only once, the resulting D.P. algorithm is a *label setting* algorithm.

A numerical example.

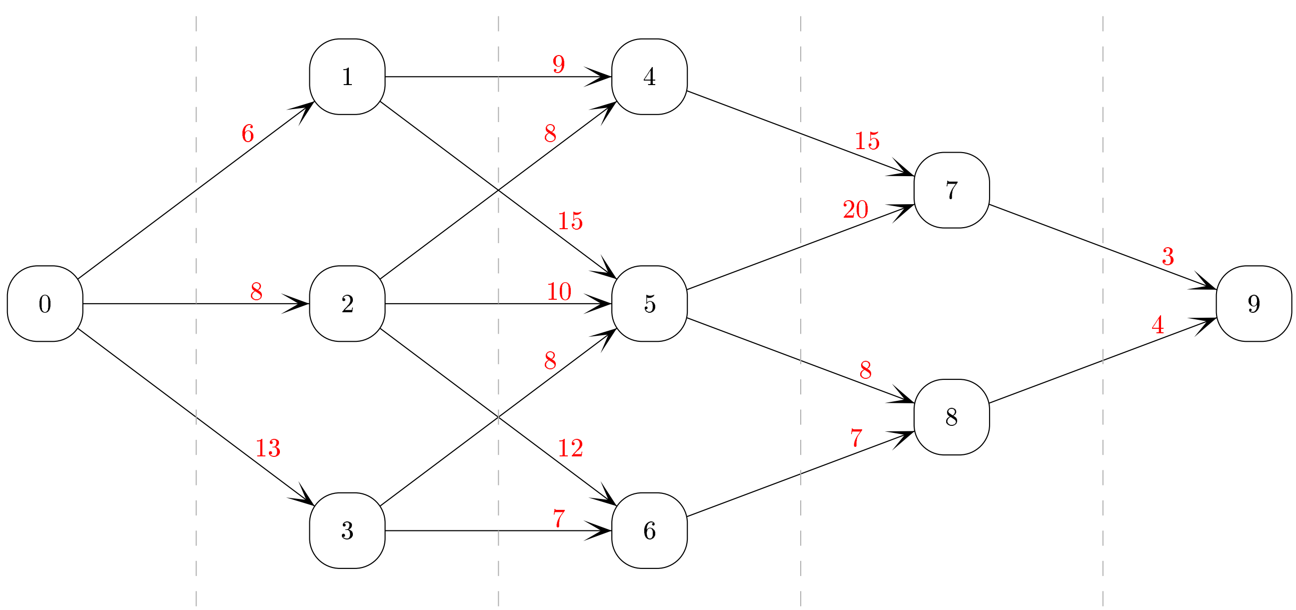

Figure 1: A weighted acyclic (and layered) digraph.

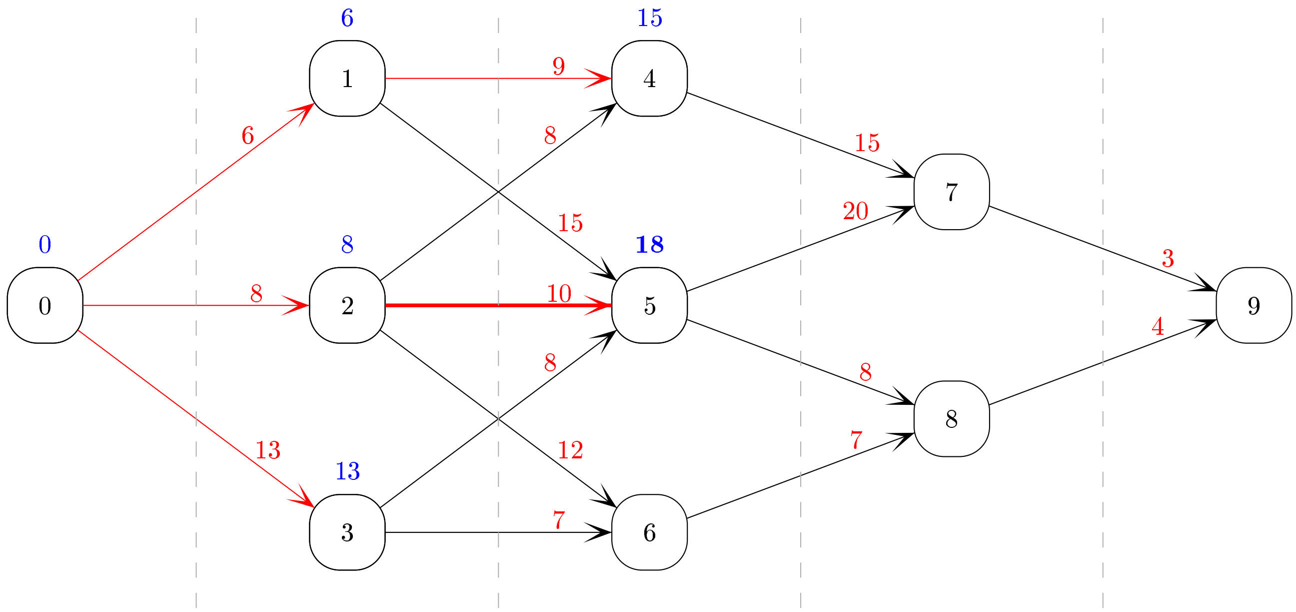

Figure 2: A state extension. Nodes 0 to 4 have already been labeled. Their associated costs are shown in the blue labels. Optimal predecessors are represented by red arcs. Node 5 is now labeled by comparing three predecessor states, yielding costs equal to 21 (from node 1), 18 (from node 2) and 21 (from node 3) and selecting the best option (the second one).

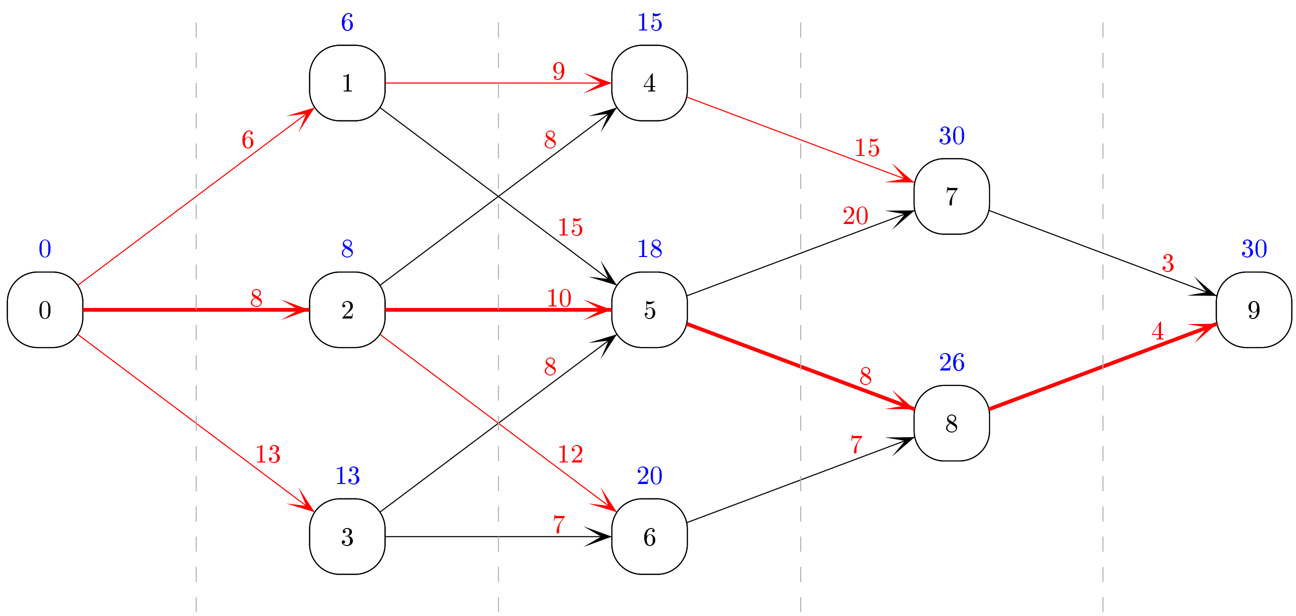

Figure 3: The example solved. The optimal value is 30 and the corresponding optimal solution can be reconstructed by following the chain of predecessors backward from node 9 to node 1: it is represented by the thick red arcs.