

Dynamic Programming

Dynamic Programming is an art of problem solving.

- ☐ Bellman
- ☐ Bellman & Dreyfus
- ☐ Dreyfus & Law

Dynamic Programming

It is a general method. For a special problem
we may be able to do better.

Let nodes denote the state of a system.

E.g. The state of a node is it shortest distance
from V_0 .

“State Variables”

Once you are in the node, you can consider several things.

E.g. Which neighboring node do you go to next.

“Control Variable”

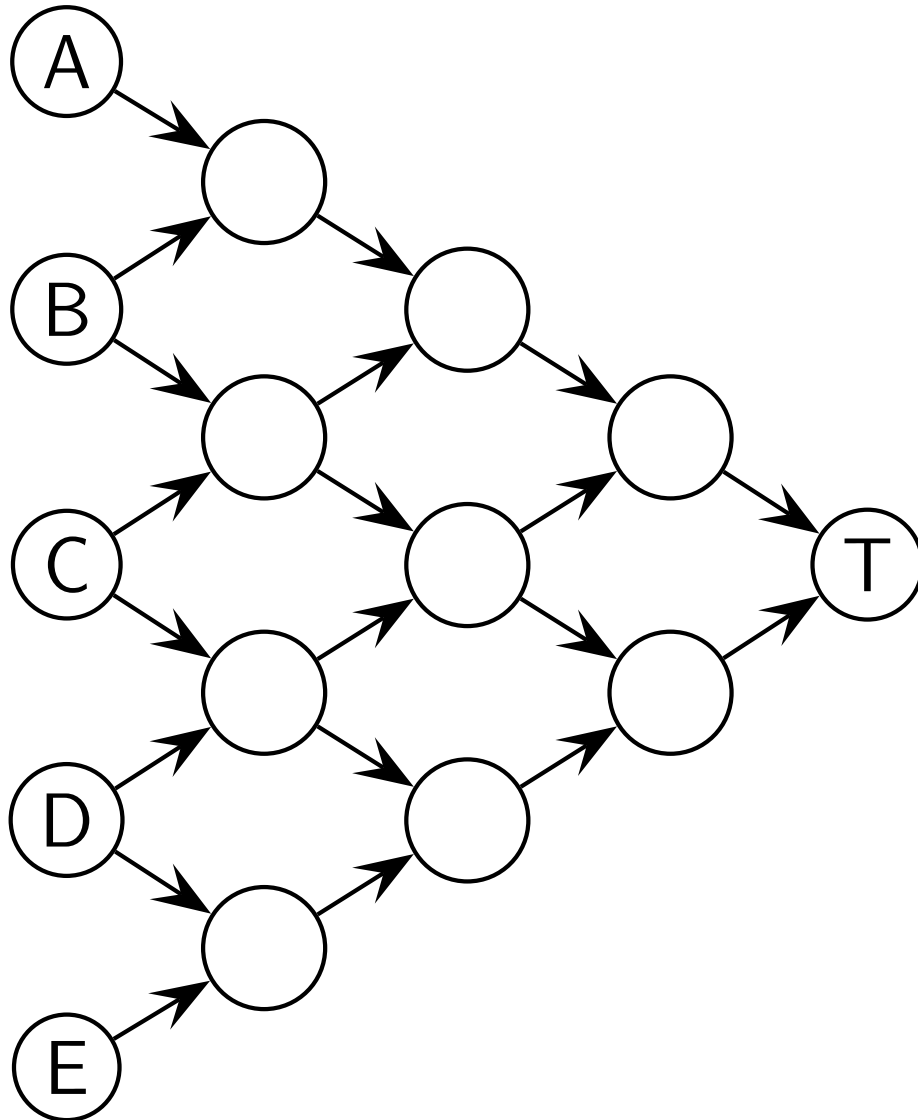
Dynamic Programming

Dynamic programming is the art of formulating the model so you can have an optimal policy.

Principle of Optimality

An optimal policy has the property that whatever the internal state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

Example



What is the shortest path to **T**?

Example Problem

Given a sequence $a_1, a_2, \dots, a_i, \dots, a_n$, select a subset of a_j with the total sum maximum and no two selected numbers are adjacent in the sequence.

E.g. 4, 1, 3, 9, 8, 7

Application: Month-to-Month rent of an apartment, where you need clean-up time.

Solution

Define $F(n)$: Max sum when a_n is selected.

$$F(1) = a_1$$

$$F(2) = \max(a_1, a_2)$$

In $F(n)$, either a_n is not selected, or a_n is selected.

$$F(n) = \max \begin{cases} F(n-1) \\ a_n + F(n-2) \end{cases}$$

$$F(0) = 0$$

Execution of this Algorithm

$$F(1) = a_1$$

$$F(2) = \max(F(1), F(0) + a_2)$$

$$F(3) = \max(F(2), F(1) + a_3)$$

$$F(4) = \max(F(3), F(2) + a_4)$$

$$F(k) = \max(F(k-1), F(k-2) + a_k)$$

$$a_i = 4, 1, 3, 9, 8, 7, 5, 8, 13$$