# Dynamic Programming Based Image Segmentation in Biomedical Imaging

Aaditya - CB.EN.U4CSE18402
Samyukth - CB.EN.U4CSE18451

## Introduction:

Dynamic Programming (DP), introduced by Richard Bellman, is a widely used algorithm paradigm to solve optimization techniques in a simple and efficient way.

In biomedical imaging, DP is a popular technique to find contours, lines and boundaries of organs, bones, vessels and cells.

This case study focuses on applications of DP in the field of biomedical imaging, particularly on the detection and tracking of contours and structures by the means of DP.

## Motivation:

The main motivation for using DP in ~~the~~ biomedical imaging is to help Physicians to automatically detect, track and analyze structures in biomedical images, to reduce the expert's workload and improve the accuracy of the diagnosis.

## Common use of DP in biomedical imaging:

→ Finding the shortest path to detect a contour (or) boundary in the image by minimizing some energy function

→ Detecting the mammographic mass or the segmentation of the optic disk in retinal fundus images of the eye, aiming to find circular structures.

→ Detecting vascular trees and representing them with graph by means of a region growing technique.

## Exploring applications of DP in detail:

① Solving shortest path problems

The single source shortest path problem is the most frequently used type and can be efficiently solved by DP.

DP sequentially solves the shortest path problem by splitting it into simpler sub-problems.

A shortest path search is often utilized for discrete energy minimization.

A common description of energy in Computer Vision consists of 2 terms: Energy based on observations in some underlying data, and energy of some prior, including constraints of smoothness:

$$E = E\_data + E\_prior \longrightarrow ①$$

Let $\langle x_1, x_2, \cdots, x_n \rangle$ be an arbitrary path of $n$ elements. Then, the energy of this path is defined as:

$$E = E\langle x_1, x_2, \cdots, x_n \rangle$$

Now, the energy of the optimal path $\langle x_1^*, x_2^*, \cdots, x_n^* \rangle$ is obtained by minimizing $E$:

$$\min(E) = E\langle x_1^*, x_2^*, \cdots x_n^* \rangle$$

Figure ① demonstrates the possible paths in a graph from node $s$ to node $t$:

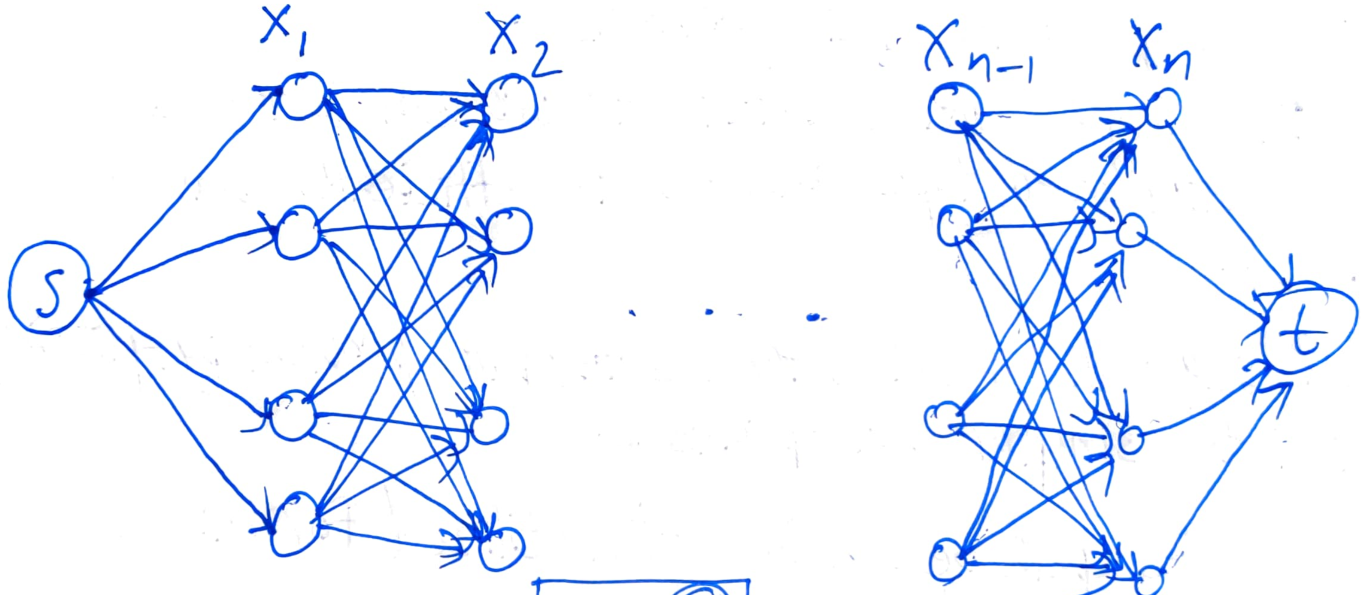


Fig. ①

In accordance to ①, the energy of a discrete path can be described as follows:

$$E\langle x_1, x_2, \cdots, x_n \rangle = \sum_{i=1}^{n} c(x_i) + \sum_{i=2}^{n} d(x_{i-1}, x_i)$$

Where

$C(x_i) \to$ cost of the edge passing through $x_i$

$d(x_{i-1}, x_i) \to$ Cost of partial path between $x_{i-1}$ and $x_i$

$C(x_i)$ can be, for example, a feature computed on the basis of image intensity data, while $d(x_{i-1}, x_i)$, also known as the smoothness term, is typically a geometric cost where specific neighborhoods can be penalized in terms of their position to each other, and thus is based on some prior knowledge of the path characteristics.

DP splits the energy evaluation of the path into simpler sub-problems, as shown below:

$$E\langle x_1\rangle = C(x_1)$$

$$E\langle x_1, x_2\rangle = E(x_1) + C(x_2) + d(x_1, x_2)$$

$$E\langle x_1, x_2, \ldots, x_i\rangle = E\langle x_1, x_2, \ldots, x_{i-1}\rangle + C(x_i) + d(x_{i-1}, x_i)$$

$$E\langle x_1, x_2, \ldots, x_i, \ldots x_n\rangle = E\langle x_1, x_2, \ldots x_i, \ldots, x_{n-1}\rangle + C(x_n) + d(x_{n-1}, x_n)$$

whith $1 \leq i \leq n$. Energy minimization is done using the following recursive formula:

$$c_1(x_1) = C(x_1)$$

$$c_i(x_i) = C(x_i) + \min\{c_{i-1}(x_{i-1}) + d(x_{i-1}, x_i)\}$$

the algorithm is as follows:

```
Algorithm shortest_dag
    for node x_1 = 1, ..., m at first state do
        c_1[x_1] <- E(x_1)
    end
    for state i = 2, ..., n do
        for node x_i = 1, ..., m at state i do
            C_i[x_i] <- min{C_{i-1}[x_{i-1}] + E(x_{i-1}, x_i)}
        end
    end
```

thus, the energy optimization problem is solved using DP.

## Conclusion:

From the above case study, we can understand the applications of DP in biomedical imaging and its significance in many real-life applications, given that DP has made solutions to many problems computationally efficient and neat. Some ~~import~~ famous DP algorithms are:

- → Unix diff ⇒ Comparing 2 files
- → Bellman-Ford ⇒ Shortest path routing in 2 graphs
- → TeX ⇒ ancestor of LaTeX
- → WASP ⇒ Winning And Score Predictor.

The above points highlight the fact that this algorithm design paradigm has a wide scope apart from academic purpose, and is being used extensively.